WARRANT - A Flexible Computer Environment for Critical Reading, Reasoning, and Writing

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

Cheryl Geisler  
*Rensselaer Polytechnic Institute*

Christine Neuworth  
*Carnegie Mellon University*

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

Follow this and additional works at: [http://repository.cmu.edu/philosophy](http://repository.cmu.edu/philosophy)  
Part of the [Philosophy Commons](http://repository.cmu.edu/philosophy)
WARRANT
A Flexible Computer Environment
for Critical Reading, Reasoning, and Writing

In the following areas, contact the following people:

- **Curriculum Development**
  - David S. Kaufer (412-578-2846)

- **Data Collection and Analysis**
  - Cheryl Geisler (412-578-3721)

- **System Design and Implementation**
  - Christine Neuwirth (412-578-8702)

- **Budget**
  - Preston Covey (412-578-8493)

First-Year Report to the Fund for the Improvement of Postsecondary Education. March 1, 1985.
# Table of Contents

1. Executive Summary  
   1.1. Original Plans  
      1.1.0.1. The Curriculum  
      1.1.0.2. The System  
   1.2. Current Plans and Accomplishments  
      1.2.1. Curricular Development  
      1.2.2. Data Collection and Analysis  
      1.2.3. System Design and Implementation  
2. Curriculum Development  
   2.1. Curricular Goals  
   2.2. The Topic of Paternalism  
   2.3. The Corpus  
   2.4. The Task  
      2.4.1. Task for Experts  
      2.4.2. Task for Novices  
      2.4.3. Task for Students  
   2.5. Possible Evolutions of the Curriculum  
3. Data Collection and Analysis  
   3.1. The Role of Empirical Work in the Development of Warrant  
   3.2. The Design of the Study  
   3.3. Methodology  
      3.3.1. Protocol Analysis  
      3.3.2. Elite Interviews  
      3.3.3. Participant Observation  
   3.4. First-Year Outcomes  
      3.4.1. Organization of the Warrant Curriculum  
      3.4.2. Advice and Models in Warrant  
      3.4.3. Plans in Warrant  
   3.5. Accomplishments after Two Months  
      3.5.1. Organization of the Team  
      3.5.2. Overall Accomplishments  
      3.5.3. A Study of Experts  
         3.5.3.1. Narrative on Expert #1  
         3.5.3.2. Narrative on Expert #2  
         3.5.3.3. Narrative on Expert #3  
      3.5.4. A Study of Novices  
         3.5.4.1. Narrative on Novice #1  
         3.5.4.2. Narrative on Novice #2  
      3.5.5. A Study of A Class  
         3.5.5.1. Narrative of the Teacher and his Class  
         3.5.5.2. Narrative of Selected Students  
   3.6. Plans for Year 2  
      3.6.1. Fall, 1985  
      3.6.2. Spring, 1986
3.6.3. Summer, 1986

4. System Design and Implementation

4.1. Proposed Design
   4.1.1. Design Goals
   4.1.2. User Interface Design

4.2. Development Context: The Andrew System
   4.2.1. Sun Workstation
   4.2.2. Berkeley Unix 4.2
   4.2.3. Vice Shared File System
   4.2.4. Virtue Software
      4.2.4.1. The Window Manager
      4.2.4.2. The User Interface Libraries
      4.2.4.3. Relation of Virtue to Warrant

4.3. Activities and Outcomes After Two Months
   4.3.1. Completing Personnel Requirements
   4.3.2. Learning the Andrew system
   4.3.3. Functional Specification
   4.3.4. Initial Implementation Architecture

4.4. First Year Outcomes

4.5. Plans for Year 2 and 3

5. Dissemination

5.1. Books
   5.1.1. Critical Reasoning/Writing Pedagogies
   5.1.2. Approaches and Methods for Curriculum Planners
   5.1.3. The WARRANT System

5.2. Conferences and Articles

6. Bibliography

7. Budget

Appendix I. Giving Advice for Critical Skills

Appendix II. Functional Specification for the Note Card Component

Appendix III. Corpus on Paternalism The WARRANT Project Spring 1985
1. Executive Summary

The WARRANT project represents an attempt to teach reading, reasoning, and writing skills to students on powerful, large-screen personal computers, supporting high resolution bit-map display graphics and multiple-windowing. This report overviews and then details our current and expected progress on the project from January 1, 1985 through September 30, 1985.

This chapter is intended to review our original plans, to indicate where and how we have extended our plans, and to highlight our current progress. The remaining chapters report on three components of the project -- curriculum development, data collection and analysis, and system design and implementation -- in greater detail. In addition, chapters are devoted to dissemination efforts and the budget.

1.1. Original Plans

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

1.1.0.1. The Curriculum

Little was firmly known about the strategies that are most effective for teaching reading, reasoning, and writing. We originally proposed, using verbal protocols, to track expert and novice writers as they work through a corpus of reading/writing assignments. This data would be used to develop a curriculum for teaching writing/reading/reasoning as interactive processes. We also proposed a textbook on critical reading, writing, and reasoning, to be written by Covey and Kaufer, detailing for teachers in a highly readable form what we had learned from the protocol data.

1.1.0.2. The System

Our original proposal also called for the development of software to deliver this curriculum on powerful workstations currently being developed as a joint project of IBM and CMU. We proposed to use these workstations to teach pilot courses in freshman philosophy and freshman composition by early 1987. Having a powerful workstation to “run” our curriculum meant that we would be able, through case studies and, perhaps, controlled studies, to determine how well our curriculum worked and what were the most effective ways of
presenting it to students. We also proposed a book, to be written by Neuwirth, Kaufer, and Geisler, to report on our development of the system, the coding of the curriculum into it, and the results of our case studies.

1.2. Current Plans and Accomplishments

While we have only been working on WARRANT full time since January 1, we are proceeding with plans that are at once far more concrete than our original proposals and yet, we think, even more ambitious. In this section, we report on top-level specifications and modifications in our plans both at the curricular, data, and systems level. Fuller reports are given in subsequent chapters.

1.2.1. Curricular Development

We have 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 amongst these authorities. Our reading/writing curriculum seeks to support this development by focusing on the following skills: critically reading and representing an author's view; summarizing that view; establishing criteria for evaluating that view and then using those criteria to evaluate it; evaluating views relative to others; situating oneself. We have also chosen to use the topic of paternalism. Chapter 2 contains further details.

1.2.2. Data Collection and Analysis

We have decided on a three-part approach to gathering data for our 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 expert writers. In addition, we have added a third foci: (3) gathering data from a teacher and students in a course where the paternalism corpus is being taught. We have also expanded our methodology to include, in addition to think-aloud protocols, elite interviews and participant observation. Chapter 3 contains further details.
1.2.3. System Design and Implementation

Following the original proposal, the computer system we have been developing to deliver the WARRANT curriculum aims for the flexible integration of easy-to-learn tools. The WARRANT system will provide students with "windows" in which they can consult the syllabus, read text, follow a plan, get advice, take notes, compose a text, and receive and make comments. The prototype system consists of a Sun workstation and the Andrew file server and system software. In addition to learning this system, we have completed the preliminary specifications for the Note Card Component. Further details can be found in Chapter 4.
2. Curriculum Development

In this chapter, we detail more thoroughly our curricular goals for the WARRANT environment. Specifically, we describe the intellectual skills that underlie our curricular efforts; and the topic on which the curriculum focuses; and the task for which the skills are intended.

2.1. Curricular Goals

As we described in the overview, our thinking about optimal curricular approaches has been greatly influenced by the work of William Perry (1970). Studying Harvard students over the course of four years, Perry observed a systematic development from absolutist or dualist, to relativist, to what he calls "committed" positions. The dualist sees the world as narrow either/or. The relativist tolerates a multiplicity of views and does not try to discriminate among them. The committed person also acknowledges diversity but seeks systematically to discriminate what she is willing/unwilling to commit to. While Perry seems to think of these positions as "stages" that evolve over the course of a liberal arts curriculum, we prefer to think of them as increasingly comprehensive "skills" that students evince as they begin to master critical reading, writing, and thinking processes.

Perry's absolutists, for example, know neither how to acknowledge problems in authorities nor question the authority itself. His relativists, in turn, can acknowledge problems in authorities and can call authority into question, but they don't seem to know how to weigh authorities against one another, how to arrive at independent conceptions of an issue, or how to situate their own views relative to other authorities. We associate all of these latter skills with Perry's notion of commitment. Our curriculum aims to teach all the skills embodied in this notion under the rubric of critical reading, writing, and thinking processes.

More specifically, we have broken down our curriculum goals into the following seven interactive skills:

1. using language reflectively (weighing expressions against concepts)
2. critical reading (understanding and analyzing text structures)
3. critical summarizing (generalizing from a critical reading)
4. critical evaluating (finding problems from the reading)
5. assumption-digging (understanding authors as committed persons)

6. situating (finding one's own commitments among committed authors and defending those commitments effectively)

2.2. The Topic of Paternalism

We chose to focus our curriculum around the topic of paternalism. Roughly speaking, paternalism concerns situations in which one adult interferes with the autonomy of another adult "for his own good." We find it to be an ideal issue for three reasons:

- Paternalism is a difficult, open-ended issue with no clear points of closure: it is an open issue for experts no less than novices; on the other hand, it is not so nebulous to prevent reasonable adjudications between competing lines of argument.

- Paternalism is an issue which has touched the personal experience of every student; students are easily able to generate examples of paternalism in their own lives; no students is put at a special advantage or disadvantage by the issue.

- Paternalism has no political axes to grind. Where a person stands on Democrat/Republican, liberal/conservative, race, or religious lines has no predictable bearing on where he or she will stand on paternalism. Paternalism, in other words, is not a special interest issue. It is a classic issue about the power of individuals to define what is good for them apart from external influences and, as such, defies stereotypical categories or labels. Thus, the potential for exporting our curriculum to a wide cross-section of public and private, religious and nonsectarian schools remains high.

2.3. The Corpus

We have selected eight articles along with introductory materials for a sixty-eight page corpus. We have written away for permission to use/reprint these materials in forthcoming reports and publications and, thus far, we have not run into any problems. (The corpus is enclosed in Appendix 3). We should note that the corpus is still evolving and we briefly discuss this in the last section.

The corpus is currently divided into four parts. We introduce each part and its rationale vis-a-vis the overall curriculum.

Introductory Materials

This first section is designed to give students some brief exposure to the multi-faceted nature of paternalism. We are currently in the process of
greatly expanding these materials for reasons we will discuss below.

Examples and Nonexamples of Paternalistic Behavior
This section is designed to give students explicit concept training on paternalism. Given examples and (close) nonexamples of the concept, students are asked to write "initial" definitions.

Selections that Seek to Define the Concept of Paternalism
This section is designed to give students exposure to different approaches to defining the concept. Dworkin defines paternalism at the public sphere, as occurring among anonymous persons. Gert and Culver and Childress limit their definition to the interpersonal realm. Among other things, students are encouraged to write on the similarities and differences between these frameworks.

Controversial Examples of Paternalistic Behavior
This section offers six complex cases of interpersonal paternalism (between doctors and medical patients) and asks students to judge whether the paternalistic behavior seems justified or not.

Selections that Seek to Define Conditions for Justifying Paternalism
The authors of these selections offer different principles for justifying paternalism. After reading and summarizing these authors, students are encouraged to apply each of these principles to each of the six controversial cases. They are asked to discuss the strengths and weaknesses of each of these principles, to formulate (or at least commit themselves to) a principle and to situate their preference among the preferences implied by the principles that seem to them less worthy.

2.4. The Task
Our data-gathering efforts are premised on the assumption that expert philosophers should have assimilated many if not all of these skills as they work through a reading/writing/reasoning assignment; that novice writers (college freshmen) should illustrate many gaps working through a comparable assignment; that experienced teachers of critical reasoning and writing should show a sensitivity to these gaps and should have built up a more or less explicit repertoire of strategies to help students overcome them. To learn as much as possible about the comparable skills of our novices, experts and students, we have given our different groups comparable tasks. In this case, however, comparable does not mean identical. We have found it necessary, even desirable, to make some adjustments in the assignments we give to our different populations.
2.4.1. Task for Experts

You are writing an article discussing the current state of thinking on medical paternalism.

