8-1988

The WARRANT Project: Learner-Centered Computer Environments For Critical Reading, Reasoning, & Writing

Preston K. Covey  
Carnegie Mellon University, dtrollcovey@gmail.com

Cheryl Geisler  
Rensselaer Polytechnic Institute

David S. Kaufer  
Carnegie Mellon University, kaufer@andrew.cmu.edu

Christine Neuworth  
Carnegie Mellon University

Follow this and additional works at: http://repository.cmu.edu/philosophy

Part of the Philosophy Commons

This Technical Report is brought to you for free and open access by the Dietrich College of Humanities and Social Sciences at Research Showcase @ CMU. It has been accepted for inclusion in Department of Philosophy by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.
The WARRANT Project: Learner-Centered Computer Environments For Critical Reading, Reasoning, & Writing

Preston K. Covey, Cheryl Geisler, David S. Kaufer, & Christine M. Neuwirth

August, 1988

CDEC Technical Report # 88-27

The WARRANT project has been supported by grants from the Fund for the Improvement of Postsecondary Education, the Alfred P. Sloan Foundation, and the Pew Memorial Trust.
The WARRANT Project

1 Paragraph Summary 1
2 Executive Summary 2
3 Project Overview 5
4 Purpose 5
5 Background 6
6 Project Description 7
   6.1 Original Plans 7
   6.2 Primary Actions 7
      6.2.1 Data Collection & Analysis 7
      6.2.2 Curricular Development 7
      6.2.3 Computer System Development 8
   6.3 The Computer Tools 8
      6.3.1 The Notes Program 8
      6.3.2 The Comments Program 15
7 Project Results 20
   7.1 Formative Evaluation During Development 21
      7.1.1 The Notes Program 21
      7.1.2 The Comments Program 22
   7.2 Formative Evaluation During Operation 22
      7.2.1 The Notes Program 22
      7.2.2 The Comments Program 23
   7.3 Plans for Continuation 24
8 Summary & Conclusions 24
9 References 37
1 Paragraph Summary

This project developed a process-based textbook and computer-based tools to help students learn to read and write arguments. The project used process-tracing methods to observe inexperienced and experienced writers at work and developed a model of the cognitive processes required to write original arguments. The project also used participant observation methods to observe a writing class as the teacher and students explored strategies for writing original arguments. These observations provided a rich source of data for identifying problems that learners encountered and the solutions that they and their teachers developed. The textbook, *The Architecture of Argument: Exploring Issues through Reading and Writing*, D. S. Kaufer, C. Geisler, & C. M. Neuwirth, (Harcourt, Brace & Jovanovich, in press), draws upon these observations to present a learner-centered curriculum, one based on learners' observed needs. The computer tools, Notes and Comments, were developed to help students and teachers overcome problems they encountered and to facilitate reading and writing processes.

Project Directors:

Preston Covey, CDEC
David S. Kaufer, English
Christine M. Neuwirth, English
Carnegie Mellon University
Pittsburgh, PA 15213
[412] 268-8702

Cheryl Geisler, Language, Literature & Communication
Rensselaer Polytechnic Institute
Troy, NY 12180
[518] 276-2724
2 Executive Summary

WARRANT:

A Learner-Centered Computer Environment for
Critical Reading, Reasoning, and Writing

Grantee Organization:

Carnegie Mellon University
Pittsburgh, PA 15213

Project Directors:

Preston Covey, CDEC
David S. Kaufer, English
Christine M. Neuwirth, English
Carnegie Mellon University
Pittsburgh, PA 15213
[412] 268-8702

Cheryl Geisler, Language, Literature & Communication
Rensselaer Polytechnic Institute
Troy, NY 12180
[518] 276-2724

Project Overview

This project developed a process-based textbook and computer-based tools to help students learn to read and write arguments. The project used process-tracing methods to observe inexperienced and experienced writers at work and developed a model of the cognitive processes required to write original arguments. The project also used participant observation methods to observe a writing class as the teacher and students explored strategies for writing original arguments. These observations provided a rich source of data for identifying problems that learners encountered and the solutions that they and their teachers developed. The textbook draws upon these observations to present a learner-centered curriculum, one based on learners' observed needs. The computer tools, Notes and Comments, were developed to help students and teachers overcome problems they encountered and to facilitate reading and writing processes.

Purpose

The WARRANT project began with one of the current consequences of the literacy crisis: College students, even at "better" institutions, do not write well. We argued that a cause of students' inability to write well goes beyond writing. It lies in their inability to read critically and, more deeply, in their inability to reason critically about what they have read or are writing about.

We argued that in order to address the problem, we needed to observe, close-hand, experienced and less experienced writers at work and to use those observations to build a curriculum that addresses learners needs. We argued that a computer-based curriculum, in conjunction with more traditional classroom instruction, would provide advantages for learning complex processes required for reading and writing arguments.
Background and Origins

The WARRANT project emerged in an environment in which there is a well-established record of excellence in research and educational computing and a firm commitment from the University for continued support. Moreover, the school has a long tradition of interdisciplinary effort of the type required in such a project. Members of the English department, for example, had already established productive working relationships with faculty in cognitive psychology, philosophy, and computer science.

Project Description

In the first two years, we collected and analyzed data on how experienced and inexperienced writers perform the following common yet crucial writing task: Read a group of authors who address a single issue and find something original to say about the issue. We also collected and analyzed data of a teacher's writing class in which the lectures, discussions and assignments addressed the same task.

Based on the data collection and analysis, we designed a curriculum that we and others believe makes a significant contribution to teaching students how to read critically, reason about what they are reading and write original essays that make a contribution to an issue. The reading/writing curriculum we developed focuses on the following processes that we identified from the data and analysis:

- Summarizing. Students learn to construct an author's line of argument and to produce a written summary of an author's position.
- Synthesizing. Students learn to design structures that allow them to compare the lines of argument of multiple authors. They learn to write a synthesis essay.
- Analyzing. Students learn to reach their own conclusions on an issue by exploring and evaluating possible positions. They learn to write an analysis of an issue.
- Contributing. Students learn to construct an original line of argument from their conclusions and to write an original essay that makes a contribution to the issue.

In parallel with the data analysis and curriculum development, we have developed two computer tools to aid writers engaged in reading and writing processes: Notes, a hypertext tool designed to help students take notes while reading and Comments, a collaborative tool for helping teachers and students respond to each others' texts.

Project Results


The course associated with the project, 76-122 Reading & Writing Arguments, continues to be taught. There will be three sections in the Fall, 1988. After the textbook appears, the department plans to make the course part of the core language, culture and history breadth requirement for all English majors.

The Notes and Comments tools are being used in sections of the following University core curriculum courses:

- 76-100 Strategies for Writing, the University Core course to introduce students to strategies for writing; required of all but Advanced Placement freshmen; In Fall 1988, 1/4 of the sections will be using our software tools.
- 76-101 Reading Texts, a Designated Writing course to introduce students to strategies for
reading literary and non-literary texts; one of several offerings for Advanced Placement freshmen; In Fall 1988, 1/5 of the sections will be using our tools.

- 76-122 Reading and Writing Arguments, a Designated Writing course. The course developed as part of WARRANT. In Fall 1988 there will be three sections, all using the software tools.

To facilitate use by instructors not directly involved with the project, we have produced a draft of a book, entitled *Reading and Writing with Andrew: An Instructor's Guide*. It will be distributed to all English faculty and graduate teaching fellows in the Fall 1988 semester. The book provides teachers with a guide to using our software and other Andrew facilities for teaching reading and writing. The computer programs have received some outside distribution. The University of California--Berkeley used Notes in a writing course for junior and senior engineers in Fall, 1987 and Spring, 1988.

The Notes and Comments programs are in the ICEC-ware Catalog/Collection. ICEC-ware is available to ICEC member institutions without charge. The ICEC central office provides members with "hot-line" and electronic-mail support for the following services: (1) installation of the software; (2) bug reports; (3) consultation with developers on other campuses who want to produce their own applications using concepts similar to those appearing in the catalog. ICEC-ware is available to non-members for a service fee of $150 for six months.

The Notes program can be used by any site that is running the Andrew interface software (7/86 release from Carnegie Mellon University). Notes runs on IBM RT-PCs, SUN-2s & -3s, and DEC VAX-stations. The Comments program can be used by any site that is running the Andrew interface software and the Andrew file system. The Comments program runs on IBM RT-PCs and DEC VAX-stations. Due to a technical problem, it does not run on SUN-2s & -3s, but it will by Summer, 1988.

Those who would like to obtain the programs should contact:
ICEC-Ware
CDEC Offices, Bldg. "B"
Carnegie Mellon University
Pittsburgh, PA 15213
[412] 268-7642
icec-support@andrew.cmu.edu (ARPA)
icec-support%andrew@cmccvb (BITNET)

The University continues to support the CECE staff required to maintain and enhance the software tools, to train new users and to answer users' questions. In addition, ICEC will provide support to member schools (and other schools for a small service fee) for people using the tools at other campuses.

