# DOCUMENT RESUME

ED 240 499 CS 007 428

| | |
|---|---|
| AUTHOR | Russ-Eft, Darlene F.; McLaughlin, Donald H. |
| TITLE | Ideas for Reading and Writing Courseware: Needs and Development Opportunities for Educational Computer Software--Reading, Writing, and Communication Skills. |
| INSTITUTION | American Institutes for Research in the Behavioral Sciences, Palo Alto, Calif. |
| SPONS AGENCY | Office of Educational Research and Improvement (ED), Washington, DC. |
| REPORT NO | AIR-26500-FR-83 |
| PUB DATE | Jun 83 |
| CONTRACT | 400-82-0021 |
| NOTE | 155p.; For the executive summary, see CS 007 427. Appendix C may be marginally legible. |
| PUB TYPE | Viewpoints (120) -- Information Analyses (070) -- Reports - Evaluative/Feasibility (142) |
| EDRS PRICE | MF01/PC07 Plus Postage. |
| DESCRIPTORS | Academic Standards; *Communication Skills; *Computer Assisted Instruction; *Computer Programs; Cooperation; Educational Innovation; Educational Objectives; Evaluation Criteria; Incentives; *Program Development; Reading Instruction; Reading Skills; *School Business Relationship; Speech Communication; Teacher Role; Writing Instruction; Writing Skills |

ABSTRACT

The product of a comprehensive project aimed at identifying reading, writing, and communication courseware needs, this report first discusses the benefits of computer assisted instruction and obstacles to its implementation in the communications area. It then describes procedures used to gather information and presents a detailed review of the study's results. The report suggests that progress in reading, writing, and communication courseware will be facilitated by clearer guidelines for the design of good courseware, increased incentives for software development, and stronger motivations for teachers to seek out and use the courseware. After discussing these three points, the report makes recommendations concerning the roles of the federal and state governments, the local community, and business and industry in supporting research and development of high quality courseware, in ensuring that teachers can use these materials effectively, and in developing adequate standards for evaluating computer software. Its appendixes include a partial list of study participants, a classification of courseware according to the language skills area, a school microware evaluation form, and sample evaluation questions. (MM)

AIR/26500/FR/83



AMERICAN INSTITUTES FOR RESEARCH
IN THE BEHAVIORAL SCIENCES
P.O. Box 1113, 1791 Arastradero Rd., Palo Alto, Ca. 94302 ● 415/493-3550



# IDEAS FOR READING AND WRITING COURSEWARE

## Needs and Development Opportunities for Educational Computer Software-- Reading, Writing, and Communication Skills*

Darlene F. Russ-Eft
Donald H. McLaughlin

American Institutes for Research
P.O. Box 1113
Palo Alto, California 94302

June 1983

* This work was supported by a contract from the Office of Educational Research and Improvement of the U.S. Department of Education, contract No. 400820021. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the Department of Education.





An Equal Opportunity Employer

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

LIST OF APPENDICES

## ACKNOWLEDGMENTS

We would like to express our appreciation to the many individuals who contributed to this effort. Ed Esty of the Office of Educational Research and Improvement in the U.S. Department of Education has assisted in numerous ways by providing information, encouragement, and critical reviews. The following members of our National Advisory Panel provided valuable suggestions on our procedures and reviews of our reports: Arthur Applebee, Mary Cron, Ed Esty, William Gattis, Bobby Goodson, Richard Hayes, Fred Holland, and Judith Langer.

The participants in our Idea Work Group meetings, who are named in Appendix A, provided us with a wealth of good ideas, only some of which appear in this report. We thank all of these participants for their contributions and particularly those who helped with the arrangements for the meetings: Beverly Bagan, Ronnie Banner, Paul Becht, Brian Doyle, Paula Filliman, Laura Latzke, Barbara Ueltschi, Robert Tinker, and Cynthia Wallet. Finally, we want to thank members of the AIR staff who contributed to this project: William Clemans, Robert Krug, and Lauress Wise for reviewing some of our reports, Annalee Elman for assisting in the literature review effort, Joseph Buckley for assisting in discussions with developers, Barbara Bessey and Jane Schubert for assisting in summarizing the tapes of the Idea Work Groups, Tom Bakke for preparing the listing of reading and writing courseware, and Susanne Horvath for making arrangements for the meetings with the Advisory Panel and with the Idea Work Groups and for generally administering the project. Although all these persons contributed to this effort, the authors assume full responsibility for the content of this report.

## INTRODUCTION

> The dawn of this new civilization is the single most explosive fact of our lifetimes. It is the central event -- the key to understanding the years immediately ahead. It is an event as profound as the First Wave of change unleashed ten thousand years ago by the invention of agriculture, or the earthshaking Second Wave of change touched off by the industrial revolution. We are the children of next transformation, the Third Wave. (Toffler, 1980, p. 25)

Alvin Toffler in his book, The Third Wave, wrote about the coming of the technological society. The convergence of computer technology and telecommunications provides the backdrop for an information revolution. However, a yawning chasm exists between the level of sophistication of the utilization of these technologies in the computing and communications fields and the level of use in education, particularly in instruction in skills such as reading and writing.

Reading, writing, communication, and computation are the fundamental skills which schools are expected to teach to American youth. But a cursory review of the many catalogs, advertisements, and lists of courseware (e.g., SoftSWAP; SWIFT Educational Software Directory; the Software Exchange; Computer Information Exchange; Electronic Learning, 1982) reveals that the availability of software in all areas except science and math lags far behind any reasonable expectations of those in schools that have acquired hardware. Furthermore, the vast majority of reading, writing, and communication courseware focuses on the simplest of tasks--spelling and vocabulary. This is true in spite of the national spotlight on declining reading scores and especially the decline in higher levels of reading and writing skills (NAEP, 1981).

There are three major causes of this state of affairs. First, instructional objectives for reading, writing, and communication involve complex combinations of basic cognitive skills. Innovations designed to meet traditional needs of teachers in this area, such as grading of compositions, are very difficult, if not impossible, to implement with existing hardware. Second, the computer medium is not as directly related to the subject matter in reading, writing, and communication as in math and

science. Language skills are characterized by many channels of information flow--including listening skills and free-form manuscript writing, resulting in a need for input-output other than screen-keyboard. Voice technology, essential for communications instruction, is still at an early stage of development. In some cases, large bodies of text must be accessed quickly, placing heavy demands on the text storage requirements. As a result of these problems, innovative ideas are needed in order to apply the computer's capabilities to meet the needs of teachers and students for improvements in reading and writing instruction. Finally, specialists in teaching reading and writing are not likely to have selected their careers because of personal skills that foster computer usage. Thus, the audience of reading, writing, and communications teachers is not yet prepared to compete with math and science teachers for computer resources.

Although several authors have discussed specific problems in the development of reading, writing, and communication courseware (Mason, 1980; Shostak, 1981; Thompson, 1980), there is a need for a systematic analysis of the key problems; this analysis can lead to an understanding of the nature of the barriers that limit the availability of high quality software for reading, writing, and communication skills. Equally important is the need for an examination of possible approaches that teachers, developers, pubishers, and educational administrators can employ to remove the barriers.