Your intended audience are readers of a popular magazine such as *The Humanist* or *Harpers*. They are well-educated people who may at some time in their lives have to deal with the issue of paternalism, either personally or professionally; but they are not technically trained in philosophy.

Your article will be a paper in two parts. In the first part you should summarize and evaluate the definitions of medical paternalism given in the first part of the corpus.

In the second part, you should summarize and evaluate the attempts to state the conditions under which paternalism is justified. They can be found in the second part of the corpus.

Please keep all of the things you write in each session and allow us to make copies of them.

2.4.2. Task for Novices

You are writing a long paper (term paper length) discussing the current state of thinking on medical paternalism.

Your intended audience are readers of a magazine such as *Atlantic* or *Harpers*. They are well-educated people who may at some time in their lives have to deal with the issue of paternalism, either personally or professionally. They will want to understand what medical paternalism is, why it is morally questionable, and under what conditions it can be justified.

Your paper should be written in two parts. In the first part you should summarize and evaluate the definitions of medical paternalism (given in the first part of the corpus), and formulate your own definition.

In the second part, you should summarize and evaluate the attempts to state the conditions under which paternalism is justified (given in the second part of the corpus) and state your own conditions for justification.
We expect that you will spend your time in three types of activities: reading, writing, and thinking; but it is up to you to decide how to go about it.

Please keep all of the things you write in each session and allow us to make copies of them.

2.4.3. Task for Students

Unlike our experts and novices, who are expected to move at their own rate without instruction, the students in our course are moving very deliberately through the corpus. The two assignments—defining and justifying contexts of paternalism—given to our experts and novices are broken down for our students into thirteen or fourteen writing assignments. The teacher of the course, like the teacher of any course, has the luxury to break down complex skills into simpler chunks that students can more easily absorb and practice on. The writing assignments in the course follow the corpus very strictly.

As we mentioned previously, students are first assigned to read examples and nonexamples of paternalism so that they can begin to grasp the features that are included and excluded from the concept. They then read the three selections on defining paternalism. Authors of these selections (Dworkin, Gert and Culver, Childress) define the concept very differently and these differences, along with their implications, are discussed. Also discussed are the constituents of a "good" definition of a difficult, controversial issue. What should a good definition do? What features must a definition exhibit and in what proportion for a definition to do what it should do? What is the difference between defining an issue and deciding it? These are questions that students have seldom, if ever, thought about or explored. Raising these questions, further, provides a supportive context for teaching the seven interactive skills of reading/writing/reasoning we have set as our curricular goals.

Having worked on defining the issue, students now turn their attention to exploring and resolving it. They are first given a list of six extended cases of paternalism, cases whose moral justification is a matter of dispute. They then "read, summarize, and evaluate" five authors (Dworkin, Gert and Culver, Childress, Carter, Komrad), all of whom offer different principles for deciding on the justifiability of paternalistic behavior. With each author's principles in hand, students are encouraged and taught to simulate each author's reasoning on each of the six cases. Students should find that some authors' reasoning squares better
with their intuitions than others, but that no author can perfectly match their own intuitions for each case or, for that matter, cases they might make up. For this reason, students are encouraged—and taught—how to explore their own premises about elusive concepts (e.g., consent, autonomy, competence) that support any attempted resolution of the justifiability of paternalistic behavior. Using their own premises, students are asked, at the end of the course, to formulate their own principle for justifying paternalism and to situate it with (i.e., discuss its strengths and weaknesses) the other principles they have read vis-a-vis the six controversial cases.

2.5. Possible Evolutions of the Curriculum

For Warrant’s curriculum, we have set a number of dissemination goals, some of which are competing: We want a curriculum that interests and challenges both students and teachers; that is fun to learn and fun to teach; that is not so difficult as to be intimidating; that is easily transportable to many institutional and disciplinary contexts; that is coherent in its educational objectives; that is diverse in its educational objectives.

As the curriculum evolves, we shall try to meet all these goals while at the same time making the curriculum increasingly self-contained. In the interests of making the curriculum more self-contained, we are already beginning to write longer, more explanatory, introductory materials on paternalism and more complete materials overviewing the entire curriculum. In the interest of making the curriculum more diverse and interdisciplinary, we are considering creating a substantial corpus of readings on paternalism from diverse areas: economics, journalism, history, religion, law, education, politics, literature, medicine, the blue-color professions, drama, ethics, and so on. In this way, students who are given the assignment to “define” paternalism will have a very rich source of data to refine and elaborate their definition. No less importantly, such a corpus will allow teachers to use Warrant to teach a general course in cultural studies in addition to or instead of a course in critical reading, reasoning, and writing.
3. Data Collection and Analysis

3.1. The Role of Empirical Work in the Development of WARRANT

In some areas, the design of computer-aided instruction is more or less an engineering problem. Educators know what they want to teach; they know how to teach it; and they can even anticipate the range of difficulties their students will run into. When development problems occur, they are usually technical—the screen is too small, the memory too limited, the parser inflexible. But the educational knowledge underlying the technology is already there.

In WARRANT's area of critical reading, reasoning, and writing skills, however, much remains to be discovered. We are planning to develop this knowledge through empirical observation. The results of our study will serve as the basis for the WARRANT curriculum by giving us the plans, advice and models called for in the original proposal. In addition, these results should assist other educators who wish to use the WARRANT architecture for their own curricula without the expense of a large-scale study. In this way, the empirical component of the WARRANT project serves both our dissemination and development goals. In the rest of this chapter, we will review the design of this empirical study, discuss our methodological choices, and, in the course of reviewing expected first-year outcomes, report some of what we have learned after only two months of work. We will end with our plans for Year 2.

3.2. The Design of the Study

WARRANT is a technology designed to deliver a curriculum on critical skills. Abstractly, any curriculum aimed at teaching skills attempts to take students at their own "novice" level and move them closer to "expert" performance. William Perry (1970) has described how students proceeding through a liberal arts curriculum make the transition from dualists to committed relativists. The empirical study described in this section is designed to illuminate this transition from several foci.

First, we have begun to observe experts at critical thinking. Despite the widespread agreement that our students lack critical skills, and the growing attempts to devise curricula to remedy that lack, we have not found an adequate description of the skills being referred to. What does it mean to be good at critical thinking? Our own expertise in critical skills is not sufficient to answer these questions; our very "expertness" means we take for
granted, and thus are unconscious of, the many strategies we use. To make us more conscious of these strategies, we have chosen to study a group of expert ethicists working on our task including, among others, Preston Covey.

Second, we have begun to observe how novice freshmen go about the same task. Students do not come to our curricula as tabulae rasae. They bring skills, strategies, and assumptions that either work for or against them. Usually teachers only become aware of these starting points when they begin to teach a curriculum—and then only dimly. For WARRANT, we need to have this information in advance. Thus, we have chosen to study several freshmen working on our task.

Lastly, we have begun to observe how one teacher and his students work together on critical skills. The opportunity to make these observations arose somewhat serendipitously—one of us, David Kaufer, was asked to teach a course in critical thinking this fall. What we once saw as a fortuitous occurrence, we now consider to be absolutely critical to our entire design. What Kaufer is doing through intuition we hope WARRANT will do more systematically, albeit, less intelligently: translate his expertise into advice that novices can use to become critical readers, writers, and thinkers. WARRANT must not simply contrast expert and novice behavior through models; it must also suggest a plan and give students advice to help bridge the gap between the one and the other. In light of these objectives, we have chosen to study how one teacher and his students intuitively cross this bridge in the course of a semester.

The relationship between these three foci (experts, novices, and a teacher & his students) and the WARRANT curriculum is represented schematically in Figure 1. The curriculum itself will be the result of an analysis, a triangulation if you will, of these three sources of information. In practice, this means, for instance, that a piece of advice in WARRANT will not simply be an abstraction of what one expert did, or of what one novice did, or even of what one teacher told his class to do, but rather will be a synthesis of the three. Furthermore, we expect that understanding how we go about making this synthesis will help us to make recommendations to other educators who want to make an intuitive synthesis in their own curricula.

In the next section, we review briefly how our choice of methodologies will make our observations as useful as possible.
3.3. Methodology

Our basic strategy for gathering information is a combination of field and laboratory research methods. In our original proposal, we planned to track our experts and novices using the laboratory method of protocol analysis (Newell and Simon, 1972; Ericsson and Simon, 1984). Since then, we have broadened our data-gathering efforts to include two field research methods: elite interviews and participant observation (Lofland, 1966; Dexter, 1970; Shatzman and Strauss, 1973; Patton, 1980; Goetz and LeCompte, 1984). We believe this combination of approaches will give us greater insight into the information we need to build the WARRANT curriculum. We detail our reasoning below.

3.3.1. Protocol Analysis

To observe how our experts and novices go about reading, thinking and writing in a normative issue, we have chosen to use the laboratory method of protocol analysis. In this technique, an expert or novice thinks aloud as s/he works to complete the task. These thoughts are tape-recorded and later transcribed and interpreted with the aid of both the material being read and the material being written (all of which is carefully preserved).

Protocols have several characteristics which make them highly suitable for our purposes in WARRANT. First, they give a rich and ordered record from which the strategies a thinker uses to solve a problem can be extracted. This record will miss some things: things that can't be put into words or things that flit through the mind too quickly. Yet it preserves
what would otherwise be lost: the order in which a thinker takes up his or her tasks, the strategy s/he uses, the many small problems and various solutions he or she encounters. Experience shows that people forget or distort this information quickly.

Second, protocols provide a setting in which the researcher can define a clear initial goal for a task. We ask our experts and novices to complete the task described in the previous chapter. Even though each one will go about this task differently, we will be able to see contrasts and similarities. Without this initial task definition, however, we could never hope to make such systematic comparisons.

Protocols can have several drawbacks which we have tried to minimize on this project. First, they can place an undue burden on the thinker and thus distort or destroy the processes we are trying to observe. Studies have shown that this occurs less than would be imagined and is usually limited to those tasks which are inherently non-verbal--painting, for example. Nevertheless, distortion can occur if the informant feels under an obligation to explain to an audience what he or she is doing rather than simply doing it. As a team, we have worked out procedures to emphasize the "doing" rather than the "explaining."

Secondly, a protocol can require informants to complete a task under conditions that makes little sense to them. We avoid this by cooperating with our informants to accommodate their normal concepts and procedures. Initially, we make sure that they can interpret our task definition in a way that makes sense to them. Subsequently, we allow them to work at their own rates in a comfortable environment. (Experts generally work in their offices or at home; novices, who live in crowded dorm conditions, work alone in a private office.) Throughout, we stress our willingness to listen to and adjust for any difficulties they may encounter.

3.3.2. Elite Interviews

In WARRANT, we want to supply students with a conceptual framework strong enough to give them control over their own goals and strategies. While protocols provide rich information about an informant's strategies and some information about goals, they often leave unexplored the informant's conceptual framework and prior assumptions. In order to capture this information, we have chosen to use a field method called elite or nonstandardized interviews in conjunction with the thinking-aloud protocols.
In an elite interview, according to Dexter (1970), “the investigator is willing, and often eager to let the interviewee teach him what the problem, the question, the situation, is...” (p. 5) Each of our informants is assigned to one member of the data collection team. This interviewer is responsible for developing an understanding of how each informant conceives of his or her task at any given point. S/he meets with the informant for about a half-hour interview between each working session to explore the informant's conceptions of what s/he has just done, what s/he is now thinking, and where s/he thinks she is going.

We use elite interviews with four groups of people. We interview our experts and novices regularly between protocol sessions. We interview our teacher (Kaufer) before and after each of his class sessions. And we interview each of his students once during the semester.