Summary and Conclusions

Despite problems of less than perfect access for our students to the technology, we remain committed to the view that computer-based tools hold the potential to achieve revolutionary improvements for students learning to read and write--and for their teachers. We hope that our project has contributed one small step toward that revolution.
3 Project Overview

In the last decade, there has been a revolution in the study of writing. Part of that revolution can be characterized as an interest in studying writing as a cognitive process (Gregg & Steinberg, 1980). Research at Carnegie Mellon has contributed significantly to that revolution, resulting in a cognitive model of the composing process (Hayes & Flower, 1980; Flower & Hayes, 1980) as well as extended explorations of several of its subcomponents, in particular, planning (Flower & Hayes, 1981), translating (Kauf er, Hayes, & Flower, 1986) and revising (Hayes, et al., 1987).

The last decade has also seen a revolution in computing: the new advanced function workstation technologies have opened up unprecedented opportunities for new ways of representing information, including formatted text, outlines, node-link hypermedia, pictures, etc., and new ways for people to interact with those representations. Carnegie Mellon has played a leading role in that revolution.

The WARRANT project aimed at exploiting both revolutions to achieve an educationally innovative solution to a pressing problem: the lack of integration in instruction in reading, reasoning and writing. The project had the following goals:

(1) to use process-tracing methods pioneered at Carnegie Mellon to understand more deeply the cognitive processes required for students to write original essays on normative issues.

(2) to use participant observation methods to observe the social context of a teacher and his students as they explored and practiced strategies for writing original essays on normative issues.

(3) to combine both sources of observation to create a learner-centered curriculum. The observational data from (1) and (2) would provide a rich source for problems learners encounter and solutions that they and their teachers developed. Such data would allow us to build a curriculum based on learners’ actual needs.

(4) to use the advanced workstation technology to create advanced writing tools that would address learners’ needs.

4 Purpose

The WARRANT project began with one of the current consequences of the literacy crisis: College students, even at "better" institutions, do not write well. We argued that a cause of students’ inability to write well goes beyond writing. It lies in their inability to read critically and, more deeply, in their inability to reason critically about what they have read or are writing about.

The problem of literacy is hardly a writing problem alone. Skills of reading, writing and reasoning are highly interactive and interdependent. Reading skill is demonstrated and evaluated through writing. Writing skill is informed and inspired through reading. Reasoning skill is shaped and tempered through reading; demonstrated and evaluated through writing; presupposed and exercised in both. So, clearly, the problem of literacy is multi-dimensional, cross-disciplinary—and circular. Reading courses must be writing courses must be reasoning courses; and vice versa. This conception of the problem is commonsense if not commonplace. We argued that solutions require uncommon means.

Epidemic illiteracy and its causes aside, there are severe problems endemic to teaching critical reading, writing, or reasoning alone—let alone in some coordinated fashion. One is the problem of feedback: Students invariably need much more feedback on their reading, writing, and reasoning than teachers can
possibly anticipate—or deliver. Another is the problem of practice: Students can't simply be expected to commit a set of techniques to memory; they need repeated practice—practice applying these techniques, practice evaluating the results of these applications, and practice generating alternative strategies when a given technique seems not to apply. We cannot provide the kinds or amount of practice required—let alone feedback on it all—by usual means.

Worse, what feedback can be provided typically comes long after the student can most effectively make use of it, which would be—optimally—in the very process of reading, reasoning, or writing. Indeed, practice and feedback of the usual sort (evaluated assignments) are less appropriate for enhancing skills than certifying their presence or absence. Grades and comments are usually addressed to the qualities of some essay or examination, some product of the student's efforts. Those efforts themselves generally take place in untutored isolation, by a process the teacher can only guess but hardly help.

The alternative is clear, if difficult to offer: Some guidance in how to proceed, some clues about what to do, in the very process of reading, reasoning, or writing. If we are trying to teach judgment and skill in how to do something, we need to provide guidance in the very act doing of it—not merely commentary on the outcome or advice for prospective application.

Unfortunately, it is not humanly possible, much less cost-effective, for us to monitor, with any frequency, how students actually go about applying the techniques we give them—or to be there to suggest to them alternative paths when current ones fail. Typically, even when students have the opportunity to ask our advice (in an office or classroom) about how to proceed in some reading, reasoning, or writing task, there is typically a lag between their noting the advice and trying to apply it. By contrast, the advantages of immediate feedback and application are well known.

It was the difficulty of effectively teaching process skills—and the special difficulty and necessity of coordinating the teaching of critical reading, reasoning, and writing—that we wished especially to acknowledge in our approach. It was this difficulty that demands an uncommon approach, beyond the competence of unaided pedagogical savvy and traditional practice—an uncommon emphasis on the processes as well as the products of learning and application, and consequently, exploitation of novel research models and technology that are process- and learner-oriented.

5 Background and Origins

The WARRANT project emerged in an environment in which there is a well-established record of excellence in research and educational computing and a firm commitment from the University for continued support. Moreover, the school has a long tradition of interdisciplinary effort of the type required in such a project. Members of the English department, for example, had already established productive working relationships with faculty in cognitive psychology, philosophy, and computer science.

Carnegie Mellon University (CMU) is a small school (around 5000 undergraduates, 2000 graduate students, 550 faculty, 2000 staff) that has chosen to invest heavily in computing.

The locus for the WARRANT project was CMU's Center for Educational Computing in English, established through an administrative initiative from the Dean of the College of Humanities & Social Sciences in 1983. Like other Centers on campus, the Center for Educational Computing in English (CECE) is funded primarily through outside grants and contracts. The University provides space and utilities, and one of the co-directors of the Center occupies a faculty line specifically allocated to pursue
initiatives in educational computing. Others on campus (e.g., the Vice Provost for University Studies, the Director of the Center for the Development of Educational Computing, the Head of the Department of English, the Dean of the College of H&SS) provide support for CECE activities that are consonant with their organizations' goals. The organizational support for the WARRANT project has been tremendous. For example, the WARRANT project has received space priority within the college, equipment support, programming support, and direct financial support from these other elements in the University organization.

Although the WARRANT project's outside funding has ended, the University continues to support the CECE staff required to maintain and enhance the software tools, to train new users and to answer users' questions.

6 Project Description

6.1 Original Plans

The WARRANT project was proposed to push educational frontiers on two independent, but highly interactive, levels:

Observation-based curriculum development. We originally proposed, using verbal protocols, to track expert and novice writers as they worked through a corpus of readings on a single issue and wrote an original essay on that issue. This data would be used to develop a curriculum for teaching writing/reading/reasoning as interactive processes.

Computer-based curriculum delivery. Our original proposal also called for the development of software to deliver the curriculum.

6.2 Primary Actions

6.2.1 Data collection and analysis

We decided on a three-part approach to gathering data for the curriculum. Two of these foci were part of our original proposal: (1) gathering thinking-aloud protocols from novice writers; (2) gathering thinking-aloud protocols from experts writers. These were to provide the basis for an analysis of the cognitive processes involved in writing original arguments from source texts. In addition, we added a third focus: (3) gathering data from a teacher and students in a course in which students were learning to read arguments, reason about them, and write an original response. In conjunction with the third focus, we expanded our methodology to include, in addition to think-aloud protocols, elite interviews and participant observation. The addition of this focus allowed us to observe an experienced teacher modifying and elaborating strategies for teaching reading and writing processes, and to talk with students about their problems and perceptions.

6.2.2 Curricular development

We tied our curricular goals very closely to William Perry's work. Briefly put, Perry’s work has shown that intellectual and ethical development in the post-secondary years is marked by the increasing ability of students to negotiate multiple authorities and, eventually, to learn how to situate themselves among those authorities. The reading/writing curriculum we developed seeks to support this development by focusing on the following skills that we identified from the data analysis:
• Summarizing. Students learn to construct an author's line of argument and to produce a written summary of an author's position.

• Synthesizing. Students learn to design structures that allow them to compare the lines of argument of multiple authors. They learn to write a synthesis essay.

• Analyzing. Students learn to reach their own conclusions on an issue by exploring and evaluating possible positions. They learn to write an analysis of an issue.

• Contributing. Students learn to construct an original line of argument from their conclusions and to write an original essay that makes a contribution to the issue.

The curriculum represents one of the major efforts of the project. We have included a copy of the textbook that we wrote, *The Architecture of Argument: Exploring Issues through Reading and Writing*.

6.2.3 Computer system development

In the initial specification for the computer system, we divided the user interface into six major components:

• A Syllabus. The component students would use to access the WARRANT curriculum.
• A Read Text window. The component students would use to read on-line text.
• A Plan window. The component students would use to access hierarchical plans for critical reading, writing and reasoning.
• An Advice window. The component students would use to access more detailed descriptions of plans, including elaborations, strategies, process and product models, and help.
• A Note Card Component. The component students would use to take notes and organize them.
• A Compose Text Window. The component students would use to compose papers.
• A Comment Component. The component students and teachers would use to comment on each others texts and read those comments.