As a context for this examination, let us first describe an "ideal" classroom of the future and then identify factors that might affect the emergence of that ideal.

## Ideal Classroom of the Future

The function of the educational institutions in our society is to assist each individual to discover and develop his or her own unique potenial to achieve as high a quality of life as is possible based on that individual's evolving values in relation to the needs of society. To achieve this goal, it is essential to maintain students' eagerness to learn and develop throughout all phases of schooling. This enthusiasm to learn is characteristic of very young children; in many schools, however, the present programs gradually transform this eagerness into bored compliance to carry out meaningless tasks emanating from externally-conceived mandates.

Teachers are burdened with ever-increasing demands for curriculum innovations, and little time remains available for the much-desired individual student attention.

Excitement over the prospects of using computers in the classroom originate from the view that they will stimulate needed changes in education. Introduction of computers will:

(1) encourage more individualization of instruction;

(2) increase the capacity of teachers to manage and facilitate learning;

(3) give children greater control over their own learning; and

(4) permit greater coordination of learning environments.

These changes will help to ensure that education will enable each child to reach his or her goals and potential.

Computers can provide needed individualization of instruction. Through diagnostic sequences, a computer can identify an individual student's particular needs and skills. A student will be able to go to the computer and take a short test to determine the skills that are lacking. The computer could determine, for example, the reading level of a student and select the appropriate text to be read and the appropriate amount of practice to be given. One child may not need training on topic X, but another may need a double-dose. The computer has the capability of facilitating implementation of the appropriate instructional sequence for each of these children, permitting more individualized and self-paced instruction fitting individual needs. Learning and instruction will be success-oriented; the child will not be allowed to make large mistakes nor to stay in loops resulting in continuing failure. This will result in further changes in the instructional environment. Students will be working at their own pace, in some cases covering many years of instruction in a few months. Teachers will have more time to work with students having particular needs. This, in turn, will lead to a reduced emphasis on grade levels; students, who are grouped by age level for social development, will not expect to be working on the same exercises in reading, writing, and mathematics. In addition, students will have individual privacy during learning, a key to risk-taking and creativity.

Teacher skills will need to change in order to achieve this kind of environment. Teachers of the future will view themselves more as facilitators and managers of learning and less as instructors or lecturers. As facilitators, teachers will need to focus on developing the right atmosphere and environment for learning. Also, teachers will become providers of access to needed resources; rather than teaching and lecturing about basic information, they will be teaching children the strategies for finding information on their own.

With individualization of instruction, children will be able to take more responsibility for their own learning. They will make decisions about the sequence and pace to follow in particular lessons, with inputs from the teacher/manager. At the same time, this situation will facilitate the interaction of students with their peers. "The image is of a child interacting only with a computer. But, you cannot work with a computer that way. There have to be other people around to talk to." Thus, the computer can bring together students who will motivate each other, facilitating learning through self-motivation. This image has been elaborated by Leonard in Education and Ecstasy (1969).

With individualization of instruction, teachers as facilitators, and children as controllers, there will need to be increased coordination of the various learning environments, specifically the home and school. Learning stimulated at school will be able to be continued and expanded in the home environment. If this occurs, the classroom will involve parents as well as teachers in the education of children.

## Factors Facilitating the Emergence of the Ideal Classroom

To achieve this ideal classroom of the future, certain developments and advances in hardware will be required. Individualization of instruction, if provided by the computer, will require easy access to computer terminals by all students. A common complaint today is that there are never enough computers in the schools. But, in the future, lower hardware costs will ensure that each student has his or her own computer. These computers will be tied to large data bases, allowing the teacher to bring all kinds of resources into the classroom. A criterion for using computers will be the efficiency in accessing, manipulating, and presenting information. Voice

synthesis will be perfected so that students can learn sound-symbol correspondence using the computer, and voice recognition will be available to allow voice input. There will be more interfacing with other technologies, such as the videodisc and telecommunications. Furthermore, computers will be portable enough so that they can easily be taken to and from school, just as textbooks are today. Such a scenario will be facilitated by the development of "book-sized" computers with a flat, fold-down screen and keyboard and with high resolution graphics. Using cable or some form of telecommunications will facilitate the coordination of multiple learning environments, including the home. Schools are now limited to six hours, but computers will, in effect, allow students to take interactive lessons home with them, permitting the extension of individualized tutoring for as many hours as they wish.

Although hardware advances will be needed, advances in the development of software will be the critical ingredient for stimulating computer use in the schools. Some educators and developers believe that what is needed is a "Visicalc for education." Development of high-quality software, especially for teaching reading, writing, and communication skills, depends on the inputs and responses of two groups--developers and teachers. These groups, with counsel from the research community, must develop clear guidelines for software in reading and writing. In addition, developers must recognize the needs of the school as a separate market and must respond with creative ideas for using the computer. Finally, teachers must realize the potential for the use of computers to teach these basic skills and must respond with creative ways of implementing this kind of instruction. Communication and coordination among these two groups can be stimulated by support at the federal, state, and local levels.

As a step toward discovering the path to the ideal schools of the future, we carried out a study to examine needs and opportunities for new courseware, especially in the areas of reading, writing, and communication. The next section describes the procedures used in identifying the major barriers to software development and in developing potential solutions. The results section discusses three key barriers to educational software development and presents some potential solutions. A final section outlines possible roles of federal, state, and local agencies and of business and industry.

## PROCEDURES

The Office of Educational Research and Improvement, U.S. Department of Education, awarded a contract to the American Institutes for Research (AIR), in Palo Alto CA, to identify the needs for reading, writing, and communication software. Rather than carry out a broad evaluation of available courseware, which would duplicate other efforts, the AIR staff proposed to start with crucial problem areas and gather information necessary to explore potential solutions to them.

As background, the project prepared an issue-oriented literature review, including research on reading and writing, evaluations of the effects of computer-assisted instruction, evaluations of existing software materials, and compilations of clearinghouse materials on the use of computers for instruction. The review addressed needs for courseware guidelines in reading and writing, for support for developing high quality software, and for teachers' acceptance of computers in reading and writing instruction. (This review can be found in Russ-Eft, McLaughlin, and Elman, 1983.)

The results of the literature review were discussed and elaborated in a meeting with a National Advisory Panel of experts in reading, writing, software development, and software publishing. In addition, the Panel members reviewed and approved the proposed procedures for the conduct of the study. (A listing of the members of the National Advisory Panel can be found in Table 1.)