3.3.3. Participant Observation

Elite interviews are our best source of information about how people conceive of their activities retrospectively or prospectively. But, as we noted in the section on protocol analysis, people are notoriously bad at reporting or predicting their actual activities. When these activities involve an informant working alone, protocol analysis can fill in our knowledge gaps about actual strategies and problem-solving. When informants work together as a group, however, the field method of participant observation gives us our best information about the actual life of the group. This is the technique we have chosen to use in our observations of the conduct of Kaufer's class.

Participant observation allows the observer a range of possible involvements. We have chosen to remain relatively unobtrusive and nonreactive during the class sessions. One of us attends classes, recording the verbal and non-verbal interaction between the teacher and his students. Once a week, a tape recording is also made and used as a check on the accuracy of the observer's notes.

3.4. First-Year Outcomes

In this section, we elaborate the connections between the empirical foci and methodology outlined above and our intended first-year outcomes.
3.4.1. Organization of the WARRANT Curriculum

For WARRANT, we began with the choice of a topic (paternalism) and objectives (reading, summarizing, evaluating, assumption-diging, situating) that were described in the previous chapter. Using his past experience, Kaufer designed the course he is teaching this fall so as to meet those objectives. The first major outcome from our empirical study will be a reevaluation and restructuring of this Stage I curriculum in light of the experience of the experts, the novices, and the teacher & his students. This restructuring will be the beginning of our Stage II curriculum and should be substantially completed in the summer of 1985.

This reevaluation and restructuring will include the following:

- the redefinition of the curriculum objectives (what objectives did Kaufer & his students end up satisfying? How did the experts and novices conceive of their objectives?)
- the partitioning of the curriculum into units (How did each group of informants break down the task? Where did they encounter their difficulties?)
- the scope and nature of the readings in the original corpus (Which readings turned out to be central? Which peripheral? Were there any subissues discussed but not covered by the original readings?)

3.4.2. Advice and Models in WARRANT

As called for in our original design, WARRANT will provide students with advice and models of both experts and novices working through the task. The models will be edited, rewritten sections of the verbal protocols. The advice will be abstracted from two sources: the expert and novice interviews and protocols, and the interviews and observations of the teacher & his students. The work of segmenting and coding protocols, interviews and observations according to strategies and goals will have been undertaken in order to redefine the curriculum objectives; it will be continued here in finer detail. This work constitutes the second phase in the development of our Stage II curriculum and will begin in the summer of 1985.
3.4.3. Plans in WARRANT

The detailed structure of WARRANT's curriculum will be delivered in the form of plans. The Plan box will help a student find out what s/he has done and what is left to do. These plans will be developed as we compare the order of tasks and strategies used by the experts and novices with those used by the teacher & his class. These plans will be the last piece to be put into place of our Stage II curriculum and should be completed in the spring of 1986.

3.5. Accomplishments after Two Months

The work of the empirical team began on January 1, 1985 and has run some seven weeks as of the writing of this report. In this section, we narrate some of our accomplishments in this time period.

3.5.1. Organization of the Team

The data collection team is one of two overlapping sub-groups within the WARRANT team. It is headed by Cheryl Geisler and includes the grant principles (Covey, Kaufer, Neuwirth, Geisler), two Ph.D. students (Ann Penrose and Alexander Friedlander), and one masters student (Wendie Wulff). Our support staff include two part-time typists who transcribe the protocol and interview tapes. The data collection team meet for weekly "review" meetings in which the progress of each of the informants is described and any issues or problems are discussed. In addition, we talk on a nearly daily basis to insure that we follow a common procedure in interacting with our informants. This team will stay intact at least through the development of the Stage II curriculum at the end of the fall, with the exception of Wendie Wulff who will be leaving us at the end of this spring semester.

3.5.2. Overall Accomplishments

Thus far, our accomplishments have been tied to the organization and collection of information from our three sources--experts, novices, and the teacher & his students. Table 1 below details the number of protocols, interviews and observations we have conducted so far. In the first seven weeks, we recorded a total of 93 of our estimated target of 339 hours of observations, or 27% of our data. Of that, 44 hours have already been transcribed. This represents what we have accomplished by February 13, 30% of the way into our data collection period of January through May.
Table 1: Overall Accomplishments

In the remaining sections, we break down these figures according to our three foci and give a narrative summary of the progress of selected informants.

3.5.3. A Study of Experts

In the development of our Stage II curriculum, we are working with five experts, spaced out over a five month period. This spacing maximizes our ability to apply the lessons learned from one expert to the next. Thus far we have begun work with three of the five at approximately three weeks intervals. We expect each expert to take about 30 hours for a total of 150 hours. Overall, we have completed 30% of this work.

<table>
<thead>
<tr>
<th>Informant</th>
<th>Date Started</th>
<th># of Protocols</th>
<th># of Interviews</th>
<th># of Observations</th>
<th>Total Hours as of Feb 13</th>
<th>Expected Hours by May 30</th>
<th>% Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>Jan 1</td>
<td>17</td>
<td>17</td>
<td>--</td>
<td>35</td>
<td>30</td>
<td>117%</td>
</tr>
<tr>
<td>#2</td>
<td>Jan 20</td>
<td>6</td>
<td>5</td>
<td>--</td>
<td>9</td>
<td>30</td>
<td>30%</td>
</tr>
<tr>
<td>#3</td>
<td>Feb 12</td>
<td>1</td>
<td>0</td>
<td>--</td>
<td>1</td>
<td>30</td>
<td>3%</td>
</tr>
<tr>
<td>#4</td>
<td>Pending</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>0%</td>
</tr>
<tr>
<td>#5</td>
<td>Pending</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>24</td>
<td>22</td>
<td>--</td>
<td>45</td>
<td>150</td>
<td>30%</td>
</tr>
</tbody>
</table>

30% accomplished, 30% through designated time period, Jan 1-May 30

Table 2: Experts
3.5.3.1. Narrative on Expert #1

Expert #1 has spent his first seventeen sessions working on the definition of paternalism. He conceives of his task as an extension of work that he has been doing for the past several years and hopes to produce a publishable paper. He took three sessions to read through the material on definition. He then began writing with an introductory session laying out the history of the concept of paternalism and the reasons it requires justification. He left this section roughly sketched and moved into a section reviewing the various authors' definitions where he realized that he needed to define certain auxiliary concepts such as coercion and consent, in one case reviewing material outside of the corpus. He has also started a catalogue of the cases offered by each author and is trying to extract their common features. A third task he has taken on is writing his own definition. Currently, he shuttles back and forth between these three tasks: the introduction, the catalogue of cases, and his own definition.

3.5.3.2. Narrative on Expert #2

Expert #2 is an instructor in the Department of Philosophy at Carnegie-Mellon University. A specialist in the area of decision theory, he has been at CMU for two years and has his Ph.D. in progress at MIT. He has been working for about twelve hours on the project. He began by reading the corpus carefully, questioning it as he proceeded and trying to formulate his own definition. Last week, he began writing, feeling the need to develop his own examples and non-examples of paternalism, first to instantiate the concept for himself, and second, to counter what he feels are incorrect examples in some of the articles in the corpus. Following the development of these examples, he plans to begin writing his definition.

3.5.3.3. Narrative on Expert #3

Expert #3, a faculty member of the Department of Philosophy at the University of Pittsburgh, specializes in historical philosophy. He has just begun to read the corpus on paternalism, moving at a rapid pace. In four hours of work he has read through most of the corpus, made extensive notes, and appears almost ready to begin the writing task. When interviewed, he discusses the issues presented by the corpus, his interpretations of them, and his formulation of the writing problem, all in a manner that makes it obvious that he enjoys this type of intellectual exercise. He has adapted a strategy of working on the task in small chunks of fifteen minutes, fitting approximately eight of these chunks into a typical
working day. He said that he "works in short spurts and thinks about the problem constantly." To compensate for this behavior he agreed to write down and turn in any thoughts he has while not in a protocol situation.

Expert #3's only difficulty has been with the protocols. At his initial meeting with members of the team, he expressed reservations about never having thought aloud before. His first session, however went remarkably well and he seems to have adopted to the process with fewer problems than he anticipated.

3.5.4. A Study of Novices

In developing our Stage II curriculum, we have decided to observe two novices work on the same task as the experts, as much as possible on their own. These novices are second-semester freshmen, neither of whom have taken the introductory philosophy course. They were the most conscientious (though not the most intelligent) of the applicants we had for a work contract of up to 50 hours, 5 hours a week for 10 weeks. One appears to be an above average student; the other about average; both are female.

Thus far, we have completed about 25 hours of protocols and interviews with these two novices, approximately 25% of our goal. We are slightly behind because we had some difficulty locating a second novice at the beginning of the semester. Currently, each novice is working entirely on her own in the task, although we are prepared to intervene if serious problems arise which prevent them from making progress.

<table>
<thead>
<tr>
<th>Informant</th>
<th>Date Started</th>
<th># of Protocols</th>
<th># of Interviews</th>
<th># of Observations</th>
<th>Total Hours as of Feb 13</th>
<th>Expected Hours by April 30</th>
<th>% Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>Jan 22</td>
<td>12</td>
<td>10</td>
<td>--</td>
<td>21</td>
<td>50</td>
<td>42%</td>
</tr>
<tr>
<td>#2</td>
<td>Feb 8</td>
<td>3</td>
<td>1</td>
<td>--</td>
<td>3.5</td>
<td>50</td>
<td>7%</td>
</tr>
</tbody>
</table>

15 11  --  24.5  100  25%

25% accomplished, 27% through designated time period, Jan 21-April 28

Table 3: Novices
3.5.4.1. Narrative on Novice #1

Novice #1 has worked for 11 sessions (approx. 20 hours) and is currently about mid-way through the task. She found the experimental situation disconcerting at first, due to a lack of familiarity with the genre of philosophical argument and the difficulty of speaking aloud while reading and writing. She encountered further difficulties with the think-aloud technique when she moved from primarily reading into primarily writing. By the end of her fifth session, however, she had settled comfortably into the task, and she continues to make steady progress.

Our interviews reveal that Novice #1 is approaching this task as she would any other college writing assignment. She began by reading the first half of the corpus (three articles on the definition of paternalism), underlining important points as she read and jotting notes to herself about the material: questions about difficult points, reminders to reread problem sections later, examples that connect the concepts discussed in the readings to her own experience. It is noteworthy that Novice #1 has been able to recognize examples of paternalism in contexts more familiar to her than those in the readings, e.g. the university's policy of requiring students to enroll in a standard meal plan and the law requiring safety goggles in her father's plant.

Novice #1 used 3 X 5 cards to take notes from the readings and from her earlier jottings and then sorted through the notecards to determine the categories to cover in her paper. She then organized these points into a rough outline and proceeded to elaborate on each in writing a draft of the first half of the paper, the section on definition. After writing what she calls a "general-to-specific" introductory paragraph, Novice #1 presents her own definition of paternalism, synthesized from definitions given in the readings. The next sections offer examples from the readings and from Novice #1's own experience to illustrate the key issues raised in her definition: responsibility, consent, and the question of morality.

In short, though this is perhaps the most challenging writing task Novice #1 has faced, she does not seem to perceive it as qualitatively different from other school-writing assignments and had no trouble deciding how to proceed. At this stage we cannot know how far her standard school-writing strategies will take her on this task. She has been able to identify some critical issues in defining paternalism, but at present these issues are largely separate and unintegrated in her draft.
3.5.4.2. Narrative on Novice #2

Novice #2 has been working for about eight hours now, over two weeks, and has made steady progress. She has almost completed her reading of the corpus and is starting to take notes. Although she had been reading without interpreting or evaluating, she is now beginning to question the material and to try to formulate a definition of paternalism that works for her. The topic has proved to be interesting to her, particularly since she has faced a family situation in connection with her grandfather's terminal illness that she now realizes is paternalistic. She is thus gaining a greater understanding of the problems inherent in trying to define and justify paternalism and is able to bring this personal experience to bear on the issue. I expect that she will start writing a draft quite soon.

3.5.5. A Study of A Class

We have chosen to study the interaction of the teacher & his class in three ways. First, we interview Kaufer, the teacher. Second, we observe the class. Third, we interview each of his twenty-eight students once during the semester at a rate of two per week. Thus far we have observed five weeks of classes, 35 hours of observation, or approximately 26% of the 89 hours we expect to collect.