We experimented with prototype Plan and Advice components, but decided against using the computer primarily as a vehicle for the delivery of instruction. Observation showed the computer to be an unwieldy vehicle for instruction in processes as complex as those involved in reading and writing. We found that teachers and students wanted to work on the task itself, not read about how other people did the task. We decided that the classroom/textbook would be the primary vehicle for instruction.

Thus, our computer system development focused on two computer tools that we hoped would help students overcome some of the difficulties that we had observed. These tools, Notes and Comments, incorporate the Note Card, Comment, Read Text and Compose Text windows of our original plans.

6.3 The Computer Tools

6.3.1 The Notes Program

Objectives

Notes is a hypertext application (Conklin, 1987) developed to investigate the effects of computers on the writing process, in particular, on the processes of acquiring and structuring knowledge when writing from source texts. Specifically, Notes is designed to provide students with procedural facilitation (Bereiter & Scardamalia, 1987) for the following activities:
• recording their own ideas (e.g., reactions, inferences, plausibility assessments) while reading,
• recovering the context for those ideas easily, and
• viewing ideas from multiple perspectives.

Theory

The Notes program addresses procedural content, that is, 'knowing how to' rather than 'knowing that.' As such, the design was informed by what is known in the field of writing about writing processes.

Theories of writing processes typically identify the following activities in writing: acquiring knowledge, viewing it from different perspectives to gain new insights, structuring knowledge according to those perspectives, selecting and possibly creating knowledge to meet goals for discourse and rearranging it so that readers with different perspectives will find it equally coherent (Young, 1971). The following explores each of these activities and comments on the use of the Notes program as a technology that can aid a writer in carrying out these activities.

Acquiring knowledge. Many of the ideas that we ultimately make use of in a text come to us while acquiring knowledge, that is, while exploring a problem and finding out more about it. Typically, ideas in a new domain do not come to us in an orderly fashion. Rather, they present a puzzle of seemingly unrelated concepts and unexplained connections. It is difficult to remember specific ideas that we learn. We often restructure ideas to fit patterns that are already familiar or drop ideas that are difficult to assimilate to familiar patterns (Bartlett, 1932).

As we read or find out new information, we are not simply recording it. We are constructing connections, drawing inferences, imagining scenarios and examples, commenting on plausibility, noting connections to other texts and knowledge as well as connections to our immediate goal and the problem we are investigating. These elaborations play an important role in acquiring new knowledge. Researchers postulate that elaborations play two vital functions: They form connections between what people already know and the new knowledge and they build multiple retrieval paths for the ideas (Reder, 1979). While it is important when reading to construct elaborations and inferences, it is equally important when writing to remember that they are elaborations and inferences, and not to confuse them with the original information.

Viewing knowledge from different perspectives. The second activity usually included in writing, especially by a theory that includes invention, involves viewing knowledge from different perspectives. Some invention theories involve explicitly teaching writers a set of perspectives; for example, Aristotle’s topoi, Young, Becker and Pike’s tagmemic grid (particle, wave, field), Burke’s pentad (act, scene, agency, purpose, etc.) or Nelson’s system of synectics. Each of these techniques provides a system for exploring concepts, an activity essential to discovering new elaborations or relationships. Most such theories stress the importance of systematically varying perspectives, a way to overcome Burke’s observation that "A way of seeing is also a way of not seeing." Indeed, studies which have examined creativity in writing have noted a direct relationship between the amount of examination of concepts from different perspectives and quality of writing and creativity (Young & Koen, 1973; Moore, 1985).

Structuring knowledge. Different perspectives also provide frameworks for structuring knowledge. Few studies have examined the process of writing while the writer is acquiring domain knowledge. Those few studies that do exist support the notion that structuring knowledge can be a significant task in writing in new domains. Newell’s study (1984), which examined the role of writing in learning, found
that writing about a new domain required writers to move from relatively isolated and detached concepts to an integrated structure. Langer's (Langer, 1984) study of the relationship between topic-specific knowledge and quality in expository writing suggests that the degree of organization of knowledge is directly related to a writer's success. Writers whose knowledge was highly organized, i.e., their knowledge base included superordinate concepts, precise meanings, analogies to other concepts, and explicit links among concepts, were most successful.

Selecting and arranging. At some point in the process of writing, the writer must decide what knowledge, both acquired and original, is going to be suitable for communicating to a reader. Moreover, the writer must decide what linear order for the ideas--what juxtapositions and connections as well as oppositions--will result in best meeting the writer's goals. Writing an essay requires shaping a complex network of ideas, not all of which are present at the beginning of the writing process, into a coherent linear structure of sentences and paragraphs. For this linear structure to be successful, the writer must have constructed systematic conceptual groupings among ideas (Meyer, 1975). When a writer knows a domain well, relatively simple reordering of available knowledge may be all that's necessary. However, when a domain is new to the writer, or the writer is reconceptualizing a well-known domain, the writer may need to engage in extensive reorganization and elaboration of his or her own understanding.

The Notes program attempts to provide procedural facilitation with aspects of each of these activities. The program is designed to allow students to perform the activities recursively (Hayes, 1980).

Needs of learners addressed

Students have tremendous difficulties with writing from source texts. For example, in a study of Freshmen students' ability to write summaries--arguably the easiest writing from sources task--only 25% of mainstream Freshmen included 2/3 of the idea units included by experts, and only 6% of the underprepared students managed to include 2/3 (Johns, 1985). In the same study, Johns reports that Freshman students introduced distortions at the idea-unit level a little over 50% of the time; approximately 10% distortions in statements that integrated two or more ideas units; and 30% in statements that make a general comment about their reading.

Our own research also highlights the difficulties students have in writing from source texts (Neuwirth & Kaufer, 1987). To examine the processes involved in writing original arguments from source texts, we collected thinking-aloud protocols from people at two skill levels, expert and novice. Five of the subjects were experienced writers. They included three professors of philosophy and two Ph.D. students. Two subjects were novices. Both were undergraduate students recruited from a writing class. We asked subjects to read eight articles on the issue of paternalism and to write an essay that made an original contribution to the issue. Writers wrote at their own pace, taking up to fifty hours to complete the task.

This writing from sources tasks was sufficiently difficult that we observed even the experienced, highly skilled writers

- focused on details at the expense of large goals and attended to information that played no role in later writing
- forgot information that would have been useful
- needed to search for information that they could only partially retrieve
- selected incorrect operators and paths and needed to backtrack
- engaged in trial search
• lost track of goals

The Notes program was designed to reduce a subset of these difficulties.

Contribution to the discipline

The Notes program extends previous lines of research in three ways. First, it focuses research attention on two subcomponents of the writing process that have yet to receive extensive examination: the writer’s memory and the external task environment. Second, in doing research necessary for building the program, we extended the process model of writing to a new task, one that is central to the process of research: writing arguments from source texts. Finally, by tapping the computer’s potential to create new, theory-based interactive representations and new frameworks for studying how users interact with them, we have increased our basic knowledge of how to design interactive systems that support complex information processing tasks. For a discussion of the Notes program’s relationship to previous programs, see (Neuwirth, et al., 1987).

Intended student audience

Our immediate audience was students in our Freshman writing courses at Carnegie Mellon. These students are highly motivated, high ability students. The writing course is required of all incoming Freshmen in all colleges: engineering, science, fine arts (e.g., drama, painting), humanities and social sciences. The students do not typically require much instruction in grammar or mechanics; the Freshman writing courses focus on the processes of writing and strategies for producing particular types of writing. Students write two or three substantial papers (10-20 pages). Many of the students, especially those in engineering and fine arts, have relatively low interest in learning to write.

All entering CMU students take a Computing Skills Workshop that teaches three computer systems, including Andrew, the system on which Notes is implemented. Thus, our documentation assumes that students are familiar with computers generally and with Andrew in particular.

As a tool, the Notes program will not benefit students who lack the knowledge, ability or motivation to use it appropriately. We believe that students need instructional support in addition to the tool in order to use the tool to learn effectively. We consider it inappropriate to consider the effectiveness of the tool apart from instructional strategies which attempt to exploit and amplify its advantages.

Although we have no direct evidence, it may be that the Notes program may have different potentials for students below certain age groups or of differing abilities. There is some evidence that students have to be at least in Grade 10 to be able to notice and take advantage of conceptual coherence in elements in order to form groups that assist memory and recall (Frankel, 1983). In another study that bears indirectly on the Notes program, low ability students benefited more from diagramming texts than high-ability students, perhaps because the high-ability students did not need extra support (Geva, 1983).

Contributions to students

The contributions this software tries to make to the needs of students are the following:

• In learning to write from source texts, students are unnecessarily burdened by the book-keeping involved. By automating much of the book-keeping and allowing students to recover the context for any note automatically, we expect the Notes program to make it easier for students to concentrate on recording their own ideas.