To clarify and explore solutions to the problems identified in the literature review, a series of 17 Idea Work Group meetings were held throughout the United States--in Bedford TX, Cambridge MA (2 meetings), Dallas TX, Gainesville FL, Glenview IL, Melrose Park IL, Osseo MN (2 meetings), Palo Alto CA (4 meetings), Richmond VA, Tallahassee FL, and Washington DC (2 meetings). The participants included 34 teachers, 25 school administrators, 14 teacher trainers, four state and three federal education officials, 22 researchers, 22 software developers, 13 publishers, and eight hardware manufacturers. Participants were identified through

Table 1

Members of the National Advisory Panel

Needs and Development Opportunities
for Educational Computer Software
in Reading, Writing, and Communication Skills

Arthur Applebee
Professor, School of Education
Stanford University
Palo Alto, CA

Mary Cron
Account Manager and Educational Specialist
Professional Computer Marketing Associates
Los Angeles, CA

Ed Esty
Project Officer
Office of Educational Research and Improvement
U.S. Department of Education
Washington, D.C.

William Gattis and Fred Holland
Education Division
Radio Shack
Fort Worth, TX

Bobby Goodson
Teacher, Hyde Junior High School
Cupertino, CA
(President of Computer-Using Educators)

Richard Hayes
Professor, Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA

Judith Langer
Professor, School of Education
University of California
Berkeley, CA

suggestions from members of AIR's Videodisc-Microcomputer Network project (McLaughlin, 1982), from members of the Advisory Panel, and from other leading experts in classroom computer use and in reading and writing. (A listing of these participants appears in Appendix A.)

The focus of the first round of all-day meetings (held in Bedford TX, Dallas TX, Gainesville FL, Glenview IL, Melrose Park IL, Palo Alto CA, and Tallahassee FL) was to identify the barriers to software development in reading and writing skills. The focus of the second round of meetings (held in Cambridge MA, Osseo MN, Palo Alto CA, Richmond VA, and Washington DC) was to explore solutions to specific barriers, using creative group problem-solving techniques. Tape recordings of the meetings were transcribed in summary form into over 200 pages of notes, and the ideas were categorized according to issue area.

In addition to conducting the Idea Work Groups, AIR contacted over 200 developers and reviewed 12 major software catalogs and the 1981, 1982, and 1983 issues of eight computer magazines and two computer news-weeklies to identify relevant software. The software packages were categorized by reading or writing skill area (using a recent basal reading series). This categorized listing (presented in Appendix B) helped to identify skill areas needing more attention.

Finally, to supplement the coverage of software developers in the Idea Work Groups, the staff held private discussions with 40 other software developers. The purpose of these discussions was to assess general perceptions about the educational software market and to identify additional problem areas.

## RESULTS

The information-gathering efforts on this project resulted in a wide range of courseware ideas in reading, writing, and communication. These ideas can be presented in three main categories. Thus, progress in the areas of reading, writing, and communication courseware will be facilitated by (1) clearer guidelines for the design of good courseware, (2) increasing the incentives for the development of reading, writing, and communications courseware, and (3) increasing the incentives for teachers to seek out and use the courseware. This report will focus on potential solutions to problems in these three areas.

(1) Clearer guidelines--how must traditional reading, writing, and communication objectives be translated in order to convert them into courseware that makes use of the computer's potential?

(2) Developers' incentives--how is knowledge of the school market to be disseminated and copying of software to be dealt with so that developers can be assured that they will recover their investment?

(3) Teachers' incentives--how can teacher training be implemented to overcome resistance, how can courseware be designed to incorporate new teachers' roles and to be more friendly to teachers of reading, writing, and communication skills, and how can evaluation standards incorporating teachers' needs be developed?

Each of these problems represents a critical barrier to realization of the potential for reading, writing, and communication courseware. Potential developers of such software find it difficult to translate higher-level objectives, such as "find the main idea in a paragraph," "determine the author's purpose," and "write a poem" into instructional sequences that make use of the computer's strengths. Clearer guidelines can remove some of the trial and error costs for development and result in higher quality software. Because the development of reading, writing, and communication courseware is more difficult, and therefore more expensive than other courseware, it is the most sensitive to factors that reduce the expected return; at present, the problem of "copying" is perceived as a major threat to large volume sales (Wollman, 1982). Alternative solutions to problems in

the marke-place, such as software piracy, can lead to an increase in developers' and publishers' incentives to produce high quality reading, writing, and communications software. Publishers are also wary of investment in this field because many teachers, especially outside the science and math areas, are either hesitant or resistant to the new technology. Methods for increasing teacher acceptance, such as suggested by Fraser, Wells, and Burkhill (1982), are needed.

In the next three sections, we discuss the three key problems outlined above. This discussion evaluates some solutions that have been proposed and indicates what questions need to be addressed by developers, educators, and publishers.

## 1. Clearer Guidelines

Overall, when reading, writing, and communication courseware have been compared with mathematics courseware, the former have proven less effective (e.g., Ragosta, Holland, & Jamison, 1982). This is not surprising given the greater difficulty of developing courseware focusing on reading, writing, and communication skills. Clearer guidelines for all communications courseware are needed. Such guidelines may, perhaps, build on general courseware considerations (Caldwell, 1980; Cohen, 1982; Gagne, Wagner, & Rojas, 1981), but they must also focus directly on the specific content area.

Several technical problems make reading and writing courseware especially difficult to design. These problems include: (1) text storage requirements, (2) need for input-output other than screen-keyboard, (3) need for a sophisticated transformational grammar, and (4) need for better understanding of reading and writing objectives. In addition, Malone (1981) has presented data suggesting that there may be specific problems in motivating children to play "word" games, but he acknowledged that this may be a function of the games that he selected.

Each of the above points must be considered when preparing recommendations for software developers; indeed, all these problems appear in the literature and were discussed by the Idea Work Group participants. However, it is the fourth problem--need for better understanding of reading and writing objectives--that has been given least attention by courseware developers and that provides the most difficult barrier. As Roberts (1982) has pointed out, the current thinking on instructional objectives is confused.

Objectives that have been taught as comprising the ability to read or write, especially at the higher levels, are defined without reference to the underlying skills that must be acquired to exhibit the ability. According to one Idea Work Group participant, "There is a wealth of research on reading skills, for example, but there is a gap between the research and the implications for teaching reading. What is needed is an identification of the critical reading skills and of methods for teaching them." This is especially true for the higher-level reading and writing skills. Well

designed courseware must be based on an understanding of the component information processing skills involved in the traditional reading and writing objectives. Only then can instructional sequences be designed which assess and build on component skills, gradually extending the hierarchical structure (Gagne, 1965; Hartman, 1981) that constitutes these abilities.

Designing a curriculum in which higher level skills are based upon an understanding of the component skills is important for traditional instruction as well as computer-based instruction. However, this approach is especially important for computer-based instruction, because, with its branching capabilities, the computer is able to deal more patiently with minute increments in skills, quickly reinforcing small successes while jumping equally quickly to appropriate elaborations or correction procedures when necessary. If courseware in the higher skill areas of reading and writing is to take advantage of the computer's unique capabilities, it must identify and address the skill needs that underlie the instructional objectives.

In focusing on skill definitions, we plan to build on Glaser's (1976) and Smith's (1978) outline for the design of instruction. Glaser identified four components of instructional design:

(1) assessing competent performance;

(2) describing initial states;

(3) identifying conditions that foster the acquisition of competence; and

(4) assessing the effect of instruction.