<table>
<thead>
<tr>
<th>Informant</th>
<th>Date Started</th>
<th># of Protocols</th>
<th># of Interviews</th>
<th># of Observations</th>
<th>Total Hours as of Feb 13</th>
<th>Expected Hours by April 30</th>
<th>% Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teacher</td>
<td>Jan 15</td>
<td>3</td>
<td>5</td>
<td>--</td>
<td>11</td>
<td>40</td>
<td>28%</td>
</tr>
<tr>
<td>Class</td>
<td>Jan 15</td>
<td>--</td>
<td>--</td>
<td>6</td>
<td>9</td>
<td>35</td>
<td>26%</td>
</tr>
<tr>
<td>Students</td>
<td>Jan 17</td>
<td>--</td>
<td>7</td>
<td>--</td>
<td>3.5</td>
<td>14</td>
<td>25%</td>
</tr>
</tbody>
</table>

26% accomplished, 32% through designated time period, Jan 15-April 28

Table 4: A Teacher and His Students

During the first three weeks of the semester, we collected Monday protocols of Kaufer planning for Tuesday's classes, observations of Tuesday's classes, and post-class interviews with Kaufer immediately after those classes. After three weeks, we found that Kaufer's planning protocols were not giving us enough insight into his plans. We also found that important conceptual developments were occurring in the unobserved Thursday classes.
Consequently we decided to revise the observation schedule by substituting a pre-class interview for the planning protocol, and extend this new schedule to cover Thursday’s classes. The summary which follows draws on material gathered according to both schedules. Some of Kaufer’s reoccurring concepts are in italics.

3.5.5.1. Narrative of the Teacher and his Class

On the first Monday, January 14, Kaufer prepared for Tuesday’s class by grading the definitions of paternalism he had asked students to write the week before. Over several hours, he generalized on their strengths and weaknesses and excerpted some examples to use with an overhead projector. In class on Tuesday, he began with his generalizations, but quickly switched to specific examples to make his points clearer. He felt that students’ definitions had been too much the product of a first reading.

In the second week, Kaufer asked his students to summarize and evaluate the Dworkin’s article on definition. The summaries he received on Monday did not satisfy him, but he couldn’t easily say why. Instead of spending time trying to generalize about the problems, Kaufer created a handout on how he went about summarizing Dworkin. The handout showed Kaufer building from a list of the points covered by Dworkin, to their significance or function in the article, next to a hierarchy relating these points, and finally to a summary presenting the major aspects of this hierarchical structure. He asked the students to use this same procedure to write their summary and evaluation of Gert & Culver for the following Monday.

In the third week, Kaufer found that students’ summaries had improved, but that they bore little relationship to their evaluations. Students did not realize that reading, summarizing, and evaluating were cumulative, building on each other. Kaufer created a handout showing the relationship between these levels, building towards the top:

   Your values
   The authors’ values
   The authors’ plans
   The authors’ points
   The authors’ propositions

In their paper on Childress for the following week, he asked his students to show the process of creating their summaries, rather than just giving the summaries themselves.

In the papers Kaufer received on the fourth Monday, students tried to state the
significance of Childress’s points in addition to giving a summary. These statements showed a lack of understanding of what they had been reading: their summaries had been masking un-understanding in vague formulas. In class, Kaufer explained the importance of the missing skill of metaknowledge or knowing what you know and don’t know, by going over several examples from students papers. The class was receptive to this unmasking and Kaufer asked them to bring a list of things they still didn’t understand on Thursday and to write a paper explaining their own metaknowledge in the course for the following Monday.

In planning for Thursday’s class, Kaufer was frustrated by his inability to teach metaknowledge directly through a heuristic. He hoped instead to model for them the skill of metaknowledge by finding the common concerns in the questions they brought him. In class, their questions did not cohere around common concerns and revealed a deep need for structure. Kaufer decided to give some structure by explaining what a good definition is. According to Kaufer, a good definition should explain why there are differences of opinion on an issue rather than deciding it one way or the other.

In the following Tuesday’s class, Kaufer continued to respond to students’ request for more structure by reviewing problems that several had brought up in their papers. He also tried to show the undergraduates in the class how the prior frameworks that some of the graduate students had brought to the class were misleading rather than beneficial. Finally, he tried to outline and stabilize a framework that the whole class could assume going into the second part of the course on the justification of paternalism.

3.5.5.2. Narrative of Selected Students

Students in the class seem to divide according to two characteristics. The first differentiates graduate from undergraduate students, and the second those who took the course to fill a requirement from those who took it out of interest or choice. We have interviewed students with all combinations of these characteristics and have found that they expressed a variety of perceptions about the learning processes they are experiencing.

Responses range from those of an undergraduate student who took the class only to fill a requirement and finds what he is covering “impractical” to someone who is a Ph.D. student in the English Department and who finds “the heuristics presented in the class just great.” There is also a middle group composed of the more talented and ambitious of the undergraduates -- typified by a woman who said that “having to think this way, having to
push myself is teaching me a lot” -- and by the less motivated of the graduate students, one of whom said “the course was practical, it fit my schedule.”

3.6. Plans for Year 2

Figure 2 below details the time-line of activities for the data-gathering team over the four years of the WARRANT project (three years of work spread over four funding years). Year 1 activities include collecting and analyzing the data on the Stage I curriculum, and developing the first component (organization) of the Stage II curriculum. These have already been described in some detail in the section on First-Year Outcomes. In this section, we will describe our data collection and analysis plans for Year 2.

Year 1

Spring, 1985
- Collect data on Stage I Curriculum

Summer, 1985
- Analyze Stage I Data
- Develop Organization of Stage II Curriculum

Year 2

Fall, 1985
- Develop Stage II Advice and Models
- User test Notecarding facility

Spring, 1986
- Develop Stage II Plans
- Collect data on novices using Stage II Advice and Models
- Collect data on Stage II Curriculum in the classroom
- User test prototype of Advice and Plans facilities

Summer, 1986
- Analyze Stage II Data from classroom and novices
- Collect data on educators' judgements on Expert and Novice Models

Year 3

1986
- Develop Stage III curriculum for WARRANT

Year 4

Fall, 1987
- Classroom testing of the WARRANT system

Figure 2: Time-Line of Data Collection and Analysis
3.6.1. Fall, 1985

In the Fall of 1986, we will develop the advice and models required by the Stage II curriculum from the observation and protocol data collected the spring before. Specifically, this will involve us in comparing the coded transcripts of the various experts, novices, and the teacher & his students to find common strategies and their variations. We also anticipate being ready to run user-tests on the first facility of the WARRANT system, notecarding.

3.6.2. Spring, 1986

In the Spring of 1986, we will complete our Stage II development work by putting together the overarching plans uniting the advice and models developed in the Fall. We will also begin to collect models of freshmen trying to take the Stage II advice, the novice models that will eventually appear in the WARRANT system. Our classroom test of the Stage II curriculum will take place in this semester, and we hope to run user-tests on a prototype of the advice-plans facilities.

3.6.3. Summer, 1986

In the summer of 1986, we will code and analyze the data collected from the classroom test and novices, and solicit educators' judgments of the expert and novice models. At this point, we will be ready to pull all the pieces of the WARRANT data base together, the activity which will occupy us in Year 3.
4. System Design and Implementation

This section describes the system design and implementation activities for the WARRANT project. First, it reviews the design and the development context of the WARRANT system, with particular attention to details that have changed since our original proposal. Next, it narrates our activities for the first two months of the WARRANT project. The major outcome after two months is a preliminary specification for one component of the WARRANT system. Finally, it outlines our plans and expected outcomes for the remainder of the first year and concludes with our plans for Year 2 and 3.

4.1. Proposed Design

4.1.1. Design Goals

In our original proposal, we outlined the following design goals for a computer-based environment that would foster the development of reading, writing and reasoning skills:

- flexible structure -- researchers observe that both experts and novices vary widely in how they work and what tools they find useful; teachers observe that novices require some degree of structure in order to develop skill. An environment with flexible structure is necessary to accommodate observed variety and to explore the amount of structure that should be provided in order to foster development.

- integration of tools -- while reading, students should have access to tools for reasoning and writing in response to what they read. While writing, students should be able to use the writing that they generated during reading. They should be able reread an on-line text without having to suspend their writing activities.

- easy to learn and to use -- students are burdened with heavy cognitive demands, and don’t need the added burden of working with a system that is not easy to learn and to use.

We have kept these design goals, but in working on design details, we have recognized the interaction of the goals. For example, the integration of tools must be flexible. A facility for taking notes while reading should be integrated with the facility for reading on-line text; however the facility for taking notes should be useable for working with off-line texts as well.
• Sun Workstation -- prototype hardware for the Andrew System
• Berkeley 4.2 Unix -- the underlying operating system
• Vice -- Andrew file system to support networked personal computing
• Virtue -- Andrew system software to support window-management for a high-resolution bit-mapped display in a multi-process environment.

4.2.1. Sun Workstation

The Sun workstation has the following hardware characteristics:

• 32-bit architecture CPU
• 2 megabytes main memory
• 16 megabytes virtual address space per process
• 10 million bits/second Ethernet local area network interface
• 9-slot Multibus card cage (5 slots used)
• deskside pedestal unit for local mass storage
• 19” high-resolution (1152 x 900) bit-mapped monochrome display
• keyboard
• optical mouse

The Sun workstation represents a departure from the hardware on which we proposed to develop our system. In our original proposal, we indicated that we would be developing on Peaches, an IBM prototype machine. For internal reasons, IBM did not deploy the Peaches, and the ITC chose the Sun as the prototype hardware for the Andrew System.

We do not foresee that this change will have any effect on our project. The Peaches were themselves intended as prototype hardware, and the design of the Andrew software anticipated the need to transport application programs to new environments. It is expected that the workstations eventually deployed on campus will have similar, but not identical, capabilities as the Suns (e.g., the screen-size could be different). Consequently, the ITC has isolated hardware dependencies in their system software. Application programs such as WARRANT will be able to run on any machine which runs the Andrew system; in turn, the
Manager; they do not interact with the hardware directly. The design maximizes the device independence and portability of application programs.

4.2.4.2. The User Interface Libraries

At the present time, the ITC has designed two user interface libraries, a Window Manager library and a Base Editor library. The libraries provide all application programs with a standard, convenient and efficient user interface, and represent the language by which application programs communicate with the Window Manager process. A detailed description of the Window Management and base Editing capabilities were included in the original proposal, and will not be reiterated here.

4.2.4.3. Relation of Virtue to Warrant

These capabilities will form the basis for the integrated set of components which we proposed for WARRANT. Multiple windows will allow us to have ReadText, NoteCard, Advice and Plan windows on the screen at the same time. The tools for window management and editing, tools which are undergoing testing for ease of use and learnability, will allow us to concentrate on the functionality of our applications and speed our development efforts. Figure 4-1 below, a xerox reduction of a screendump we generated from the Andrew system, illustrates the speed at which we can generate a mock-up for the user-interface.1

The screen in Figure 4-1 has four functional areas, or windows: A Read Text window for reading on-line texts, A Read Notes window for browsing and organizing note cards, and two NoteCard windows for displaying and composing note cards. The initial layout is determined by the application program and the Andrew window manager process. The user is free to rearrange and resize the windows. It is, of course, possible to have additional windows on the screen.

Figure 4-1 illustrates multiple windows within the Andrew System. Application programs can request the Window Manager process for particular screen layouts, but the ultimate control of the layout is up to the user. The Figure also illustrates the speed at which we can generate a mock up for the user-interface. It also allows us to keep a desirable separation of the user-interface implementation and the application implementation.

1It is important to note that the Read Text, Read Notes, and Note Card programs pictured in Figure 4-1 are not functional except for the user-interface display; we generated them as exercises in learning the Window Manager and Base Editor libraries.
Andrew system has the potential to run on any machine which runs Berkeley 4.2 Unix and has hardware capabilities similar to the Suns. As we argued in the original proposal, this design gives the WARRANT system a good potential for widespread use on campus, and for dissemination to other colleges in the country.