• Synthesizing a body of research literature is a daunting organizational task. By allowing students to form classes by which to group their notes and to create alternative organizations, we hope to make the task easier.
There are many ways to "write" ideas down, to record the connections between them, to juxtapose ideas, perhaps discovering new connections: pencil and paper, 3x5 note cards, tape recorders, text-editors, etc. Some of these are better than others for aiding the processes of invention and arrangement just outlined. This section explores the benefits of note cards for carrying out some of these activities.

Writers use note cards for three primary reasons: First, note cards provide an external store for a large body of knowledge that as yet has no coherent linear structure. Second, note cards provide a convenient way for writers to record their own reactions, elaborations, and interpretations of texts while still maintaining a record of sources that the writer may want to return to or to acknowledge. Third, note cards provide a way of representing knowledge that makes some inventive activities easier.

The first benefit to writers using note cards is that they provide a convenient way to record ideas in a text. As noted above, recording concepts and propositions is particularly important when there might be a tendency to fit new knowledge to familiar but inappropriate patterns.

The second benefit to writers using note cards is that note cards give them a convenient way of recording their own reactions, elaborations, and interpretations of texts that they are reading while still maintaining a record of the source. By recording the source together with the elaboration, note cards make both available for review and reevaluation. The importance of review and reevaluation in learning a new domain has been cited by writing researchers as a reason that writing has a major role to play in learning (Emig, 1971).

Various studies that are relevant to taking notes have explored the strategic significance of elaborations during reading. A study of elaborations during reading in which the elaborations are written down rather than unwritten (mental or verbalized) found that written responses led to better post-test responses than unwritten (Michael & Maccoby, 1961).

The third and most distinctive benefit for note cards is their power as a representational medium. A given network of ideas can be represented by a number of different structures, some of which are better than others for enabling a person to work. For example, numbers are usually better represented with Arabic than with Roman numerals. Likewise, the various structures that are encouraged by the use of note cards are better than an initial, relatively fixed, linear structure when a person needs to seek out relationships among ideas. Note cards facilitate alternative representations for the linear structuring of concepts, allowing writers to experiment with tentative arrangements until the writer discovers or can impose a workable framework.

The previous section argued that note cards provide a better representational system for writers working in new domains than linear structuring: note cards facilitate a writer's exploration for alternative structures of ideas. Despite this advantage, however, conventional note cards have disadvantages. Not infrequently, writers forget the context for the original note, and must return to the source material in order to make sense of the content of the card. A similar problem occurs with paraphrasing in notes: the writer introduces inaccuracies. Writers, especially inexperienced ones, tend to spend all their time writing down quotes from the source texts rather than recording paraphrases, elaborations, inferences, interpretations, etc.

The foremost problem with note cards arises when the writer is struggling to impose a workable framework on the material: Although notes offer a more tractable medium for this activity than 8x11
paper, creating alternative frameworks nevertheless destroys the previous order. Writers have two alternatives to circumvent this problem. First, they can make duplicates of note cards, a time-consuming venture. Second, they can number note cards and then record the structuring by means of the numbers. Reconstructing the ordering is then possible, but like the duplication solution, also time-consuming.

When a writer is working with texts that are stored in the computer, the Notes program keeps a link between each note and the specific region in the source text from which it came. We reasoned that such a facility would free the writers (1) to paraphrase because they would always be able to easily recover the quotation, and (2) to record their own elaborations, reactions, inferences, etc., because they could easily recover the context for them.

Recovery of context is only easily accomplished when the texts are stored in the computer. Although this is possible in a writing course in which the number of readings is small, it will be a number of years before we see vast numbers of texts stored on computers. Thus, the primary benefit of computer-based notes in the near future will be its potential for helping writers create alternative organizational frameworks more easily. Unlike paper, the computer does not collapse the storage and display of information. Because of this feature, the computer can be easily programmed to allow writers to create and view alternative organizations of their notes. Creating new alternatives does not destroy previous organizations and the computer can easily keep track of the book-keeping involved.

Description of the Software

The Notes program consists of two basic objects, source texts and notes, and a single derived object, lists of notes. Source texts are those texts the user is reading and wants to take notes on. The source texts can be online or off, but the following discussion illustrates a user taking notes on an online text. Notes are those texts the user composes in order to record elaborations of the source texts, i.e., the user’s record of his or her "writing" of the text. Notes are online.

In addition to the basic objects, the Notes program consists of a single derived object: lists of notes. In the current version of the Notes program, the lists are automatically compiled. Lists have a linear order, alphabetically by the author of the source text and within sources, by the user-created name of each note. The user can also create alternative lists, typically based on ordering principles that the Notes program cannot compute automatically. The alternative lists allow the user to impose a hierarchical structure on the notes as well.

Figure 1 illustrates how the screen might appear to a user who is in the midst of reading on the topic of creativity. The user has taken notes on two source texts: one by Hayes and one by Perkins. The system maintains a list of all the notes a user has taken in the region labelled All Notes List. At this point and at any point, the user can select from a range of activities: view the notes, create classes and classify the notes, form alternative organizations for the notes, etc. The user controls the order of these activities. Let’s suppose that the user wants to take more notes on one of the source texts, Hayes, "What is a creative act?" To do so, the user opens a set of menus and uses a mouse to select Open from a Source Text menu card.

Taking Notes. To take a note, the user selects the region in the source text where he or she wants to take a note, moves the mouse cursor anywhere in the selected region, opens a menu, and chooses Take Note from the pop-up menu (see Figure 2).
Composing a Note. When a user chooses Take Note, a note region appears below the source text. The source text itself is recentered, if necessary, so that the selected region for the note remains visible on the screen. An icon appears in the source text. The icon looks like a footnote in a square and indicates that there is a link between the source text and the note (see Figure 3).

To compose a note, the user moves the mouse cursor inside the note region, clicks the left mouse button and begins composing ("Why is it important...?"). The Notes program uses the Andrew system base editor, so the user has the full functionality of an integrated text-editor/document-formatter to compose. In addition, the user can copy material from the source text or from other windows on the screen and paste it into the note.

Although the note region approximates a 3 X 5 card, the text of the note can be as long as the user desires. If the text that the user composes exceeds the space allocated to a note region, the entire text will not be visible. However, the user can scroll the text to view different parts of it or enlarge the Notes program window so that more text is visible.

In addition to composing the text of the note, the user must also compose a name for each note ("Why Criteria?"). A name is a mnemonic for the contents of the card, and is used by the Notes program to display a list of notes that have been composed.

To take another note, the user selects a region of text and chooses Take Note again. The previous note is replaced by a blank note and except for the name of the note, which is put into the Notes listings, the previous note is "put away" from view.

Viewing Notes. After the user has taken a number of notes, perhaps in a different session, he or she may wish to review the notes. To view notes, the user positions the mouse cursor in the All Notes List, points at a note of interest and clicks the left mouse button. The note appears in the View Notes region (see Figure 4).

The user can display up to four notes at a time. In addition, the user can ask the program to expand the viewing region so that more notes can be viewed. When the viewer has been viewing a series of notes and calls up a new note, the new note will appear in place of the note that has been dormant the longest.

Alternative Lists. In addition to viewing notes from the All Notes List or from the source text, the user can also create alternatively organized lists of notes, called alternative lists. Alternative lists support viewing notes from alternative perspectives. Users can create as many alternative arrangements as they need. They can cut and paste across different lists. In addition, they can display different alternatives on the screen and compare them.

Classifying Notes. Classes play an important role in the Notes program: classes allow users to group notes together. For example, while taking notes or after, a user may group notes according to classes that he or she creates. The classes might be related to the content or structure of the source texts, or to the nature of the elaborations that the user has composed. Figure 4 shows three classes: Original, Value, and Ability, located in the region labelled Classes at the top of the screen.

To create a class, the user displays the classes by means of a menu and chooses Add a New Class from the Edit Classes menu. There are also options to Delete a class or to Rename a class. Because deletion affects notes which might be in the specified class, the user is first informed of the number of notes which are in the class and asked to confirm or cancel the deletion. If the user responds Confirm, the class is deleted; the notes in the class are not deleted, but only removed from the class.
To add a note to an already existing class, the user makes the note the current note and clicks on its class name. The class name highlights to indicate the current note is a member of the class. Notes can be added to as many classes as the user desires. To delete a note from a class, the user makes the note the current note, and clicks on the class name. The class box is de-highlighted to indicate that the note has been deleted from the class.

Searching. The user can search for notes on the basis of content, classes, the author of the source text, the title of the source text, the date and time the note was created, and the date and time that the note was last modified. These facilities allow users to locate notes automatically. For example, if the user had taken notes on two source texts on creativity, one by Hayes and the other by Perkins, and classified several of the notes in a user-created class of definition, the user could search for all the notes by Hayes or Perkins that are in the class definition. The search results in a listing of those notes appearing on the screen. The user can view the contents of particular cards in the search result in the same way as any list of notes.

Implementation. The Notes program runs on advanced function workstations—IBM RTs, SUN2s & 3s, and VAXstations. It runs under Andrew, a window-management and base environment for UNIX 4.2 BSD (Morris, 1986).