These four components also appear in the task analysis process developed by Smith (1978). This process is summarized in Figure 1.

Central to any instructional development is the analysis of competent performance leading to a description of what is to be learned. This first component is similar to Smith's (1978) question: What are the target behaviors? For this first component of instructional design, Glaser emphasizes the importance of the information processing approach to the analysis of skills (e.g., Simon & Chase, 1973), in which skills are seen to be combinations of (1) knowledge structures (e.g., vocabulary) and (2) process structures or internal programs for solving problems (in this case,

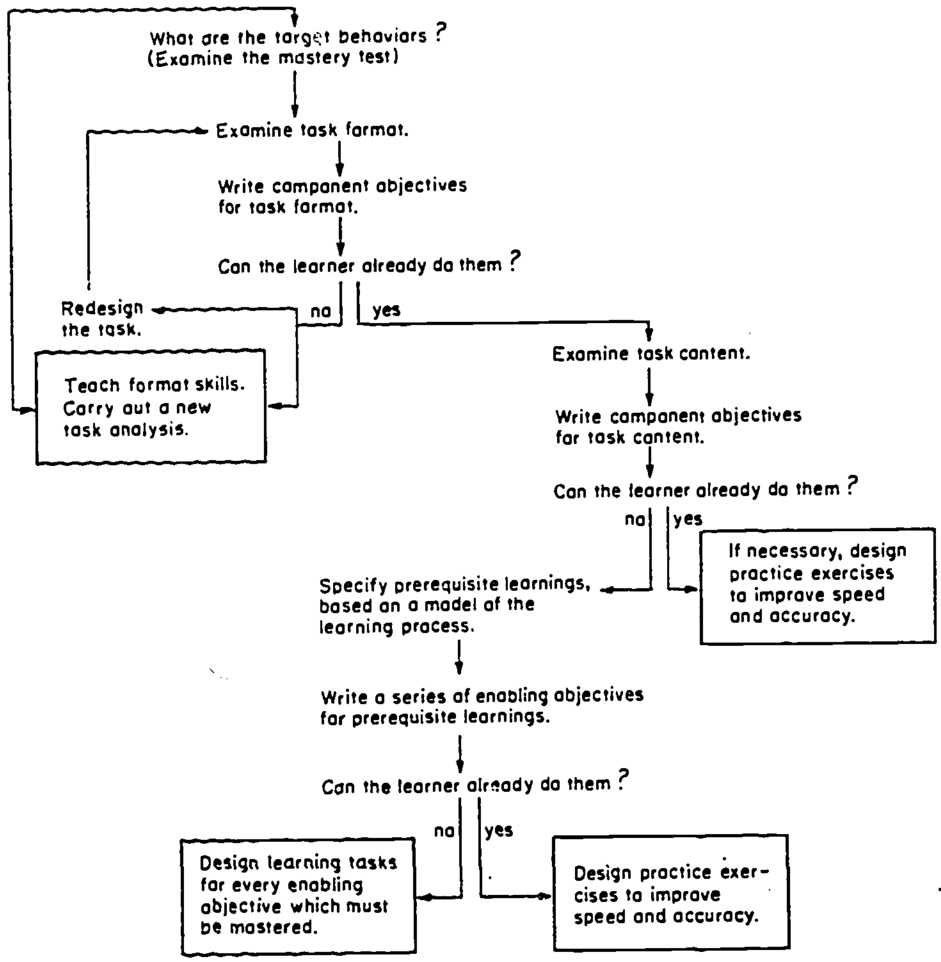


Figure 1. Summary of the task analysis process presented by Smith (1978).

the problems of reading and writing). These concepts are widely accepted by cognitive psychologists as central to the understanding of higher mental processes. Norman (1980) has pointed out the crucial role of knowledge structures in the acquisition of new knowledge; and Schank and Abelson (1977) have demonstrated the importance of process structures or scripts in human performance.

The second component in Glaser's model, assessment of initial states, is similar to the specification of prerequisite learnings identified in Smith's model. Glaser provided a set of questions that must be addressed:

> "What are the details of that a child knows and does not know at particular points in his or her learning? What are the details of the skills that he or she is developing? What needs to be improved? What strengths can be capitalized on? What do various developmental levels and various cultural backgrounds mean for what should be taught and how it should be taught?" (p.13)

Although analysis of reading and writing into a hierarchy of skills has been incomplete, researchers have shown the value of skill analysis for efficient instructional design (Holtzman, 1975; Frederiksen, 1982, 1983). Holzman (1975), for example, studied the recognition of sequential patterns in letter series completion problems. He used the results to develop an effective method for teaching the detection of relations and the discovery of periodicity. This provides evidence that instruction based on tutoring and practice in the important incremental skills can prove superior to the traditional syncretic design of instruction. It is necessary, however, to identify those skills.

The third component of Glaser's model focuses on conditions that can be implemented to foster the acquisition of competence. An analysis is needed to determine the different ways in which individuals process and acquire increasingly complex skills and knowledge. Three important factors of each learning situation must be examined: (1) how to evoke and build on existing skills and knowledge structures (e.g., Greeno, 1976; Norman, 1980), (2) how to manipulate conditions and events surrounding the individual, such as reinforcement; and (3) how to develop, consolidate and generalize learning-how-to-learn skills.

The fourth component, assessment of the effects of instructional implementation, can be directly applied to the question in Smith's

taxonomy: Can the learner already do the target behaviors? Glaser emphasizes that this measurement should focus on clear descriptions of competent performance rather than merely apply traditional norm-referenced testing. One example of this approach, which courseware developers might emulate, is Carroll and Freedle's (1972) analysis of the processes involved in the comprehension of written language.

The scope of our study of the design of reading and writing software focused on the first three components of Glaser's model. The fourth component, assessment of the effect of instruction, must await the development of the instructional designs. (Of course, assessments of the effects should take place throughout developmental efforts.) The central problem specific to reading and writing courseware is the lack of analysis of instructional objectives in terms of cognitive processes and structures.

Recognizing this critical need, we have analyzed the domains of reading and writing in three stages:

1. identification of the functional domain generating a particular instructional objective;

2. identification of specific information processing skills required for the functions; and

3. inference of recommendations for courseware development.

The courseware development process frequently starts with a specific idea for a series of excercises or activities. In order to go beyond an interesting game or a repetitive sequence of similar exercises, the developer should analyze the skill domain according to the stages listed above. The developer should begin by locating the idea within the functional domain of reading and writing. This provides the basis for communication to teachers about the courseware and for extending the courseware concept beyond the initial germ of an idea. Next, in refining the exercises and activities, the courseware developer should identify the different ways that students can interact with the program and the information processing skills that will be developed and consolidated through these interactions. This analysis can then lead to implications for the courseware development. (This process represents an ideal, requiring familiarity with the skill domain; and it is our intent to provide

some needed background for this process, based on our literature review and the contributions of the Idea Work Group participants.)

The three stages, proceeding from functional domain to information processing skills to courseware recommendations, form the structure of our presentation, separately for reading and for writing. While the outcome falls short of a cookbook for developing high quality courseware, we hope that the combined inputs of our Idea Work Group participants will furnish guidelines and ideas to facilitate the developer's task.