4.2.2. Berkeley Unix 4.2

In order to implement the Vice networked shared file system, the ITC modified the Berkeley Unix 4.2 kernel. A modification of the operating system, of course, could adversely affect dissemination potential. The modification, however, only concerns the local-area networking of the personal computers. The Virtue software runs without the modification, and the initial distribution of Andrew to other schools, which is now underway, does not require the modified kernel.

4.2.3. Vice Shared File System

The Vice shared file system is at the heart of the campus networking for the Andrew System. However, for the most part, users of the Andrew system do not need to be aware of it. The Virtue system is independent of Vice. Other sites can run Andrew with their own networking schemes, or run without the workstations being networked. The initial distribution tapes of Andrew to other off-campus sites do not contain Vice.

4.2.4. Virtue Software

4.2.4.1. The Window Manager

At the heart of the Andrew system software is the Window Manager program. Normally, one Window Manager process runs on each workstation, invoked automatically when someone logs in. The Window Manager process manages the workstation display, tiling the screen with multiple windows and mediating between the application programs in those windows and the user of the workstation. Application programs running under Virtue do not interact with the user directly. They interact with the Window Manager process, requesting the process to allocate resources (e.g., space on the screen) and to draw things. Because the Andrew system is a multi-process environment, the Window Manager program can be juggling requests from a number of application programs.

Andrew application programs mediate their interaction with the user through the Window
Discussions of paternalism are often marred by the failure to consider the wide variety of paternalistic acts. Thus Gerald Dworkin in his article "Paternalism" says: "By paternalism I shall understand roughly the interference with a person's liberty of action justified by reasons referring exclusively to the welfare, good, happiness, needs, interests, values of the person being coerced. All Dworkin's examples are of laws or regulations which he considers paternalistic. Though he does recognize that there is such a thing as "parental paternalism" he simply assumes that it will always involve the parent's attempt "to restrict the child's freedom in various ways." Paternalism in law doubtless does involve interference with liberty most of the time, but this is due to the nature of law, not to the nature of paternalism. The first of the above quotations also suggests that Dworkin incorrectly regards interfering with a person's liberty of action as entailing that the person is being coerced. The following example shows that an adequate account of paternalism must allow not only for paternalistic action in which no person is being coerced but also for paternalistic action which does not involve interfering with anyone's liberty of action.

Mr. W., a member of a religious sect that does not believe in blood transfusions, is involved in a serious automobile accident and loses a large amount of blood. On arriving at the hospital, he is still conscious and informs the doctor of his views on blood transfusion. Immediately thereafter he faints from loss of blood. The doctor believes that

Dworkin, G.

- definition paternalism
- distinguishes pure/impure paternalism
- examples - no justification
- existing vs. future paternalism
- relation to liberty of persons
- definition - refinement
- relation to harm principle
- vs. compulsion for collective interest
- relation to individual's values

Cort, B. & Culver, C.

Re. Dworkin's definition
religious sect example

Figure 4-1: Andrew System Example Screen Layout for Warrant
4.3. Activities and Outcomes After Two Months

The following describes the activities and outcomes after two months on the WARRANT project. The list below summarizes the accomplishments.

- completed the project team
- began learning the Andrew System
- completed a preliminary functional specification for one component of the WARRANT system.
- completed an initial user-interface implementation for a subcomponent.
- explored designs and implementation for other components.

4.3.1. Completing Personnel Requirements

To complete our project team, we hired a programmer, Richard Chimera, who began work January 7. Chimera has a B.S. degree in Applied Math/Computer Science, May 1984, Carnegie-Mellon University. Before joining our project, he worked at System/Technology Development Corporation, integrating software tools for system reliability evaluation into a menu-driven workbench. In addition to a solid background in software engineering, Unix and C, Chimera, while a student, worked as a User Consultant, and in his interview he demonstrated a superior understanding of inexperienced users' needs.

Our original plans were to hire a more experienced person who had demonstrated the ability to carry out a large-scale programming project. Changes in the final budget allocations, however, precluded our doing so. We did, however, consult with applicants' references, asking them to estimate the ability of applicants to carry out our project. Chimera received high ratings.

In addition, we have begun two on-going activities designed to ensure that the system development will have a successful outcome despite a less experienced programmer. First, Neuwirth has been working closely with Chimera, setting performance goals, providing models for external specifications, and actively working through the consequences of design decisions with him. Second, we are consulting with our Information Technology Center (ITC) liason, Mike Kazar, a Ph.D. in Computer Science from Carnegie-Mellon University, who is
providing technical assistance when required.²

4.3.2. Learning the Andrew system

Before we began detailed design specifications for WARRANT, our first step was to learn the Andrew System.

We received a Sun workstation on February 6. Before that time, Neuwirth and Chimera worked on public Sun workstations located in the ITC. Since receiving our Sun, all principals in the WARRANT project have become familiar with the user interface for the Andrew System.

We will be using C and some Unix constructs in our initial implementation of WARRANT. Since both Neuwirth and Chimera knew Unix and C, we did not need to do much here except learn about socket connections, which we are considering using in the implementation of WARRANT for interprocess communication between the WARRANT components.

We will be using the ITC Window Manager and Base Editor libraries in the implementation of WARRANT. Chris Neuwirth, in work last fall for CMU's Communication Design Center, has written a Programmer's Guide to the Window Manager library, and has been asked to develop a similar guide for the Base Editor library. Because of this work, done partly in anticipation of our FIPSE grant, Neuwirth was already experienced with the Andrew libraries before the grant period began. Chimera has been learning the libraries in January-present.

In conjunction with learning the libraries, Chimera has been attending weekly meetings of the Author's Guild for Andrew, a forum jointly run by the Information Technology Center and the Center for Development of Educational Computing in which people developing educational application programs for the Andrew system can discuss issues and problems and exchange ideas.

²The ITC has assigned a liason to each recipient of a prototype workstation. The liaisons were chosen for goodness of fit between their backgrounds and particular faculty projects. Kazar is currently working on the text-editing interface for the Andrew system, and is an ideal person for our project.
4.3.3. Functional Specification

We have organized the User Interface Design specification process according to the following procedure:

- discussion of facilities and needed capabilities, with special emphasis on typical tasks the system is intended to facilitate
- initial specification by Chimera or Neuwirth
- meetings between Neuwirth and Chimera to discuss functionality and implementation
- revision of specification
- weekly meeting with Principals to discuss functionality
- revision of specification

Our initial efforts for the functional specification have focused on the Note Card component for the system, since the user testing of this component is least dependent on having the available the curricular materials from the data collection and analysis effort. The detailed specification thus far for the Note Card component is included in Appendix A. Briefly, we have divided the Note Card component into two main subcomponents: A Note Card window in which students can take notes and view them, and a Read Notes window in which students can browse through a collection of notes and organize them. Dividing these functions will allow students to have multiple note cards on the screen, yet have a succinct way of viewing collections of note cards. For the functionality of the Read Notes program, we have concentrated on providing facilities for grouping notes by topics, by sources, etc., and for forming subgroups of note cards and putting the note cards into different orders.

4.3.4. Initial Implementation Architecture

The architecture for implementation will have the following, logical components:

- data base files containing representation of WARRANT objects, e.g., documents, note cards, comments, plans, and advice.
- memory structures containing the in-core representation of the objects in WARRANT.
• window and base-editor interface in which the users can view and work with the objects.

• printed representations of the WARRANT objects.

• data collection facilities for recording user's interactions with WARRANT.

We are keeping in view the goal of building the system so that teachers can build and modify WARRANT objects, thus tailoring it to their needs, but the initial implementation will not include a WARRANT maintainer's interface.

The system will be implemented initially in C, at the present time the best supported language for the Andrew System. The implementation will utilize the Window Manager and Base Editor libraries. We have not yet decided on the data base system which we will use for the file representation of WARRANT objects. We are considering using GRITS, a data base access facility which is part of the Andrew System. A primary advantage of GRITS is its representation of data as ASCII files so the data is accessible to other text editors and programs. A primary disadvantage is its prototype nature, since it is not as well-developed as some commercial systems which already run under Unix (e.g., INGRES). We will try to use GRITS, since using a commercial system would require other schools to obtain licenses for the system, and thus limit our dissemination potential.

We have a window and base-editor interface implemented for one component of the Note Card facility, a program to allow users to compose and display a Note Card. A screen-dump of it is included in the appendix.

4.4. First Year Outcomes

During the rest of the project year, we will continue to concentrate on making WARRANT a functioning, integrated computer system. Before outlining our detailed plans, we first review the steps we will be following for the entire project:

• top-level design with emphasis on needed capabilities, ease of use

• implementation of prototype

• refinement of prototype
informal tests with experts and novices; solicitation of feedback

release to friendly users (i.e., members of Author’s Guild for Andrew, Center for Development of Educational Computing, Graduate and Faculty in English); solicitation of feedback

revision based on feedback with new release; process reiterated

more formal assessments of usefulness and acceptability to users with special focus on its impact on the execution of the tasks with which it is designed to assist

service functioning for extended period in prototype environment with automatic collection of use statistics

Documentation of the system has been an on-going effort from the start of the project and will continue throughout the project life.

The following steps are outside the scope of the three-year project, but steps we intend to take, should the project have a successful outcome:

- assessment of large-scale usefulness
- program changes to allow widespread distribution
- general release and marketing with plans for maintenance and updates

For the remainder of Year 1, we plan

Spring, 1985  Complete design specification for Note Card component

Implementation of Note Card component

Complete design specification for Read Text & Compose Text components

Implement Read Text & Compose Text components to level necessary for interaction with Note Card Component, including inter-process communication links

Informal tests with experts and novices of Read Text, Compose Text & Note Card

Summer, 1985  Revisions of Read Text, Compose Text & Note Card

Design specifications for Plans & Advice
4.5. Plans for Year 2 and 3

Fall, 1985
- Design of Syllabus component
- Informal tests with experts and novices for Plans & Advice
- Refinement of Plans & Advice based on feedback
- Release of Plans & Advice to friendly users
- Refinement of Plans & Advice based on feedback
- More formal assessments of usefulness and acceptability to users of Read & Compose Text and Note Cards, with special focus on novices vs. expert use in the absence of plans and advice

Spring, 1986
- Design of Comment facility
- Implementation of Syllabus
- Informal tests of Syllabus with experts and novices
- Refinement of Syllabus
- More formal assessments of Plans & Advice, with special focus on the usefulness of plans & advice within the environment Revisions of Note Card, Read & Compose Text based on assessments

Summer, 1986
- Design of Statistics gathering and analysis routines
- Implementation of Comment facility
- Informal Testing of Comments
- Revision of Comment based on feedback
- User Release of Syllabus
Revision of Syllabus

Formal Assessment Syllabus with special attention to ease of access

Revision of Plan and Advice based on assessment

**Year 3**

**Fall, 1986**

Implementation of Statistics log & analysis facilities

Release of Comment facility to friendly users

Revisions based on feedback

Formal assessment of Comment facility with special attention to the use that teachers and students make of it

Revision of Syllabus based on assessment

**Spring, 1987**

Revisions to system

Informal testing of statistics facility

Revision based on tests

Formal assessment of WARRANT without the statistics log facility

Revision of Comment facility

**Summer, 1987**

Pilot study of WARRANT system under statistical logging with friendly users

Revision of WARRANT system

Assessment of WARRANT system with statistical logging with special focus on the usefulness of the information

Revision of WARRANT

**Year 4**

**Fall, 1987**

Service functioning for extended period in prototype environment with automatic collection of statistics