6.3.2 The Comments Program

Objectives

Comments is a computer tool developed in order to study computer support for response to writing. Writers and readers can use the Comments program to "talk" about a piece of writing over our campus-wide network of advanced-function workstations. The program is intended to make it easier for students to perform the following tasks:

- share their work in progress with other members of the class, friends, and their teachers
- ask for clarification of written comments
- share plans for revision based on written comments
- communicate about whether a draft has addressed previous comments

Theory

There are two primary theoretical perspectives that support the use of response to writing in learning to write. The first emphasizes writing as a social act (Bruffee, 1973, 1985; Gere, 1987). By interacting with each other, teachers and students can engage in a dialogic/dialectic that stimulates invention (LeFevre, 1987). A dialogic activity engages writers in a conversation with readers and other writers, primarily to become aware of views other than their own. A dialectic activity engages writers and readers in a search for agreement. Such a search can sometimes lead a writer to reassess the validity of his or her own point of view. Both dialogic and dialectic responses can provide powerful motivation for a writer to elaborate or justify his or her opinions.

The second perspective emphasizes that writing is a skill whose acquisition requires knowledge of outcomes, and that knowledge of outcomes is best given by a group of readers (students and teacher) rather than only one reader (the teacher). More than one reader more closely simulates actual communicative situations in which writers must grapple with conflicting readings. In a meta-analysis of pre-test to post-test effect sizes, Hillocks (1986) concludes that a "combination of peer and teacher feedback is consistently somewhat stronger than only teacher feedback..."
The Comments program is intended to support both these theoretical perspectives by providing a communication-support system for writers and readers, students and teachers.

Needs of learners addressed

Despite the strong theoretical reasons for believing that response to writing is helpful, research results are somewhat mixed (DiPardo and Freedman, 1987; Freedman, et al., 1986). Detailed observational case studies (Berkenkotter, 1984; Ziv, 1984) offer compelling evidence that students' failures to benefit from peer and teacher feedback often stem from failures to communicate effectively about that feedback. For example, many students who fail to understand a comment on their writing do not see it as an opportunity to communicate with their teachers or peer group; rather, they reject the comment out of hand or—also not uncommonly—accept it despite an unresolved misunderstanding/disagreement.

Obviously, students' views of comments will affect how they read them. Students often don't understand their teachers' comments (Butler, 1980; Hahn, 1981; Schwartz, 1983). Students often think that teachers' comments reflect teachers' confused readings rather than their own confused writings and so discount the value of the comments (Hahn, 1981). Students are frustrated by what they perceive as a lack of consistency in comments (Summers, 1982) or when teachers' and peers' comments contradict each other (Ziv, 1984).

Because of the theoretical perspective and research findings outlined, in developing tools and methodologies for computer-based support for response to writing we concentrated on developing ways to facilitate communication. For example, the Comments program is designed to be primarily a communication support system. The program exploits the campus-wide personal computer network to enable students and teachers to "talk" more easily about each others' responses to writing. This design decision assumes that students' confusions and difficulties are an inherent part of the communication process. By providing a system that facilitates communication, we hope to provide a system that encourages students to represent the act of reading comments as an interpretive act, on a par with the interpretations required in reading any text.

Contribution to the discipline

The Comments program's contribution to the discipline consists in its hypertext implementation (Conklin, 1987) of support for a social interaction view of response to student writing.

Intended student audience

Perhaps even more than with the Notes program, we believe that the Comments program will not benefit students who lack the knowledge, ability or motivation to use it appropriately. We believe that students need instructional support in addition to the tool in order to use the tool to learn effectively. For example, members of CECE (Center for Educational Computing in English) meet on a weekly basis with instructors using the Comments program to discuss theories of response to writing, ways the Comments program can implement those theories, and ways of communicating with students about the role of response in learning to write and the social/cognitive skills that students/teachers need to exercise in order to interact supportively. We consider it inappropriate to consider the effectiveness of the tool apart from instructional strategies which attempt to exploit and amplify its advantages.

Contributions to students

We hope that the Comments program meets the following student needs:

- Need for timely response. Teachers and students using the Comments program report that
they are able to give more timely response. For example, some teachers encourage students to submit a draft when they need a response. These teachers typically do not respond to every aspect of a student's draft; rather they report taking a few minutes to skim a paper that a student submits, focusing their response on some aspect of the student's draft, then returning the draft.

- Need for sharing work quickly and easily. Students submit drafts to other students and their teacher when they most want feedback. Students do not need to wait until class to exchange papers easily. They do not need to hunt for physical mailboxes, something few students are willing to do, especially for off-campus addresses.

- Need to make sense of unclear, inconsistent, or other problematic comments. When students encounter problematic responses, students can easily send the paper on to another reader; or they can use the dialog capabilities of the program to ask the first reader for a clarification.

- Need to know whether the next draft has addressed comments. Students can submit a revision with comments and "discuss" the revisions using the Comments program.

Use of computers

Pen and paper technology has several drawbacks that computer technology potentially can overcome. First, marginal comments, because of lack of space, are often mere phrases, short questions or brief statements. Rarely are they longer than one or two sentences. Often they do not indicate the exact location of a problem because doing so would interfere with the legibility of the student's original text. They are often barely legible. Terminal comments, on the other hand, do not indicate the exact location of a problem, though they can be longer. Although it is possible to adopt a system that combines the advantages of both (e.g., some teachers put an endnote number at the location of each feature in the text and write more detailed comments at the end), computer technology facilitates the bookkeeping involved in such a combination system (e.g., if a teacher numbers an endnote 1 and, at a later point, another 2, the teacher is forced to number as 1.a, 1.b, etc. any comments added later that fall between 1 and 2; however, the computer can keep track of numbering automatically. Indeed, the Comments program does this when printing comments).

Some students report feeling as though the integrity of their paper is reduced when handwritten comments appear throughout the entire text. Computer technology can help preserve integrity, by allowing commenters to place a small, relatively unobtrusive icon within the text to represent an area of interest. Connected to this icon is the comment, which is composed and viewed in a separate panel, beneath the entire text of the paper. This comment region provides unlimited space for a reader to record his or her thoughts, allowing for more detailed comments than in typical pen & paper situations. Because computers do not collapse the storage and display of information, the icons can be hidden altogether, allowing the student see the text 'as-if' there were no comments whatsoever.

Computers can potentially support the multiple purposes teachers have in responding to students' papers. Teachers have at least four ways of thinking about comments on student papers (Purves, 1984): (1) the diagnosis of students' writing problems, either problems with the product or process problems inferred from the product; (2) teacher-assisted revision, in which the teacher helps the student revise by detecting error, diagnosing error, or correcting the error; (3) reader-response in which the teachers' comments emulate questions that a colleague/reader might have about the text; and (4) the justification of an evaluation, typically a grade.

In pen & paper technology, it is usually not possible for teachers to fulfill all these goals simultaneously without the labor and expense of xeroxed copies. For example, in diagnosing a problem, teachers often
use comments to write down hypotheses about what the problem is, how the student should fix it, etc., only to change course later on and decide that another hypothesis is warranted. Because teachers typically write these intermediate hypotheses on their only copy of the text, students often see these, or the scratched out remnants of them. Likewise, in justifying an evaluation, teachers often wish to make comments that correspond to the entire set of evaluative criteria. In contrast, in teacher-assisted revision, the teacher typically wants to establish an agenda for instruction—to pick out some subset from the manifold possibilities in order to focus the student’s concerted attention. Although not the focus of our research interest, electronic copies potentially facilitate the management of the multiple function of comments.

As noted in previous sections, we believe one of the primary advantages of computer technology over pen and paper technology will be its capability to facilitate students and teachers working together to respond to writing. A campus-wide network frees students from often frustrating activities of scheduling meetings with their instructors or peers to discuss a paper. We do not mean to imply that students and instructors should not meet face-to-face—either in class or outside. Simply that the computer technology provides an additional mode of communication—one that our users seem to judge to be a welcome middle ground between face-to-face meetings and hard copy written comments.

We hypothesize that the increased ease and efficiency of arranging to "talk" about papers results in more communication between student writers and readers; in turn, more communication may produce more effective writing by students. The ability to comment on a comment also encourages communication, as students are able to respond immediately to comments they have received, by requesting further clarification or expressing their disagreement. This ability to discuss comments may in turn yield greater understanding by students, as well as increase their knowledge.

Pen and paper retains some advantages over computer technology. For example, people report a difficult time "getting a sense" of texts that are displayed online, even on large-screen (19x19), high resolution bit-mapped displays (Haas & Hayes, 1986). For this reason, we recommend to students and teachers that they do not try to use the Comments program to replace hard copy, but to use it in a way that exploits the advantages of the various modes of communication (i.e., face-to-face, hard copy, computer-based comments, etc.).