Reading

Reading can be defined as having a conversation with someone who is not present. It means that the reader can "talk" with Hemingway, for example, and can "listen" to what he has to say. This emphasizes the communications aspects of reading. There is a symbol system between two senses -- visual and auditory.

According to Gibson and Levin (1976), reading consists of "extracting information from text." By text, these authors refer not only to printed words and sentences but also to combinations involving text and pictures, diagrams, graphs, and so forth. Reading consists of a variety of skills; some are bottom-up skills and some are conceptual skills. Poor readers differ from good readers on all these skills. Poor readers are not as good or as fast at decoding; they do not use context as well; they are not as good at monitoring their own comprehension processes; and they do not have a good knowledge of text structures to use to organize comprehension. Thus, reading requires the interaction of numerous cognitive processes and problem solving strategies.

The communication aspects of reading are discussed in Moffett and Wagner (1976). Literacy skills, defined as reading and writing, can only be considered to be "basic skills" in a relative sense. Although these skills are taught early in school and provide the foundation for later book learning, their acquisition follows the development of oral speech, as well as the acquisition of extensive nonverbal experience. Thus, two levels of coding (or translating information) precede the so-called "basic" literacy skills:

(1) Conceptualization (non-oral) -- coding of experience into thought;

(2) Verbalization (oral) -- coding of thought into speech; and

(3) Literacy (written) -- coding of speech into print.

This means that the literacy skills of reading and writing depend upon prior codings of experience into thought and of thought into speech.

Across all levels, both encoding and decoding skills are required. Thus, for example, speaking relies heavily upon the encoding skills of

composing or formulating implicit words or thought into explicit words, and listening involves the decoding skills of comprehending or interpreting explicit words into implicit words or thoughts. Writing focuses on encoding skills in transcribing words, while reading focuses on decoding skills in word recognition. Figure 2 provides a simplistic picture of this four-way communication structure. In addition, it must be recognized that many educators and researchers emphasize that reading, for example, requires both decoding and encoding skills.

Most basal reading programs at least acknowledge this four-way nature of the communication process, while focusing on reading as the ultimate goal. Let us take the Addison-Wesley Reading Program as an example. The general skill areas include:

(1) Reading readiness;

(2) Listening skills;

(3) Speaking skills;

(4) Reading skills; and

(5) Writing skills.

Reading readiness includes nonverbal encoding and decoding skills such as visual discrimination, visual memory, directionality, auditory discrimination, auditory memory, conceptual development, motor skill development, social awareness, and creativity. Listening and speaking skills require the development of auditory discrimination, aural comprehension, oral reading, oral expression, and dramatizing. Finally, the reading and writing skills involve phonetic analysis, recognizing memory words, structural analysis, vocabulary, comprehension, study skills, written expression, capitalization, punctuation, usage, structure, spelling, and handwriting. (See Freedle, 1979, for a review of the current theories and research on communication or discourse processing.) Future developments of educational software in the area of communication skills must emphasize the complementarity and interaction of these skills.

Langer (1981) discusses three important points concerning reading that must be considered when attempting to implement reading instruction via the computer:

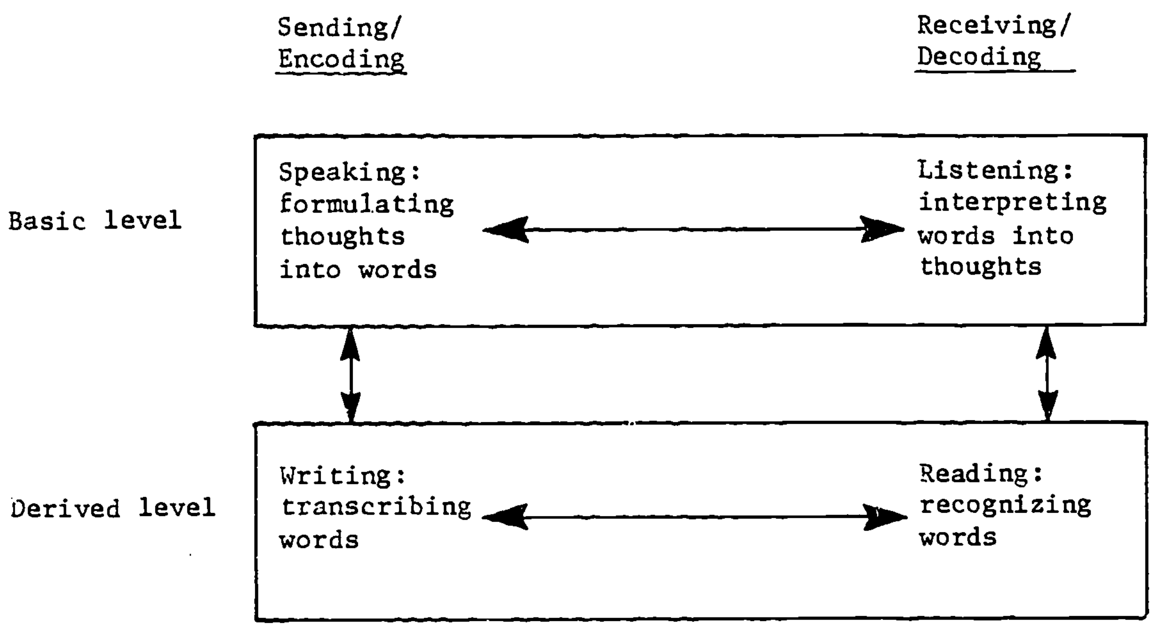


Figure 2. The two directions and two levels of verbal communication (adapted from Moffett and Wagner, 1976, p. 13)

(1) Reading is an interactive process involving the construction of meaning by the reader in interaction with the text composed by the author.

(2) Reading instruction should be process-based. Improvements in reading comprehension will occur if readers can be helped to know what text-based clues to use in a particular situation and to know when and how to use them.

(3) Reading instruction should be learner-based. Such instruction will assist readers to develop metacognitive control over their own skills and strategies.

Reading, as an interactive process, involves the coordination of the reader's prior knowledge, the author's purpose as conveyed in the text, and the reader's use of the actual text (Langer, 1981; Rumelhart, 1975). Rumelhart (1980, 1981) discusses how data structures for representing generic concepts stored in memory, or schemata, influence an individual reader's interpretation of the text. Indeed, reliance on text cues diminishes as readers become more proficient; meaning is constructed through concept processing (Bobrow & Norman, 1975). Presumably, the skill of using textual cues can be recovered from the reader's repertoire when difficult passages are encountered. Thus, a proficient reader utilizes both "top-down" (or goal-oriented) and "bottom-up" (or surface text controlled) processing. Unfortunately, many of the instructional programs provide practice only in bottom-up tasks. So, software development is needed to correct this imbalance by focusing especially on top-down tasks, such as question generating.