The following table summarizes the activities and plans for the project.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Spring 85</td>
<td>Note Card Read/Compose Text</td>
<td>Note Card driver implementation for Read/Compose Text</td>
<td>Note Card Read/Compose Text</td>
<td>Note Card Read/Compose Text</td>
<td>Note Card Read/Compose Text</td>
<td>Note Card Read/Compose Text</td>
<td>Note Card Read/Compose Text</td>
<td>Note Card Read/Compose Text</td>
<td>Note Card Read/Compose Text</td>
</tr>
<tr>
<td>Summer 85</td>
<td>Plan Box Advice Box</td>
<td>Plan/Advice Read/Compose text Integration</td>
<td>Note Card Read/Compose Text</td>
<td>Note Card Read/Compose Text</td>
<td>Note Card Read/Compose Text</td>
<td>Note Card Read/Compose Text</td>
<td>Note Card Read/Compose Text</td>
<td>Note Card Read/Compose Text</td>
<td>Note Card Read/Compose Text</td>
</tr>
<tr>
<td>Fall 85</td>
<td>Syllabus</td>
<td>Plan/Advice</td>
<td>Plan/Advice</td>
<td>Plan/Advice</td>
<td>Plan/Advice</td>
<td>Plan/Advice</td>
<td>Plan/Advice</td>
<td>Note Card Read/Compose Text</td>
<td>Note Card Read/Compose Text</td>
</tr>
<tr>
<td>Spring 86</td>
<td>Comment</td>
<td>Syllabus</td>
<td>Syllabus</td>
<td>Syllabus</td>
<td>Syllabus</td>
<td>Syllabus</td>
<td>Plan/Advice</td>
<td>Note Card Read/Compose Text</td>
<td>Note Card Read/Compose Text</td>
</tr>
<tr>
<td>Summer 86</td>
<td>Statistics Log &amp; Analysis</td>
<td>Comment</td>
<td>Comment</td>
<td>Comment</td>
<td>Syllabus</td>
<td>Syllabus</td>
<td>Plan/Advice</td>
<td>Note Card Read/Compose Text</td>
<td>Note Card Read/Compose Text</td>
</tr>
<tr>
<td>Fall 86</td>
<td>Statistics Log &amp; Analysis</td>
<td>Comment</td>
<td>Comment</td>
<td>Comment</td>
<td>Comment</td>
<td>Comment</td>
<td>Syllabus</td>
<td>Note Card Read/Compose Text</td>
<td>Note Card Read/Compose Text</td>
</tr>
<tr>
<td>Fall 87</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Warrant &amp; Statistics Log</td>
</tr>
</tbody>
</table>

WARRANT: SYSTEM DESIGN & IMPLEMENTATION SCHEDULE
5. Dissemination

5.1. Books

5.1.1. Critical Reasoning/Writing Pedagogies

Covey and Kaufer have decided that the slants of teachers of philosophy and teachers of composition are sufficiently different to merit the writing of separate pedagogies. Covey will accommodate his book to the background and interests of the teacher of philosophy who wants to teach critical reading and writing. Kaufer, working in association with Neuwirth and Geisler, will author a critical reading/writing book addressed specifically to teachers of composition. We feel this separation will give the WARRANT project maximum coverage in both fields. Appendix 1 contains a chapter by Kaufer on giving advice for critical skills.

5.1.2. Approaches and Methods for Curriculum Planners

Our data-gathering effort, spread as it is among experts and novices working through a corpus on paternalism and a teacher teaching the same corpus (with the same assignments, though broken down) gives us a unique vantage from which to raise such general educational questions as What is a curriculum? What should its goals be vis-a-vis novice and expert performance? and, practically, if you don't have half a million dollars from FIPSE but still want to use the WARRANT system, how do you build a curriculum out of a set of readings and assignments that you want to teach? We realize that these questions are extremely important for us to address, for already members of the literature, history, and social science faculty have approached us with an interest in using WARRANT for their own curricular assignments. We have conceived a book that will discuss the empirical and educational issues involved both in designing a reading/writing curriculum and in designing one specifically for the WARRANT environment. Cheryl Geisler, head of the data collection team, is taking the lead on this book.

5.1.3. The WARRANT System

Our plans have not changed for a book on the WARRANT system which will detail the educational assumptions we have built into it, what we learned through user testing, case studies, and, perhaps, controlled studies with different students using different versions. Chris Neuwirth, who heads the system development team, is taking the lead on this project.
5.2. Conferences and Articles

In addition to these books which represent a long range dissemination strategy, the WARRANT team has already started attending conferences and authoring articles to get the word out about WARRANT. We list here the activities completed or scheduled so far.

<table>
<thead>
<tr>
<th>Date</th>
<th>Name(s)</th>
<th>Event/Activity Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>November, 1985</td>
<td>Preston Covey, David Kaufer</td>
<td>FIPSE Project Directors Meeting</td>
</tr>
<tr>
<td>February, 1985</td>
<td>Cheryl Geisler</td>
<td>Paper to CALICO, Computer Assisted Language Learning and Instruction Consortium</td>
</tr>
<tr>
<td>February, 1985</td>
<td>Cheryl Geisler</td>
<td>Article invited for CALICO Journal</td>
</tr>
<tr>
<td>March, 1985</td>
<td>Christine Neuwirth</td>
<td>Presentation to Special Interest Group on Computers, CCCC, Conference on College Composition and Communication</td>
</tr>
<tr>
<td>March, 1985</td>
<td>Preston Covey, Peggy Seiden</td>
<td>AAHE, American Association of Higher Education</td>
</tr>
<tr>
<td>April, 1985</td>
<td>Preston Covey</td>
<td>APA, American Philosophical Association, Western Division</td>
</tr>
<tr>
<td>May, 1985</td>
<td>Christine Neuwirth</td>
<td>Computers and Writing Conference, UCLA</td>
</tr>
</tbody>
</table>
6. Bibliography


Appendix I
Giving Advice for Critical Skills

What is the difference between generic and task-specific advice for critical skills? Consider this dialogue, which I had with a student immediately after I had assigned an evaluative essay in my writing class in a previous year:

1. Student: Okay, what do you want me to do again?
2. Kaufer: You’ve read the article by Dr. Houts, right?
3. Student: Right.
4. Kaufer: Okay, I want you to write an evaluation of the article.
5. Student: I thought it was pretty good. Do you want me to write that?
6. Kaufer: If you want. But an evaluation needs to be more than a simple adjective here or there. You have to make an argument for why you reacted as you did.
7. Student: I’m confused. First you said I have to write an evaluation paper. Now you’re telling me to write an argument paper. Which should I write?
8. Kaufer: No, what I’m saying is that to make an evaluation is to make an argument about how you felt about the paper. You can’t say that it’s just pretty good or not pretty good. You’ve got to give reasons for why you believe one or the other.
9. Student: I understand, but I don’t know how to give reasons. I just think the essay is pretty good.
10. Kaufer: One way to give reasons is to look for standards that you think a paper of this type should satisfy. When we evaluate something, we always have standards in the back of our mind. For example, if you tell your friend the dorm food is lousy, you usually have standards in the back of your mind about what good food should have, right?
12. Kaufer: Sure you do. You know about the all great meals you’ve had over the years and if you think hard enough, you could probably list what you think needs to go into a great meal.
13. Student: (Thinking) Meat, maybe...
14. Käufer: That may be too specific. You get a lot of hamburgers at the dorm. Do you like them?

15. Student: No. Not really.

16. Käufer: Okay, are you telling me that all of your great meals had meat in them?

17. Student: Yeah, I guess so.

18. Käufer: Okay, are you telling me that all the meat in your great meals had to be cooked a certain way?

19. Student: Well I think I ate most of them at pretty expensive restaurants...

20. Käufer: That’s getting us off the track a bit...You’ve got to think what those good restaurants DID to make them great meals... Have you ever read a review of a restaurant in a newspaper?


22. Käufer: Does the reviewer ever recommend that you go to a restaurant because it’s expensive?

23. Student: Not really.

24. Käufer: Well, why will he recommend or not recommend a restaurant?

25. Student: He’ll recommend it if the service is good, if the meals are cooked as they should be, the meat tender and juicy, the wine appropriate....

26. Käufer: Yeah, don’t you recognize these as standards that the food critic uses to evaluate the restaurant?

27. Student: Yeah.

28. Käufer: What we would do with a food critic who just wrote in his newspaper column, “I think the restaurant’s pretty good.”

29. Student: (laughs) Probably fire him.

30. Käufer: Yeah. Now I just told you a few minutes ago to evaluate a paper. But you wanted just to tell me it’s “pretty good.” What should I do with you?

31. Student: (laughs) Fire me.

32. Käufer: (laughs) Do you think you understand now what I mean by standards for a paper?
33. Student: Well I think I understand what you mean by standards for evaluating a restaurant. I'm not sure I know what it means to have standards for evaluating an essay?

34. Kaufer: Well one quick way to think of why people evaluate anything is that they want to know if the thing they're evaluating does what it trying to do. The restaurant wants to have first rate service and food so it can attract lots of customers and make lots of money. Right?

35. Student: Right.

36. Kaufer: The food reviewer wants to find out whether the restaurant is living up to these standards. Right?

37. Student: Right.

38. Kaufer: Now, I'm asking you to evaluate Dr. Houts' paper. What he is trying to do?

39. Student: I think he's arguing that I.Q. tests don't really work to test people's intelligence.

40. Kaufer: Do you think he convinces you?

41. Student: Sure.

42. Kaufer: Why?

43. Student: Because he gives a lot of examples of I.Q. questions that seem really stupid. That's why I think his paper is pretty good.

44. Kaufer: If you were to write a review in the newspaper of Houts' article, what would you write?

45. Student: I guess I'd write that his article was good because he shows that I.Q. tests are invalid.

46. Kaufer: Do you think anyone would believe you, just saying that?

47. Student: Not really.

As this dialogue suggests, teaching reasoning to a student writer can be a humorous exercise. It is undoubtedly a challenging, long, arduous, and taxing one. We can begin to appreciate the complexities of the effort by contrasting it with a simpler kind of coaching, namely coaching a student who is trying to solve a puzzle or win at a game. In the gaming context, the student knows what she is supposed to do (i.e., knows the rules for
winning the game or solving the puzzle) even before she begins. She also knows what legal moves can be taken to reach these goals. What she probably doesn’t know—and where she may need coaching—are the best moves for her to choose, given the alternative moves available. Learning optimal, or even effective, strategies for game-playing or puzzle-solving takes practice and, perhaps, coaching. For small games (e.g., tic-tac-toe), where the alternative number of moves is relatively small, learning these strategies only requires a relatively short training period. For large games (e.g., chess), where the alternative number of moves is enormous, such learning, like learning strategies for reasoning during writing, demands an indefinite training period.

The similarities of chess and writing in the indefiniteness of the training period imply that the important differences between teaching reasoning to a student writer and teaching strategies to a student game-player owe not so much to the number of choices the student must make; for in both chess and writing, the student must make too many choices to count. Rather the important differences seem to lie in the greater number of types and varieties of choices the student writer must make. The chess player, as already suggested, needn’t make choices about her goals or what counts as a legal or illegal move. Writers, on the other hand, must choose their own goals and decide for themselves what counts as a “legal” or productive “move” through a writing task. No less important, unlike chess coaches, teachers of writing can and characteristically must span many different domains of knowledge to help their students advance through a single writing task.

We can illustrate these claims about writing from the above dialogue. Kaufer had given the student an assignment to evaluate an article by a Dr. Houts. The student understands something about the word “evaluation,” but doesn’t really know how to internalize the goal of “evaluating a paper.” Much of the dialogue is taken up with Kaufer coaching the student on what it means to set this goal. One can hardly imagine spending so much time coaching a chess player on what it means to capture the opposing king. The dialogue also makes clear that there are no legal or illegal moves, only more or less productive ones. Coupled with this observation is the observation that no domain of knowledge seems necessarily “out of bounds” for the discussion at hand. What began as a session to discuss the evaluation of Dr. Houts’ paper took some twists and turns into dormatory conditions, restaurant reviews and restaurant experiences. It’s hard to imagine a chess coach requiring a student to make such leaps of knowledge to better understand chess strategy. But in this dialogue Kaufer clearly felt that the student’s experience with restaurants might help her
better understand and internalize the goals of the writing assignment.