Description of the software

Integration with the campus-wide network. As noted earlier, the Comments program is integrated with our campus-wide local area network. A computer network is a system of independent computers (today, typically personal computers or workstations) that communicate with one another and allow users to share computer resources such as hardware (e.g., laser printers), programs (e.g., large statistical packages), or files (e.g., essays). Our campus' local area network allows students and teachers to access their files from any workstation on campus. For example, if a teacher sends a message to a student, the student can go to any public workstation on campus and read the message.

Figure 6 depicts the Comments start-up screen, which displays the major network communication options: Send a paper and Read a paper. Other options include Retrieving a Paper (which retrieves a paper, typically from someone who has failed to return it); Delete Comments (which deletes the comments on a paper, typically when the user is done with a draft); Help (the online help), etc.

Sending a paper. When a user chooses to send a paper, the Comments program uses the campus-wide network to send the paper to the people who will be responding. Users can send their own papers, or they
can send along another person’s paper, analogous to passing along a hard copy paper to the next person on a distribution list. There is no limit to the number of users who can be sent a paper (the students and teachers, however, have organized themselves into groups of two to five). The recipients of the paper are notified by electronic mail messages that the user would like them to comment on a paper.

Making comments. To comment on a paper, the user chooses Read a Paper and types in the name of the person whose paper he or she wants to read. Normally, the commenter can read the paper, but cannot change it except to add comments (N.B. The comments are stored separately and do not actually change the file containing the paper).

The user can make comments that are tied to specific regions of text or that apply to a location in the text (e.g., a global comment at the beginning of a paper). To make a comment, the user points to a location in the text or selects a region of text, opens a menu and chooses Make Comment from the pop-up menu (see Figure 7).

Composing a comment. When the user chooses Make Comment, a comment region appears below the text. The text itself is recentered, if necessary, so that the selected region for the comment remains visible on the screen. An icon appears in the text. The icon looks like a triangle and indicates that there is a link between the text and the comment (see Figure 8).

To compose a comment, the user moves the mouse cursor inside the comment region, clicks the left mouse button and begins composing. The Comments program uses the Andrew system base editor, so the user has the full functionality of an integrated text-editor/document-formatter to compose. In addition, the user can copy material from other texts in other windows and paste it into the comment or the user can insert text that is stored in files.

Although the comment region approximates a 3 X 5 card, the text of the comment can be as long as the user desires. If the text that the user composes exceeds the space allocated to a comment region, the entire text will not be visible. However, the user can scroll the text to view different parts of it or enlarge the Comments program window so that more text is visible.

Multiple comments and comments-on-comments. To make another comment, the user selects a region of text and chooses Make Comment again. The previous comment is replaced by a blank comment region. The user can make as many comments as desired. The user can comment on other comments (perhaps comments made by the author or other readers), by pointing to a location within the comment and choosing Make Comment. Figure 9 depicts a screen showing comments on comments. This is the facility that allows writers and readers to "talk" about a text.

The user can make comments at different sessions at the workstation. When the user is done commenting, choosing Send a Paper allows the user to return the paper to the author or pass it along to another person for comments. The author or other person is again notified via electronic mail.

Viewing comments. The program provides four options for viewing comments. The user can find the next comment in a text by choosing Next Comment from the pop-up menus; the user can find the next new comment (i.e., a comment that he or she has not seen before) by choosing Next New Comment; the user can scan the text for comment icons and point and click on one of the icons to view a comment; finally, the user can point and click on one of the comments in the comment chain, represented on the right of the screen (see Figure 9). The comment chain represents a chain of comments on comments.
Revising with comments. The author of the paper can revise it from within the Comments program. There are options for deleting comments as well as text. The user has the full functionality of an integrated text-editor/document formatter.

Establishing a dialog. If the author does not understand or disagrees with a comment, the author can choose Make Comment, ask for elaboration, then choose Send a Paper to send the paper back, continuing a dialog with his or her readers.

Implementation. The Comments program runs on advanced function workstations—IBM RTs and VAXstations. It runs under Andrew, a window-management and base environment for UNIX 4.2 BSD (Morris, 1986). It is also dependent on the Andrew ITC file system.

7 Project Results
In the original proposal, we planned to achieve four major results within the duration of the project:

• Goal 1: The WARRANT computer system would be operational and have undergone a major formative evaluation on one reading/writing task in the domain of normative argument.
  • Result 1: The entire system has been operational for a semester and has undergone a major formative evaluation, described in detail below.

• Goal 2: The completion of a textbook for students on goals and strategies that are effective for reasoning about normative issues.
  • Result 2: The textbook, entitled The Architecture of Argument: Exploring Issues through Reading and Writing, David S. Kaufer, Cheryl Geisler, and Christine M. Neuwirth, is completed and will be coming out from Harcourt, Brace & Jovanovich in January, 1989. We have enclosed a manuscript copy with this report.

• Goal 3: A book for teachers that describes how to use WARRANT as a tool for different pedagogies in reading and writing.
  • Result 3: A draft of the book, entitled Reading and Writing with Andrew: An Instructor’s Guide, is completed and will be distributed to all English faculty and graduate teaching fellows in the Fall 1988 semester. (Andrew is the name of the CMU computing system on which the WARRANT software runs). We have enclosed a manuscript copy with this report.

• Goal 4: The technology underlying WARRANT (if not exactly the WARRANT system itself) will be readily available to the colleges and universities that are members of the Inter-University Consortium for Educational Computing (ICEC).
  • Result 4: The WARRANT system software is in the ICEC-ware Catalog/Collection. ICEC-ware is available to ICEC member institutions without charge. The Notes and Comments programs are listed in the catalog as "can be freely used for educational purposes but not copied or distributed for commercial purposes." The ICEC central office provides members with "hot-line" and electronic-mail support for the following services: (1) installation of the software; (2) bug reports; (3) consultation with developers on other campuses who want to produce their own applications using concepts similar to those appearing in the catalog. ICEC-ware is available to non-members for a service fee of $150 for six months.

The Notes program can be used by any site that is running the Andrew interface software (7/86 release from Carnegie Mellon University). Notes runs on IBM RT-PCs, SUN-2s & -3s, and DEC VAX-stations. The Comments program can be used by any site that is running the Andrew interface software and the Andrew file system. The Comments program runs on IBM RT-PCs and DEC VAX-stations. Due to a technical problem, it does not run on SUN-2s & -3s, but it will by Summer, 1988.
Those who would like to obtain the programs should contact:

ICEC-Ware
CDEC Offices, Bldg. "B"
Carnegie Mellon University
Pittsburgh, PA 15213
[412] 268-7642
icec-support@andrew.cmu.edu (ARPA)
icec-support%andrew@cmccvb (BITNET)

7.1 Formative evaluation during development

7.1.1 The Notes Program

Throughout its development, we have been conducting formative evaluations of the Notes program, where by "formative evaluation" we mean a study that attempts to evaluate a program in order to improve it.

Participants. The participants in the evaluation have been experienced and inexperienced computer users with no prior experience with the Notes program. Some had no prior experience with Andrew, the computer system on which Notes is implemented.

Methods. Each participant comes to two sessions. The first session is a training session. In the training session, we provide a one-on-one tutorial introduction to those parts of the Andrew system that participants need in order to work with the Notes program. The training time on Andrew averages about 30 minutes. Then we give participants a hard copy tutorial introduction to the Notes program and ask them to work through the tutorial at their own pace. The average time to work through the tutorial is about 45 minutes for experienced computer users, 90 minutes for inexperienced.

In session two, we ask the participants to read two short articles on an issue (controlling human behavior), and to write an essay that (1) synthesizes the issues from the other two essays as a springboard for developing a position on the issue and (2) lays out their position on the issue. To make the task demanding, we impose a time constraint: 45 minutes to read the essays and 45 minutes to write a draft. We ask participants to use the Notes program to take notes and write their essays. In both training and work sessions, we ask participants to think-aloud as they work and we record what they say [Ericsson & Simon, 1984].

All participants have taken the full 45 minutes to read the essays and all participants but one have taken the full 45 minutes to write the draft of the essay.

Our observations of the errors participants make and the thinking-aloud protocol data gave us a wealth of information about specific problems with the program, problems that were, for the most part, relatively easy to fix. But the most valuable information about the overall design of the program comes from interviews with participants after they have completed the reading and writing task. The interview questions, based in part on a set developed by Hidi and Klaiman [Hidi83], focus the users' attention on the process of taking notes and probe for the Notes program's effects on their usual note-taking processes.
7.1.2 The Comments Program

We evaluated the Comments program interface by a task analysis and informal user testing (Keim & Greene, 1987). We have also interviewed our teacher and student users about their use of the program.

Both programs have menu options that allow users to report problems to us, and we also maintain campus-wide electronic bulletin boards where users can discuss reactions, problems, etc. These have also been valuable sources of formative evaluative information for us.