Process-based instruction recognizes that the reader accomplishes a reading task by applying appropriate processes to use cues presented in the text. Each text possesses certain word-feature, syntactic, and semantic cues that the reader can use to develop hypotheses about the meaning of text (Goodman, 1973; Adams, 1980). Failure to recognize these cues and to use them appropriately can lead to difficulties in reading and comprehending texts. Langer (1981) suggests that software developers might consider the development of a decision-making structure that could be used for the on-line processing of a variety of texts being read for a variety of purposes.

Metacognition refers to deliberate and conscious control over one's own cognitive processes. This involves self-reflection and awareness of what is known, what needs to be known in a particular situation, and what must be done if something goes wrong. In the situation of normal, rapid reading, the reader may be considered to be in the "automatic pilot" state where component processes may be subconscious. Therefore, emphasis should be placed on the "debugging state", where some of the processing activities become conscious (Woods, 1980). Brown (1980) listed the following as metacognitive processes used in reading:

(1) clarifying the purposes of reading, that is, understanding the task demands, both explicit and implicit;

(2) identifying the aspects of a message that are important;

(3) allocating attention so that concentration can be focused on the major content area rather than trivia;

(4) monitoring ongoing activities to determine whether comprehension is occurring;

(5) engaging in review and self-interrogation to determine whether goals are being achieved;

(6) taking corrective action when failures in comprehension are detected; and

(7) recovering from disruptions and distractions.

She also suggested that there are many more deliberate, planful activities, such as setting goals for reading, that render reading an efficient information-gathering activity.

Instructional activities that help students learn how to acquire information from text (Brown, Campione & Day, 1980) are needed. Computer-based activities can be designed to help the reader become aware of the demands of the reading task, to judge what needs to be known, and to identify strategies to be used when problems arise.

The basic message of recent reading research is to broaden the concepts of reading instruction and to identify skills that differentiate good from mediocre readers, going beyond the simple decoding tasks. Some comments from Idea Work Group participants help to define this need:

It is easy to develop curriculum materials, including software, that teach and provide drill-and-practice in recognizing letters, letters, groupings, sight words, and word groupings. Much more difficult is the area of reading comprehension and concept formation.

We need reading comprehension software, because we have the greatest difficulty teaching that.

Comprehension skills are critical. It doesn't really matter whether you learned using the phonics method or some other method. What is important is whether you understand what you are reading.

Educators must realize that all the small skills are not necessarily critical to performing the final act of reading and understanding.

In order to translate this message into software guidelines, we must identify particular links between reading tasks and learning sequences that improve skills needed for those tasks. The link involves analysis of three levels: (1) functional domains, (2) information processing skills, (3) courseware implications.

Functional domains. There are "traditional" instructional objectives that most teachers feel are important when considering teaching reading. However, some argue that these traditional listings of reading objectives may have little to do with successful teaching. Several reading programs train children in specific skills, but still some students fail to learn to read. What may be needed is an identification of the critical reading skills they lack and of methods for teaching them. This can be done by looking at the information requirements needed for achieving that particular reading skill. Thus, to be effective, software should be designed to address specific skills that are of critical importance to the reading process.

Although an information-processing approach may facilitate the design of high-quality software, the developer must also convince teachers of the usefulness of the resulting product. In order to communicate with teachers and to indicate the usefulness of a piece of software, any new skill domain should be compared with those tasks (and instructional objectives) that characterize traditional taxonomies. Therefore, we will start with the traditional objectives for teaching reading.

One of the most widely used taxonomies of reading comprehension skills is that of Barrett (1976).

1.0. Literal Recognition or Recall.

1.1. Recognition or Recall of Details.

1.2. Recognition or Recall of Main Ideas.

1.3. Recognition or Recall of Sequence.

1.4. Recognition or Recall of Comparisons.

1.5. Recognition or Recall of Cause and Effect Relationships.

1.6. Recognition or Recall of Character Traits.

2.0. Inference.

2.1. Inferring Supporting Details.

2.2. Inferring the Main Idea.

2.3. Inferring Sequence.

2.4. Inferring Comparisons.

2.5. Inferring Cause and Effect Relationships.

2.6. Inferring Character Traits.

2.7. Predicting Outcomes.

2.8. Inferring about Figurative Language.

3.0. Evaluation.

3.1. Judgments of Reality or Fantasy.

3.2. Judgments of Fact or Opinion.

3.3. Judgments of Adequacy or Validity.

3.4. Judgments of Appropriateness.

3.5. Judgments of Worth, Desirability, or Acceptability.

4.0. Appreciation.

4.1. Emotional Response to Plot or Theme.

4.2. Identification with Characters and Incidents.

4.3. Reactions to the Author's Use of Language.

4.4. Imagery.

This taxonomy distinguishes at the top level among tasks requiring memory, inference, evaluation, and appreciation. Pearson and Johnson (1978) provide a comparable listing of skills, but they distinguish among concept level

comprehension, propositional level comprehension, and making judgments about the written word. Another example can be found in Heilman (1967), which includes a list of 10 abilities that are prerequisites for critical reading:

(1) The ability to recognize the meanings of words;

(2) The ability to select the one appropriate meaning of a word which may have many meanings;

(3) The ability to deal with figurative language, not insisting on literal meanings when the author does not intend literal interpretation;

(4) The ability to determine the author's main ideas;

(5) The ability to paraphrase, or restate what the author has written;

(6) The ability to see the relationship between one part and another and of all parts to the whole;

(7) The ability to adjust to the author's organizational pattern;

(8) The ability to determine the author's purpose, his intent in writing, his point of view, his biases, or whom the author is addressing and with what goal in mind;

(9) The ability to draw inferences which are not specifically stated in the data; and

(10) The ability to recognize literary devices such as humor, satire, or irony, and to detect mood or tone (Heilman, 1967, p. 426).

Each of these abilities is functionally stated, in terms of a type of problem that is encountered in critical reading, rather than as a set of cognitive processes. Effective courseware for critical reading, on the other hand, must build on analyses of these abilities.

Similar lists of abilities can be found in materials of publishers of elementary curricula and tests. For example, the reading comprehension items in the California Tests of Basic Skills (California Testing Bureau, McGraw-Hill, 1981) are divided into seven categories:

(1) The student will identify a picture that shows the meaning of a written sentence. (sentences with different structures).

(2) The student will extract details from a passage to answer who, what, where, or when questions.

(3) The student will analyze the feelings, traits, or motives of characters in a passage.

(4) The student will identify the author's purpose or viewpoint, the main idea or tone and mood expressed in a passage.

(5) The student will draw conclusions from or recognize cause-and-effect relationships in a passage.

(6) The student will differentiate between reality and fantasy, between fact and opinion, or between forms of writing (news article, interview or speech, legend or history, biography or autobiography).

(7) The student will recognize techniques of persuasive writing or figurative writing (testimonial, bandwagon, personification, simile, hyperbole, metaphor, imagery, irony).