Besides helping us illustrate the complexities of teaching reasoning to a writer over and above teaching strategies in a gaming context, the above dialogue can also help us sort out some of the different types of writing instruction teachers currently can make available to students. First, there is generic instruction. Generic instruction consists of instruction whose validity does not depend on the context in which it is given. Writing textbooks are characteristically replete with generic instruction about how to write. Such instruction, however, is not limited to textbooks and often occurs in our informal interactions with students. Kaufer’s turns 6, 8, and 10 offer generic information about evaluating an essay that could easily be found in a textbook. The validity of these comments in no way depends upon the fact that Kaufer and the student are six turns deep in a conversation about how to evaluate an article written by Dr. Houts.

There are both advantages and disadvantages of generic instruction. Among the advantages, generic instruction is meant to be easy to understand. Students don’t need to be deep in a specific writing task or context to understand instructions like “evaluating something presupposes applying standards to it,” or “think about the needs of your audience” or “avoid the passive voice if it makes your prose awkward.” With clear examples or analogies, students should be able to understand these instructions and, it is hoped, apply it to their own writing whenever appropriate. Notice that all of Kaufer’s initiatives into topics like dorm food and restaurant critics are attempts at examples or analogies to help the student better understand the relation of standards to evaluations. Moreover, none of these examples or analogies depends upon the fact that the student has been asked to write an evaluation of Dr. Houts or I.Q. tests. One could expect to find comparable examples or analogies in any writing text that sought to teach evaluative writing.

A second advantage of generic instruction is that it is general enough to apply to a wide variety of writing tasks and contexts. There is a grain of truth in the notion that students only need to take one generic writing course and learn from one generic writing textbook. The principles of good writing, taken generically, don’t need to change much, if at all, across writing tasks and competency levels. We should and often do try to hold second-grade writers to the same generic standards of coherence, clarity, correctness, and adaptiveness that we ask of doctoral students. Writing textbooks vary because these standards need to be promulgated to a very diverse group of students; they do not vary
because the standards themselves do. Thus, if a student has a wealth of generic information about "effective" strategies for writing, it's likely this information will continue to serve the student well on writing projects into the future.

However, there are also disadvantages attached to generic instruction and these disadvantages are directly tied to its advantages. While generic instruction may be easily comprehensible because it doesn't require students to bury themselves in the detail of specific writing tasks and contexts, students can only apply such instruction within a specific task and context--namely their own. It is thus one thing for students to understand the generic wisdom of an isolated exhortation like "keep your readers' needs in mind." It is quite another thing for them to apply this exhortation productively when they are faced with a concrete writing task and thus find themselves having to factor it in along with hundreds of other substantive and rhetorical details.

Student writers are thus faced with a serious problem trying to transfer the generic instruction they are given into instruction that can help them on their specific writing tasks. To offset this difficulty, teachers need to rely on a type of instruction that is not generic, but task-specific. Because writing texts have thus far offered only generic, not task-specific, instruction, teachers of writing have had to look elsewhere to deliver it to students. Where they have usually looked to deliver it is in the conference scheduled between the time the assignment is given and the time it is due. The conference is the ostensible forum where teachers can help students transfer the generic instruction of the classroom or the textbook into instruction that will serve their immediate needs.

Unfortunately, the conference has not proven to be a reliable forum to facilitate this transfer. Perhaps it could be reliable if the transfer were a simple linear process. If it were, the teacher could simply search through each generic-type instruction and discuss the implications that each should bear on the student's task. But the transfer is more like a chain reaction of events that impact upon other events far too numerous to predict in advance. Consequently, the implications of the transfer from generic instruction to the student's specific writing task must be continually worked out at every moment of the writing process. Under present conditions, the student is completely left to his or her own devices to work out these implications.

Returning to the introductory dialogue, we can see that while Kaufer's interaction with the
student goes far enough to diagnose and remedy some gaps in the student's generic understanding of the role of standards in the evaluation process, the student's turns 45 and 47 still betray an inability to transfer this understanding to the immediate assignment. At turn 47, it became apparent to Kaufer that if the student were to make any headway on this transfer, it would only be by actually starting to write—an activity beyond the bounds of the conference. Other types of conferences might well have occurred after the student had written a draft. But then Kaufer (or any teacher) would be left to make sense of and evaluate the hundreds of transfer decisions that the student had made on her own. Far from helping the student learn how to make those decisions, Kaufer (or any teacher) can now just be around to appraise their consequences.

While it seems apparent, then, that student writers need both generic and task-specific instruction, the current situation is that they are getting only the former. Conferences with students may fill in gaps on generic instruction left over from the classroom or the textbook, but they don't allow teachers the forum to teach the important task-specific, transfer skills students need during the act of writing.

No less important, the generic instruction that student writers do currently receive is heavily weighted to overlap with the instruction teachers now offer in conferences or the classroom. By interacting with students and diagnosing their problems, teachers are able to develop lesson plans, and it is these lesson plans that ultimately turn into writing texts. The result is that most of the generic instruction that writing teachers now offer is instruction about what to do well before writing needs to begin (prewriting) and well after it is possible to end (editing). While many "process-based" writing texts are now beginning to discuss generic strategies that can be acted upon during the writing process itself, writing teachers still know precious little about possibly generic strategies that serve expert writers well as they write, as they address a specific writing task from moment to moment.

The WARRANT project, including the empirical and systems effort, is designed to add to the teacher's arsenal by,

1. uncovering detailed generic strategies about expert performance during the writing process

2. uncovering task-specific strategies and seeing how they interact with generic instruction during the writing process (at least part of this activity should help us address the transfer problem)
3. providing an environment (replete with electronic note-carding and text-annotating) to make it possible for teachers to discuss and monitor student performance during the actual course of writing

4. providing an environment for testing optimal strategies for delivering instruction to students during the course of writing
Appendix II
Functional Specification for the Note Card Component

The Notecard Program ...................................... 1-4
The Read Notes Program .................................... 1-12
The EditFormat Program .................................... 1-3
The NoteCard Program

The NoteCard program will display a window in which the user can compose or read a note.

To compose a note while reading a text, the user will do the following:

**Action.** In the ReadText window, select the region of text to which the note should apply, then choose Take Note from a pop-up menu.

**Response.** A NoteCard window will appear in a new window, as shown in the figure below.

![Figure 1-1 A NoteCard window](image)

The NoteCard window is composed of three fields which are always exposed to the user’s view: the reference field, subject field and note field. In addition, the NoteCard window has two fields which are normally hidden from view but can be exposed: a set of user-defined classes for the note and the original text to which the note applies.
The Reference field

If the text that the user took the note from has reference information associated with it, the reference field will be automatically filled in with the author's name, title, and other bibliographic information for the text.

If the text does not have reference information, the user may create it by taking the following action:

**Action.** Pop-up the NoteCard program's menu and select Edit Reference.

**Response.** A Reference Editor program will appear in a new window, displaying a set of bibliographical classifications for the text. The possible classifications will follow those available in SCRIBE, a document formatting system, including the following:

- Article
- Book
- In Book
- Phd. Thesis
- Technical Report
- Unpublished ms.

**Action.** The user selects a classification for the text by clicking the left mouse button on a classification box.

**Response.** The Reference Editor program will display the set of fields for the selected classification (e.g., Author, Title, Journal, Year).

**Action.** To fill in a field the user clicks on the left mouse button on a field.

**Response.** The program prompts for the information in a base-editor layout.

To alter the reference classification for a text, the user will do the following:

**Action.** Click on a box representing the classification.

**Response.** The program will display the contents of the fields associated with the classification. The user need only fill in the fields which do not carry over from the previous classification.

The Subject field

Like the subject field in memos, this field is intended as a synopsis of the contents of the note.
The Note field

This is the functional area in which the user will compose a note. The note can be arbitrarily long.

The pop-up menu items that provide the full functionality of EditText will be available. In addition, the following three Menu cards, with menu items specific to the Note Card program, will be available: Top menu, Hide/Expose, Edit Classes.

The following menu items will appear on the Top menu card:

- **Paste.** Allows the user to paste text that has been previously cut or copied.
- **Edit Reference.** Allows the user to create or alter the Reference field. Creates a ReferenceEditor window.
- **Save.** Stores the note. If the storing is successful, the message *Note Card: Note saved.* will appear in the window's response line. If unsuccessful, the message *Note Card: Unable to save note.* will appear.
- **Save & Quit.** Stores the note and quits.
- **Quit.** Ends the NoteCard program and makes the window go away. If the user chooses Quit with an unsaved note card, the following question will appear in the response line:

  Do you really want to quit? no

  The user will press the RETURN key to cancel the Quit action. To continue the Quit action, the user will delete the word no and type y (for yes), then press the RETURN key.

The Hide/Expose menu card will toggle fields which can be hidden from view or exposed to view. If a field is currently hidden, the menu item will allow the user to expose the field. If the field is exposed, the item will allow the user to hide it. The following fields will appear on the Hide/Expose menu card:

- **Classes.** Exposes or hides the list of classes to which the note card belongs. The user can add a note card to a new class, delete a note card from a class, and rename a class (see the Edit Classes Menu card below).

- **Associated Text.** The text to which the note card applies. If the user exposes the text, the Note Card program looks for a process which is running ReadText on the text file. If found, it sends a message to ReadText to display the associated text in a ReadText window. If not found, it creates a new window. If there is no associated file, the following message will appear in the response line: *Note Card: Associated text not found.*
The following menu items will appear on the Edit Classes menu card:

**Add Class.** Prompts the user for the name of a new class for the note card. Duplicate class names not allowed.

**Delete Class.** Prompts the user for the name of a class to delete.

**Rename Class.** Prompts the user for the name of the class to be renamed; then prompts for the new name.
The Read Notes Program

The Read Notes program will display a window in which the user can peruse and organize a group of note cards.

To create a Read Notes window, the user will do the following:

Action. In any program which allows access to Read Notes, choose Notes from the Read menu.

Response. A window similar to the one below will appear on the screen.

```
Dworkin, G.

- definition paternalism
- distinguishes pure/impure paternalism
- examples -- no justification?
- existing vs. future paternalism
- relation to liberty of persons
- definition -- refinement
- relation to harm principle
- vs. compulsion for collective interest
- relation to individual's values

Gert, B. & Culver, C.

- Re. Dworkin's definition
- religious sect example
```

Figure 2-1 A Read Notes window

The Read Notes window displays all the note cards in the user's current directory. In Figure 2-1, the note cards are grouped by the author's last name, e.g., Dworkin, Gert & Culver, etc. Within each group, the note cards are listed in the order in which they were created and they are identified by the subject of each note, i.e., a brief mnemonic for a note card's contents that is normally composed by the user when the note card was created. The user may specify other groupings, identifications and orderings for listing note cards, either temporarily or as a permanent preference. The user may also browse and search through the note cards, forming subgroups and new orderings which can be saved.
Viewing the Contents of Note Cards

The Read Notes program lists note cards, but the contents of the cards are not displayed unless the user asks to view a note card.

To view the contents of a note card from the Read Notes program, the user will do the following:

Select the Note Card

Action. Click the left mouse button on a note card of interest. (The user can use the scroll bar to the left of the note card listings to scroll to note cards that are not visible, and can enlarge the window.)

Response. The note card clicked on will have a box drawn around it, indicating that it is selected. Clicking on the left mouse button somewhere else will deselect the note card.

Choose View from the Top Menu

Action. Move the mouse cursor anywhere inside the selected region, pop-up the menus and choose View.

Response. The note card will appear in a NoteCard Program window. The note card will remain visible until the user chooses Quit or Hide from the menus associated with the NoteCard window.

The user can repeat this procedure to view other note cards, so the user can have as many note cards on the screen as desired, subject to the system limits (currently 20 windows per wm process).
To view the contents of neighboring notecards:

Alternatively, once the user has selected a note card, the selection can be extended to include any number of neighboring note cards in the listing. This is a time-saving procedure for viewing the contents of multiple note cards. It is especially useful when the user has grouped the note cards into convenient listings.

Select and extend the selection

Action. Select the first note card of interest, move the mouse cursor to the last note card in the listing that is of interest and click the right mouse button.

Response. The selection of note cards will be extended to include the first note card through the last note card.