7.2 Formative evaluation during operation

7.2.1 The Notes Program

We have operated Notes in an average of 5 sections of Freshmen English courses for each of five semesters. (We have not expanded its use because demand for Andrew workstations, the system on which Notes is implemented, far exceeds the supply.) Two of the sections are typically Strategies for Writing, the introductory writing course that all but advanced placement students take; one section is usually Reading Texts, a literature course designed to teach students strategies of interpretation and response; two sections are typically Reading & Writing Arguments, a writing course designed to teach students how to write original essays on an argumentative issue. The program has also been used successfully with junior and senior engineers in a writing course at UC-Berkeley. Although not part of our original target audience, several graduate student teaching fellows have used the program for their own research notes. The Notes program has also been distributed on the ICEC tape to schools belonging to the Inter-university Consortium for Educational Computing.

The most ambitious use of the Notes program occurred in 76-122 Reading & Writing Arguments, Fall 1987. The two instructors for the course arranged to have approximately 40 source texts online for students (The sources were also available on reserve in the library). About 10 additional source texts were available only in the library (in part, this was to motivate a need for taking some notes on hard copy sources). All the source texts are on a single issue-literacy. The students were required to use these source texts for their final paper in the course, a paper in which they were to construct an original position on some aspect of the literacy issue. The instructors required the students to take notes using the Notes program.

We observed 10 of these students, 5 from each section (one student dropped out for personal reasons), in two sessions. In the first session, we trained students to criterion on using the program. In the second session, we used a within subjects design to observe them taking notes using online sources and the Notes program versus hard copy sources and notes. We took thinking-aloud protocols as they worked. In addition, we collected all their Notes for each online session and their final papers. We interviewed the instructors and the students and administered a questionnaire to students.

Student attitudes. After being required to use the program to take notes, we measured students' attitudes toward the program by using a set of 7 point scales (For example, 1 = Extremely unhelpful, 2 = Very unhelpful, 3 = Slightly unhelpful, 4 = Neutral, 5 = Slightly helpful, 6 = Very helpful, 7 = Extremely helpful). The scales were based on a set used by (Good, 1981).

As seen in Table 1, students were slightly positive on all dimensions except the item measuring their perceptions of the program's speed, which they rated slightly negatively. We are encouraged by students' positive attitudes toward a program that they were required to use for a challenging assignment. We are still in the process of analyzing other data from the study.
7.2.2 The Comments Program

We have operated Comments in an average of 5 sections of Freshmen English courses for each of three semesters. Although not part of our original target audience, several units within the University have reported using the program. The Comments program has also been announced in the ICEC catalog as available to schools belonging to the Inter-university Consortium for Educational Computing, but it would need considerable work to set up for a network other than the Andrew File System.

Use statistics

When we first released the Comments program, there were 83 potential users, 4 teachers and 79 students. We did not require its use; we simply announced its availability. We tracked its use for the remaining weeks in the semester. The program averaged 83 uses (start-ups)/week (35 S.D). Not all users used the program every week. The average number of users/week was 33 (11.4 S.D.). We speculate that its use fluctuated with writing assignments.

Impact on skill acquisition

MacNealey (in prep.), reports that students working with the Comments program succeeded in fixing more grammatical errors than students working with a traditional hard copy handbook. Although further research is required, this result may be due to the greater diagnostic specificity that the Comments program facilitates.

Student attitudes

After being required to use the program to make comments, we measured ten students' attitudes toward the program by using a set of 7 point scales (For example, 1 = Extremely unhelpful, 2 = Very unhelpful, 3= Slightly unhelpful, 4 = Neutral, 5 = Slightly helpful, 6 = Very helpful, 7 = Extremely helpful). The scales were based on a set used by Good (1981).

As seen in Table 2, students were neutral to slightly positive on all dimensions except the items measuring their perceptions of the program's speed, which they rated slightly negative, and familiarity, also slightly negative.

We also asked students to rate the Andrew text-editor, EZ, a "what-you-see-is-what-you-get" integrated text-editor/document formatter that has undergone a good deal of interface testing. Students rated the EZ text-editor as significantly more familiar than the Comments program (p < .01). Since students are trained on EZ in the Computing Skills Workshop but not on Comments, this result is not too surprising. Students also rated EZ significantly more easy (p < .05) and pleasant (p < .05) than Comments. We were already aware of some ease-of-use problems based on our interviews with users, so we had revised the interface for Comments to simplify the network transactions. The interface described in this report reflects that revision; the student attitudes reflect the old interface. We will be collecting new data to see whether we have improved it.

Patterns of interaction

As noted earlier, we interviewed students and teachers about their use of the program. Some of them reported that the program has altered their patterns of interaction. Some teachers encourage more frequent submission of drafts; although they still maintain a draft due-date, students are encouraged to submit more often with specific questions. Students are enthusiastic about the increased frequency of
interaction as well as not having to wait until class or office hours to ask a quick question. We are currently conducting a study that is looking at possible effects of the network tools, including the Comments program. Based on interview data with teachers and students, we are focusing on possible changes in patterns of interaction and any effects on quality of writing. We are studying two sections of writing that are using network tools; two contrast sections.

7.3 Plans for continuation

The University continues to support the CECE staff required to maintain and enhance the software tools, to train new users and to answer users' questions. In addition, ICEC will provide support to member schools (and other schools for a small service fee) for people using the tools at other campuses.

The course associated with the project, 76-122 Reading & Writing Arguments, continues to be taught. There will be three sections in the Fall, 1988. When the textbook is published (January, 1989), there are plans for making the course part of the core language, culture and history breadth requirement for all English majors.

We have spin-off initiatives underway for continuing the work that we began here. These include major grant proposals to a government agency and to a private foundation.

8 Summary and Conclusions

We have attempted to outline our research, curricular and computer systems insights in the body of the report. Here, we confine ourselves to observations related to some outstanding problems. Although CMU has one of the highest access rates of students to computers in the country, our efforts have been hampered by the fact that demand for workstations continues to exceed supply, even in a computer rich environment such as CMU. We learned that these tools may be useful, but that it is unlikely students will use them outside a course that requires it until access to the technology improves. We are encouraged by the latest developments in advanced function workstation technology—a technology that we continue to believe is essential for supporting extended reading and writing tasks. The prices for high-function workstations continue to drop. In response, CMU will be recommending that students buy such machines.

Besides access, a continuing problem that plagues educational computing remains hardware/software incompatibility. Because our textbook is printed on paper, everyone in the country will have access. Because of hardware and software incompatibilities, the same cannot be said of our computer tools. Our tools run on Andrew, a system that was designed to maximize portability to different hardware. As a result, our tools run on IBM-RT-PCs, SUN 2s & 3s and DEC VAX-stations. They may soon run on MAC IIs—but under UNIX, not the MAC operating system. We remain committed to the idea that educational software should be as portable as books—an idea that seems like a long way off as we face yet another influx of incompatible hardware and software into our college.

Despite these very real problems, we remain committed to the view that computer-based tools hold the potential to achieve revolutionary improvements for students learning to read and write—and for their teachers. We hope that our project has contributed one small step toward that revolution.

Consequential
Length of time
Copy cats
Originality criterion
Newness
Value vs. Consequential
Subjectivity and value?
Being there
Housepainter example
Intentionality


Memory
More creative
Criteria for postulating an ability

Figure 1: The Notes Program Window
Creative acts come in a great variety of forms. A creative act may be quite ordinary and inconsequential—for example, it might be something as simple as making up a bedtime story to tell our children—or it may be world shaking—as was Galileo's invention of the science of physics. A creative act may involve years of concentrated work—consider the decades Darwin devoted to developing the evidence for the theory of evolution—or it may be brief—condensed into a sudden flash of insight—the sort of insight that drove Archimedes naked from his bath shouting, 'Eureka!'

What is there about these very different acts that leads us to call them all 'creative'? Typically, we apply fairly stringent criteria in judging creativity. In most cases, we require an act to pass three tests before we call it creative. First, we must believe that the act is original. Second, we must believe that it is valuable. And third, it must suggest to us that the person who performed the act has special mental abilities. For example, when we see what the person has done, we ask ourselves, 'How did she ever think of that?' or, 'How did he have the patience to work all that out?'

Let's examine these conditions in order:

Originality

We certainly wouldn't judge a painter creative who simply copied the pictures of other painters. To be judged creative, painters must use their own resources to shape the painting. They must paint their own pictures.

We don't mean though that everything in a creative work must be original. Painters, writers, and inventors routinely use ideas borrowed from

Figure 2: Taking a Note
Creative acts come in a great variety of forms. A creative act may be quite ordinary and inconsequential—for example, it might be something as simple as making up a bedtime story to tell our children—or it may be world-shaking—as was Galileo’s invention of the science of physics. A creative act may involve years of concentrated work—consider the decades Darwin devoted to developing the evidence for the theory of evolution—or it may be brief—condensed into a sudden flash of insight—the sort of insight that drove Archimedes naked from his bath shouting, ‘Eureka!’

What is there about these very different acts that leads us to call them all ‘creative’? Typically, we apply fairly stringent criteria in judging creativity. In most cases, we require an act to pass three tests before we call it creative. First, we must believe that the act is original. Second, we must believe that it is valuable. And third, it must suggest to us that the person who performed the act has special mental abilities. For example, when we see what the person has done, we ask ourselves, ‘How did she ever think of that?’ or, ‘How did he have the patience to work all that out?’