A different approach to the definition of the functional domain appears in the work of Jenkins and Pany (1981). They began by identifying three causes of reading comprehension failure: (1) the lack of appropriate abstract knowledge structures or schemata needed for making sense out of text, (2) difficulties related to the structure of the passage, such as complex or unfamiliar vocabulary, sentence syntax, and text organization, or to the requirements of the passage, such as the requirement for special thinking skills like syllogistic reasoning, and (3) inattention or superficial or misdirected attention (e.g., concern with correct word calling rather than abstracting meaning). They then identified the instructional variables in reading comprehension associated with the three problems. Finally, they reviewed the research related to each of these variables with regard to the effects on reading comprehension. Figure 3 summarizes the results of their review. It provides some indication of areas that should be incorporated in software development efforts. For example, a developer can concentrate on creating a program that forces the student reader to "chunk" written information. We should caution that a developer should not use this listing as an indication of approaches to be eliminated from software. In most cases, the conclusions were that the research results were uncertain because of a mixture of positive and negative findings. This may be due to the design of the research or to the

fact that the selected variable affects performance for certain individuals under certain conditions. Nevertheless, the listing provides some ideas of dimensions to be considered when developing software to teach reading comprehension.

Summaries of reading skill hierarchies can be found in Rosenshine (1980) and in Jenkins and Pany (1980). These authors conclude that the major taxonomies and hierarchies, found in basal reading programs and in factor-analytic studies on reading, appear to be more similar than different. Clear, substantive differences among basal reading programs do, however, appear in the number of exercises and questions devoted to specific skills. Unfortunately, few comparative evaluations of these programs exist, so teachers and software developers lack the needed data for identifying critical components. In a sense, each of these "abilities" represents a need and an opportunity for courseware development, but without a cognitive process analysis any courseware that is developed will consist merely of unorganized drill and practice (though possibly graphically quite enjoyable).

In order to provide a clearer idea of the type of analysis that is needed, we have selected one ability as an example, "finding the main idea in a paragraph." This ability, posited as a single skill in most scope and sequence charts, encompasses a variety of bottom-up and top-down skills. In its simplest form, this ability can be defined as finding which idea is the central, or main, idea of each paragraph. This definition focuses attention on the structure of a paragraph and the relationship of sentences within a paragraph.

Alternatively, this ability of finding the main idea in a paragraph might be considered as merely an easily presentable subset of a much larger domain of extended "transformational grammar" in which linguistic units larger than sentences must be processed. In the latter case, we are not interested so much in the structure of individual paragraphs as in the skill of summarizing the major ideas in any multi-sentence passage. Of these two conceptions of the ability of finding the main idea of a paragraph, the former is "less extensive" in that it is particularly directed at paragraph structure, while the latter is "more extensive," representing transformational grammar skills in general.

| Instructional Variables related to Reading Comprehension | Results of Research using the Instructional Variables |
|---|---|
| 1. Instructional Variables Addressing Schema-Related Problems | |
| Advance organizers | do not appear to represent a high success intervention for enhancing comprehension |
| Titles and paragraph heads | show some positive effects of thematic paragraph heads |
| Pictures | are appropriate when reading for new information, but no evidence to indicate that pictures make students better comprehenders |
| Background knowledge | does not represent a set of skills to be taught as part of reading instruction |
| 2. Instructional Variables Addressing Linguistic/Reasoning Problems | |
| Rapid decoding | indicates that slow decoding, with numerous hesitations and repetitions, breaks up the continuity of thought affecting comprehension |
| Vocabulary | can affect comprehension at the sentence level, but does not seem to improve overall reading comprehension |
| Cloze hypothesis/ test procedure | appears to affect reading comprehension |
| Organizational strategies | appears that activities assisting student to "chunk" information during reading can enhance comprehension |
| Specific subskill instruction | is typical of school instruction; lack of evidence as to whether mastery of specific subskills affects overall comprehension |
| Auding-reading relations | imply that efforts to improve understanding of spoken messages will affect reading comprehension |

Figure 3. Results of research on instructional variables related to reading comprehension as reported by Jenkins and Pany (1981).

3. Instructional Variables Addressing Attention-Related Problems

| | |
|---|---|
| Questions and purpose setting | have uncertain effects on reading comprehension |
| Incentives | alter performance, but have uncertain effects on reading comprehension |
| Reading strategies | have conflicting findings, deserve more attention from researchers |

Figure 3. (continued) Results of research on instructional variables related to reading comprehension as reported by Jenkins and Pany (1981).

Information processing skills. As the research literature indicates, the traditional taxonomies and hierarchies for reading skills are not an ideal basis for developing software to teach reading. An analysis of the information processing skills is also needed. Frederiksen (1982, 1983) provides an example of the kind of analysis that is needed. His componential theory of reading identifies a set of functionally defined information-processing components, which in interaction with one another enable reading with comprehension. This theory begins with the identification of three major processing levels: (1) a set of word-analysis processes, (2) discourse-analysis processes generated by the text; and (3) an ability to combine information from word and discourse processes, called integrative processes. Figure 4 shows how the integrative processing can lead to greater efficiency in both the word and discourse processing.

Using this model, Frederiksen developed hypotheses concerning correlations among the components. These hypotheses were tested using a series of experimental tasks, such as anagram identification, word recognition, reading phrases in paragraph context, and so forth. Based on the analyses of the results, the final model included the following independent components of performance:

I. Letter recognition

II. Perception of multi-letter units

III. Decoding

IV. Word-recognition efficiency

V. Speed in applying context

VI. Extrapolation of a discourse representation

VII. Influence of topicality of referent

VIII. Semantic integration of antecedents

This model can be compared to Thurstone's multi-factor theory of intelligence (French, 1951; Thurstone, 1938; Thurstone & Thurstone, 1941) and Guilford's model of the structure of intelligence (1967). Primary mental abilities identified by Thurstone that relate to Frederiksen's model include: perceptual speed, verbal comprehension, word fluency, associative