Repeated clicks with the right mouse button on other note cards can extend or reduce the selection. Choosing View from the pop-up menu within the extended selection will cause all the note cards within the region to appear in NoteCard windows.

By browsing through the listing and viewing notes cards, the user can examine particular cards, and cut and paste from the contents of the cards into other texts.
Classifying note cards

Classes play an important role in the Read Notes program: they are used for grouping note cards together in the listings and for forming subgroups of note cards.

In addition to the viewing the listings of the note cards, the user can choose to view the classes to which the note cards belong. New classes can be created, deleted, or re-named, and selected note cards can be added to classes or deleted from them.

The initial set of classes for the note cards are taken from the classes to which the user assigned them when they were created.

To view the classes for the note cards:

Action. Choose Expose from the Classes menu.

Response. A rectangular box labeled Classes will appear at the top of the Read Notes window and any existing classes will appear, displayed with boxes, in the Classes box.

When the user exposes classes, the Expose item in the Classes menu is toggled to Hide. In addition, the menu items Create, Delete, and Rename become available on the Classes card, and remain available until the Classes are hidden.

To create a class:

Action. Choose Create from the Classes menu.

Response. Read Notes prompts in the response line for the name of the new class.

Action. The user types in the new class, and presses RETURN.

Response. The new class appears at the top of the Read Notes window in the Classes box.
To delete an existing class:

**Action.** Choose Delete from the Classes menu.

**Response.** Read Notes prompts in the response for the name of the class that should be deleted.

**Action.** The user types in the name of the class to be deleted, or cuts and pastes it in from the Classes box at the top of the Read Notes window.

**Response.** If the class exists, the class will disappear from the entries under classes. All note cards which were in the specified class will be deleted from the class. If the class does not exist, Read Notes will respond in the message line: Read Notes: Couldn't find the named class.

To rename a class:

**Action.** Choose Rename from the Classes menu.

**Response.** Read Notes prompts in the message line for the name of the class that should be renamed.

**Action.** The user types in the name of the class to be renamed, or cuts and pastes it in from the Classes Box at the top of the Read Notes window.

**Response.** If the class exists, the Read Notes prompts in the message line for the name of the new class. If the class does not exist, Read Notes will respond in the message line: Read Notes: Couldn't find the named class.

**Action.** The user types in the new name for the class.

**Response.** Read Notes updates the affected class in the Class Box. All note cards that were in the old class are in the renamed class.
To add note cards to a class or classes:

**Action.** Move the cursor inside Class box and click on the class (or classes) to which the Note Cards are to be added.

**Response.** The Classes are highlighted.

**Action.** Select the Note Cards of interest, move the cursor inside the selected region and choose Add to Class from the pop-up menu.

**Response.** The selected note cards are added to the highlighted classes.

To delete note cards from a class or classes:

**Action.** Move the cursor inside the Class box and click on the class (or classes) from which the Note Cards are to be deleted.

**Response.** The Classes are highlighted.

**Action.** Select the note cards of interest, move the cursor inside the selected region and choose Delete from Class from the pop-up menu.

**Response.** Read Notes will prompt in the message line with Delete selected notes from classes? no

**Action.** Backspace and type y for yes and press RETURN.

**Response.** The selected notes will be deleted from the specified classes. If a note card has no classification, it is considered to be in the class Unclassified.
Forming sub-groups of note cards

The Read Notes program provides two ways to form subgroups of note cards: asking the Read Notes program to search for cards which meet a particular specification and browsing through the cards and specifying that particular cards should constitute a subgroup. These two methods can be combined, so an initial selection can be automatic, and then hand-tailored.

Searching

To form a group of note cards by searching, the user will do the following:

**Action.** Choose Search from the Listings menu.

**Response.** A Search Notes window will appear on the screen. The window will contain the major search classifications, including the following:

- Author
- Classes
- Subject
- Creation Time
- Modification Time

To specify a search:

**Action.** The user clicks on the search fields and values of interest.

**Response.** The search fields and values are highlighted.

**Action.** In the Search Notes window, the user chooses Start Search from the Top Menu.

**Response.** A Read Notes window appears on the screen which contains the listings for the notes which met the search specification. The title of the Read Notes window is taken from the old notes window. For example, if the old window is entitled *paternalism* the new ReadNotes window is entitled *SubGroup 1, paternalism*. The new ReadNotes window has many of the functionalities of the old ReadNotes window. However, the Listings menu toggles from Search to Save as SubGroup.
Example 1. To search for all the notes on Dworkin that have the class 'definition':

**Action.** The user clicks on Dworkin in the Author box.

**Response.** Dworkin is highlighted.

**Action.** The user clicks on 'definition' in the Classes box.

**Response.** Definition is highlighted.

**Action.** The user chooses Start Search from the Top Menu.

**Response.** All the notes on Dworkin having to do with definition appear in a new Read Notes window.

Example 2. To search for all the notes on either Dworkin or Gert & Culver in the class 'definition':

**Action.** The user does the actions for Example 1.

**Response.** All the notes on Dworkin in the class definition appear in the new Read Notes window.

**Action.** In the Search Notes window, the user clicks on Gert & Culver.

**Response.** Gert & Culver is highlighted; Dworkin de-highlighted. (If the field value is not a unique key, then the user must de-select the previous selection explicitly or it will remain highlighted and be and-ed. e.g., the user could search for notes which are in both 'definition' and 'problematic'.)

**Action.** The user chooses Start Search.

**Response.** The note cards which meet the new specification are added to the note cards in the new Read Notes window.
Name
EditFormat - formats note card listings

Synopsis
EditFormat note_card_filename

Description
EditFormat edits the options associated with the format of note card list­
ings. It can be invoked directly from the Typescript or by the Format menu item on the Listings menu card in the ReadNotes program. If Read­Notes encounters a collection of note_card_files with inconsistent formats, the default will be used.

Format Options

Each option specifies a number of possible formats for the note card list­
ings. There are three major areas of attribution in the EditFormat window: Group, Order and Identification.

Group. Selects the manner in which note cards in the listing should be grouped. Only one can be selected for a given listing. The default is group by author. There is some interaction between groups and identifications. The following are some possible groups:

- **Author.** Groups the note cards by author of the note card's reference.
- **Subject.** Groups the note cards by subject. Changes the default identification from subject to Author, Page Number.
- **Title.** Groups by the title of the note card's reference.
- **Page Number.** Groups by page number of the note card's reference. This is of limited utility with notes from multiple articles.
- **Disable Groups.** The note cards will not be grouped. Display will be by the specified order and identification.

The following additional flag controls the order of the groups themselves:

- **Group Ascending/Descending.** Controls whether the order of the groups is displayed from a-z or z-a; 0-n, n-0.

Order. Selects or modifies the order in which the note cards will be listed. If note cards are grouped, then it controls the order within groups. The following are some possible orders:

- **Creation Time.** Orders notes by the time of their creation.
- **Page Number.** Orders notes by the page number reference.
To save the result of a search:

**Action.** In the new Read Notes window, choose Save as Sub-Group from the Listings menu.

**Response.** The program prompts for a name for the Sub-Group.

**Action.** Type a name for the group, and press RETURN.

**Response.** The program will respond Saved. If the named SubGroup already exists, the program will ask for confirmation. The new subgroup will be added to those in the SubGroup box.

While the new ReadNotes window remains on the screen, an option to copy a note from the old window to the new window appears. To tailor the automatic search, the user can select notes from the old Read Notes window and copy and paste them into the new Read Notes window. Paste only appears as an option on the Top Menu card in Read Notes windows that are SubGroups.
Forming new orders for groups of note cards

As the user works with the note cards in composing a text, an important activity involves forming the note cards into new orders.

To form a new order for a group of note cards, the user will do the following:

Action. Select the note card or note cards of interest and choose Move from the selected region menu.

Response. The cursor changes shape to a hand-pointer.

Action. Click on the left mouse button and while holding it down, move the cursor to the location desired for the note card.

Response. The selected note cards will follow the mouse cursor.

Action. Release the left mouse button.

Response. The selected note cards are moved from their old location to the specified position.

If the note cards are being displayed by groups, a note card cannot be moved across a group boundary.

To save a new ordering:

Action. Choose Save as Special Order from the Listings menu.

Response. The ReadNotes program prompts for the name of the Special Order.

Action. Type in name.

Response. The program responds Saved. If the name already exists, prompts for confirmation.

Both SubGroups and Special Orders are titles for menu cards. To see existing SubGroups and Special Orders, the user can choose Expose. The following toggle-on-expose items are available:

Create. Prompts for a name. Creates Read Note window.
Delete. Prompts for a name.
Rename. Prompts for the old name and new.
Show. Prompts for a name. A Read Notes window is created and the note cards are listed.
**Controlling the appearance of the listings**

The user can control how the listings appear. The options include controlling how note cards are grouped, how the note cards are ordered, and how note cards are identified.

**To change the appearance of the note card listings:**

**Action.** In the Read Notes program, choose Format from the Listings menu.

**Response.** The EditFormat program will appear in a window.

**Action.** In the EditFormat program, click on the formats of interest.

**Response.** The formats will be highlighted.

**Action.** In the EditFormat program, choose Update Listings from the Top Menu.

**Response.** The listings in the Read Notes window will be changed.

**Controlling how note cards are grouped in the listings**

In the EditFormat program, the box entitled Group allows the user to control how the note cards are grouped. The fields include the following:

**Author.** Note Cards are grouped in the listing by author. The authors are arranged alphabetically.

**Classes.** Note Cards are grouped in the listing by classes. If a note card belongs to more than one class, it is displayed under both sub-headings. The classes are arranged alphabetically.

**Disable Groups.** The note cards should not be grouped.

etc.
Controlling how note cards are ordered in the listing

The box entitled Order allows the user to control the order in which the note cards are listed. The fields include:

- **Subject.** The note cards will be ordered alphabetically by subject.
- **Creation Time.** The note cards will be ordered by their creation time.
- **Modification Time.** The note cards will be ordered by their modification time.

etc.

Controlling how notes are identified

The box entitled Identification allows the user to control the way in which note cards are identified. The fields include:

- **Author.** The identification will include the author.
- **Subject.** The identification will include the subject. This is the default.
- **Page Number.** The identification will include a page number.
- **Creation Time.** The identification will include the creation time.

etc.

To make the format changes a permanent preference:

- **Action.** In the EditFormat window, choose Save as Preference from the Top Menu.
- **Response.** The specified format will be a permanent preference.
Modification time. Orders notes by the last time the notes were modified.

Subject. Orders notes alphabetically by subject.

An additional flag controls whether the order is Ascending/Descending:

Order Ascending/Descending. Controls whether the ordering is from a-z or z-a; 0-n or n-0.

Identification. These are diverse flags which control how note cards are identified in the listing, any subset of which may be chosen. An enable of a flag turns it on for the listing. A disable turns the flag off if it is on due to some grouping. The following are some possible flags:

Subject. The subject of the note is displayed in the listings.

Author. The author of the note card’s reference is displayed.

Title. The title of the note card’s reference is displayed.

Reference. The complete reference is displayed. This enables Author, Title, Journal, Page Number, Book, as appropriate.

Page Number. The page number of the note’s reference is displayed.

Creation Time. The creation time of the note is displayed.

Menu Options

The menu options differ slightly depending on whether EditFormat is invoked directly or from a program.

The first two options apply only if invoked from a program.

Update Listings. The new format is applied to the note card listings in the window.

Update & Quit. The new format is applied to the note card listings in the window and the program exits.

Reset to Preference. The note card listings in the window are updated to reflect the permanent preference. The option selections are highlighted appropriately.

The next two options appear from any invocation:

Save as Preference. The current format is placed in the data file for the note card listings as a permanent preference. Preferences only control the default for the initial display of the note card listing; they can be temporarily overridden by Updates.
Quit. The program terminates without saving or applying any changes. If there were changes, prompts for confirmation.

Files

See /cmu/eng/carm/warrant/doc/NoteCard.f for a specification of the file format for a note card file.

See Also

ReadNotes(1)

Bugs

Author