Why criteria?

Why is it important to have criteria?


Consequential
Length of time
Copy cats
Originality
Criterion
Newness
Value vs. Consequential
Subjectivity and value?
Being there
Housepainter example
Intentionality
Why criteria?


Figure 3: Composing a Note

Consequential Length of time
Originality criterion
Newness
Value vs. Consequential
Subjectivity and value?
Being there

Copy cats

Copy cats not creative. But someone can be 'inspired' or 'influenced' by another's work and still be creative. Where do we draw the line?

Maybe borderline case aren't important.

OK. But presumably Archimedes was working on the problem for some time before the 'Eureka!'

Chance favors the well-prepared mind.

Isn't originality just another word for creativity? What is the concept going to explain about creativity? publicity subject

Figure 4: Viewing and Classifying Notes
<table>
<thead>
<tr>
<th>Abilities</th>
<th>Alternative Lists</th>
<th>criteria for postulating an ability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Housepainter example</td>
<td>Subjectivity and value?</td>
<td>Both Hayes and Perkins discuss criteria for creativity. Whereas Hayes' seems to be driving at ways to distinguish creative acts from ones that are not creative, Perkins' criteria seem designed to distinguish a specifically creative ability from other, perhaps more general mental abilities.</td>
</tr>
<tr>
<td>Being there</td>
<td>Value vs. Consequential</td>
<td></td>
</tr>
<tr>
<td>Analogy</td>
<td>Length of time</td>
<td></td>
</tr>
<tr>
<td>Intentionality</td>
<td>Copy cats</td>
<td></td>
</tr>
<tr>
<td>Hayes idea</td>
<td>Originality criterion</td>
<td></td>
</tr>
<tr>
<td>Abilities mix vs. stuff</td>
<td>Newness</td>
<td></td>
</tr>
<tr>
<td>Really a mix?</td>
<td>Value vs. Consequential</td>
<td></td>
</tr>
<tr>
<td>Talent or knack</td>
<td>Subjectivity and value?</td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>Being there</td>
<td></td>
</tr>
<tr>
<td>More creative</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creative is better?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Criteria for postulating an ability</td>
<td>Perkins</td>
<td></td>
</tr>
<tr>
<td>Perkins</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


Consequential Length of time Copy cats Originality criterion Newness Value vs. Consequential Subjectivity and value? Being there Housepainter example Intentionality Why criteria?


Figure 5: Alternative Lists
Open the menus and choose one of the following from the Options menu card:

Send a Paper -- to send a paper.
Request a Paper's Return -- to request the return of a paper you have already sent.
Read a Paper -- to read or make comments on a paper.
Delete Comments -- to remove all comments from a paper.
Help -- to call up the program's online help.
Report a Program Problem -- to report a problem with the program.
Quit -- to quit the Comments program.
Calculus is a technical discipline, one that has only a few fundamental concepts. Yet, to understand these concepts, which are, in themselves, not intensely difficult, a wealth of knowledge must be known beforehand. These concepts have numerous applications, this is where much of the technical aspects come in, that is separating the different applications. It is in understanding all the various applications that a text comes into use. Usually, a text defines the concept in theoretical terms, then in general, more basic, terms, with the goal of clearing up any confusion that the student had in reading the previous section. Finally, most texts give practical examples of the problem that they are discussing. The problem that this paper will discuss is how well a subject, in comparison to other subjects, and to what degree the subject understands the text.

The text that I am using is one designed for a college with a strong mathematical department but not as strong as Carnegie Mellon's. It is also not as weak as the textbook for C.M.U's humanities and Social Sciences. It is probably for a school that has a more integrated humanities and science program. It could also be useful for high school A.P. courses. It's name is

Figure 7: Making a Comment
Calculus is a technical discipline, one that has only a few fundamental concepts. Yet, to understand these concepts, which are, in themselves, not intensely difficult, a wealth of knowledge must be known beforehand. These concepts have numerous applications, this is where much of the technical aspects come in, that is separating the different applications. It is in understanding all the various applications that a text comes into use. Usually, a text defines the concept in theoretical terms, then in general, more basic, terms, with the goal of clearing up any confusion that the student had in reading the previous section. Finally, most texts give practical examples of the problem that they are discussing. The problem that this paper will discuss is how well a subject, in comparison to other subjects, understands the text and to what degree the subject understands the text.

The text that I am using is one designed for a college with strong mathematical department but not as strong as Carnegie Mellon's. It is also not as weak as the textbook for C.M.U's humanities and Social Sciences. It is probably for a school that has a more integrated humanities and science program. It could also be useful for high school A.P. courses. It's name is

<table>
<thead>
<tr>
<th>commenter: dm46</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Find someone—a roommate, another person in the class, or a person in your dorm—and explain to them what your paper is going to be about in a few sentences.</td>
</tr>
<tr>
<td>Write down what you said to them.</td>
</tr>
<tr>
<td>3. Compare it to what you have written in this paragraph. (I predict that the content of the two paragraphs will be about the same, but that you will be happier with the paragraph you generated by explaining the paper to your friend.)</td>
</tr>
<tr>
<td>4. Get back to me with your results and discussion on this small experiment.</td>
</tr>
</tbody>
</table>

Figure 8: Composing a Comment
Calculus is a technical discipline, one that has only a few fundamental concepts. Yet, to understand these concepts, which are, in themselves, not intensely difficult, a wealth of knowledge must be known beforehand. These concepts have numerous applications, this is where much of the technical aspects come in, that is separating the different applications. It is in understanding all the various applications that a text comes into use. Usually, a text defines the concept in theoretical terms, then in general, more basic, terms, with the goal of clearing up any confusion that the student had in reading the previous section. Finally, most texts give practical examples of the problem that they are discussing. The problem that this paper will discuss is how well a subject, in comparison to other subjects, understands the text and to what degree the subject understands the text.

The text that I am using is one designed for a college with strong mathematical department but not as strong as Carnegie Mellon's. It is also not as weak as the textbook for C.M.U's humanities and Social Sciences. It is probably for a school that has a more integrated humanities and science program. It could also be useful for high school A.P. courses. It's name is comment: dm46

predict that the content of the two paragraphs will be about the same, but that you will be happier with the paragraph you generated by explaining the paper to your friend.

Get back to me with your results and discussion on this small experiment.

OK, I tried what you suggested and this is what I came up with:

While calculus uses only a few fundamental concepts, it is a highly technical discipline. The concepts, in and of themselves, are not too difficult, but a wealth of knowledge is necessary to be able to understand and use these concepts correctly. The decisions regarding how to apply the concepts are what make calculus such a technical discipline. A calculus text must be able

Figure 9: A Dialogue in the Comments Program
<table>
<thead>
<tr>
<th>Dimension</th>
<th>Mean</th>
<th>S.D.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helpful</td>
<td>5.5</td>
<td>0.8</td>
</tr>
<tr>
<td>Nice</td>
<td>5.4</td>
<td>1.0</td>
</tr>
<tr>
<td>Friendly</td>
<td>5.3</td>
<td>0.9</td>
</tr>
<tr>
<td>Fast</td>
<td>3.1</td>
<td>1.6</td>
</tr>
<tr>
<td>Familiar</td>
<td>4.7</td>
<td>1.3</td>
</tr>
<tr>
<td>Powerful</td>
<td>5.0</td>
<td>0.8</td>
</tr>
<tr>
<td>Useful</td>
<td>5.5</td>
<td>0.9</td>
</tr>
<tr>
<td>Easy</td>
<td>5.4</td>
<td>1.2</td>
</tr>
<tr>
<td>Pleasant</td>
<td>4.8</td>
<td>1.4</td>
</tr>
</tbody>
</table>

N = 9

Table 1: Student Attitudes toward the Notes Program
Helpful
Nice
Friendly
Fast
Familiar
Powerful
Useful
Easy
Pleasant

(p < .01)

(p < .05)

\[
\begin{array}{c}
\text{Strongly Disagree} \\
\text{Strongly Agree}
\end{array}
\]

\[\square - \square = \text{Comments Program}\]

\[\bigcirc - \bigcirc = \text{EZ Text-editor Program}\]

Table 2: The Comments Program vs. The EZ Text-editor Program
References


Notes

1A planned extension will allow users to see note cards by other program-generated orders, such as grouped by classes that the user has put the notes in.

2Because the system that Notes is implemented on allows multiple windows, the entire screen may not be devoted to the Notes program. If a user has more than one window on the screen, Notes occupies the portion of the screen that the user has allocated for it, but the Notes window itself would still appear as described. In addition, the Notes window itself can take on different appearances. For example, the user can hide various regions of the Notes program's window from view and expose regions to view. For example, if the user is primarily engaged in taking notes, he or she may not want the listing and viewing regions exposed to view and there is an option to Hide/Expose the All Notes List and Viewing regions.