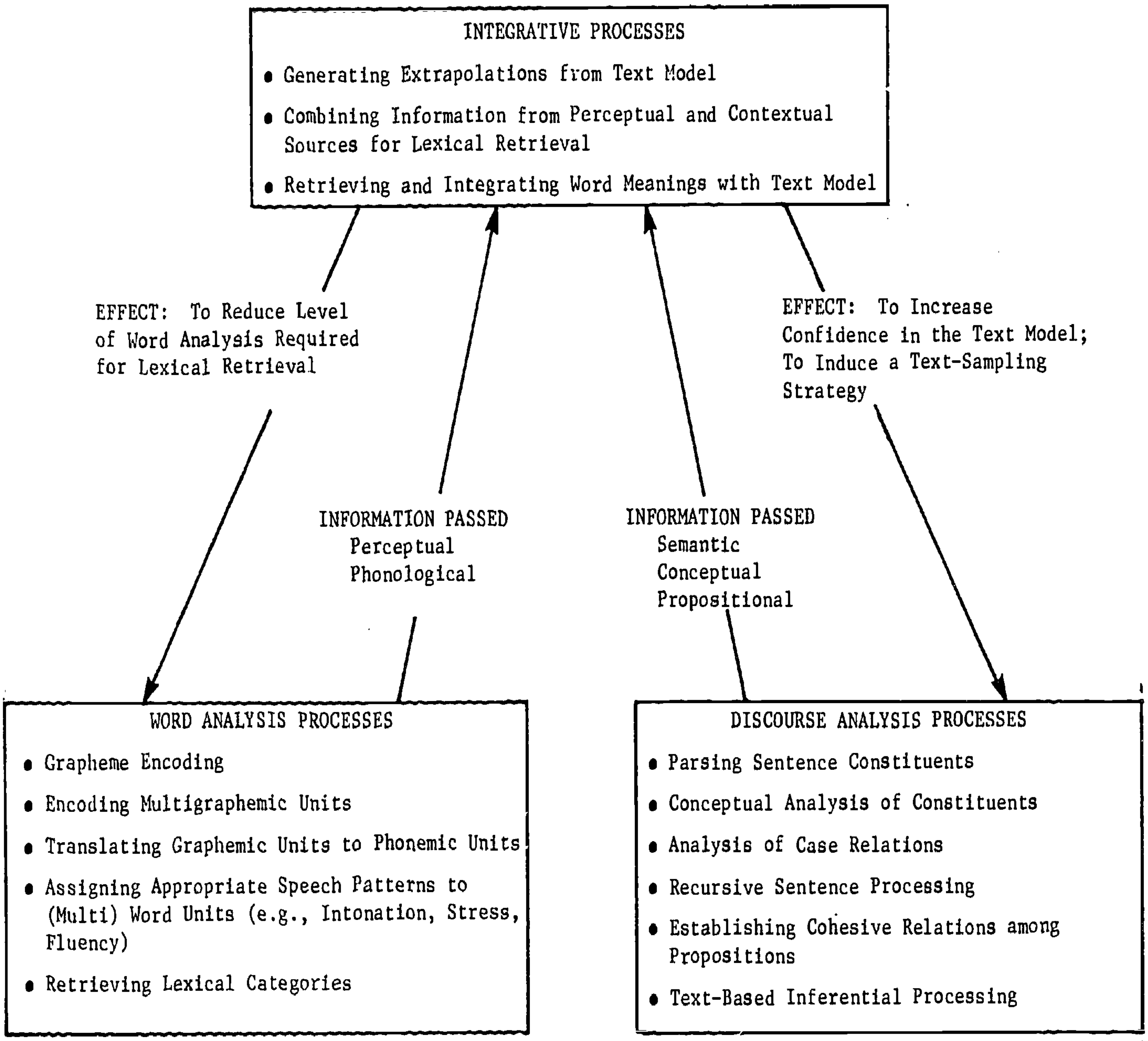


Figure 4. Categories of reading processes and the nature of their interactions (taken from Frederiksen, 1982)

memory, and induction or general reasoning. Guilford, in comparison, proposed a model classifying intellectual traits into three dimensions:

| contents | operations | products |
| --- | --- | --- |
| figural | cognition | units |
| symbolic | memory | classes |
| semantic | convergent production | relations |
| behavioral | divergent production | transformations |
| | evaluation | systems |
| | | implications |

Although one might claim that reading only involves semantic contents, we can see that there are figural contents in terms of letter recognition and perception of multi-letter units, symbolic contents in grammatical rules, and behavioral contents in understanding the author's point of view. Many of the reading skills involve convergent production, since there is a need to close the communication gap that exists between the reader and the author. However, extrapolation of context, involving the activation of semantically related items in memory, can be viewed as similar to divergent production.

Theoretical analyses, such as those proposed by Frederiksen, Guilford, and Thurstone, may be productive for identifying the information-processing skills underlying the ability to read. Having identified these skills, one could then develop programs to teach mastery of these skills.

Another approach to analyzing the information processing requirements embedded in reading tasks is based on the work of John Seely Brown and his associates on non-reading tasks (Brown & VanLehn, 1980; Burton, 1981; Collins, Brown, & Larkin, 1977; VanLehn & Brown, 1980). These researchers developed methods for diagnosing systematic student errors by fusing computer science tools with modeling techniques from cognitive science. These diagnostic systems have been used to analyze students' work to construct an extensive catalogue of precisely defined systematic errors or bugs for a well-defined task: place-value subtraction. In the "Buggy model" that was created, a student's errors are seen as symptoms of a "bug" or a discrete modification to the correct skills. For example, we can consider the following two errors made by a student:

$$\begin{array}{r} 500 \\ -65 \\ \hline 565 \end{array} \qquad\qquad \begin{array}{r} 302 \\ -243 \\ \hline 149 \end{array}$$

Both errors are accounted for by the bug named 0-n=n. In this case, the student has a modification to the subtraction procedure dictating that when the top digit in a column is 0, the answer written for that column is the bottom digit. Such bugs in the subtraction procedure may be the result of deleting part of the correct procedure, of adding incorrect sub-procedures, or of replacing correct subprocedures by incorrect ones. Thus, rather than merely determining whether or not a student has mastered the skill of place-value subtraction, these researchers have identified specific subskills that the student has not mastered. A similar kind of analysis could be undertaken for components of the reading process. For example, there are failures on numerous spelling rules ( e.g., i before e except after c) and grammatical rules ( e.g., use a singular verb--runs--with a singular pronoun--she).

As a part of any analysis of information processing requirements for reading comprehension, it might be useful to focus on three stages of reading: before the text is read; while the text is being read; and after the eyes have left the page. Each of these stages requires a different set of information processing skills. Instructional decisions before reading focus on vocabulary and conceptual background needed for a particular text. Instructional decisions during reading emphasize the anticipation of structural, organizational, and rhetorical elements and sensitivity to cohesive aspects of the text. Instructional decisions after reading focus on post-questions, feedback, and textual recall. Further description of these stages can be found in Langer (1982).

Let us now return to our example of how to analyze the skill of identifying the main ideas in paragraphs. In the less extensive model, directed at paragraph structure, the emphasis is on identifying the relations among a set of sentences that are in the same "paragraph." For example, it is important in deciding among competing candidates for main idea to identify which ideas are elaborations, expansions, examples, analogies, contrasts, or qualifiers of others.

In the more extensive model, the emphasis is on generating trial main ideas based on the content of the prose and then checking them against the passage and modifying them to improve the fit to the passage, while avoiding excessive complexity. The subtlety of statements of ideas, rather than relations among sentences, is at the core of the more extensive model.

According to Guilford's model, the reader is using divergent production to generate the idea and evaluation to check and modify them. Using Frederiksen's model, the critical skills involve a combination of context-utilization processes: (1) generative use of context for activating semantically related items in memory and (2) use of contextual cues to increase the speed of lexical processing.

In all the models, there are certain assumed prerequisites, such as word analysis processes (e.g., grapheme encoding and retrieval of lexical categories) and lower level discourse analysis processes (e.g., parsing sentence constituents). A student who had not mastered the extraction of meaning from sentences or who failed to know the meaning of key words in a passage could not be expected to be very successful in identifying the main idea in a paragraph. Given these prerequisites, according to the less extensive model of the skill, the cognitive processes involved in the skill include:

- identifying relations among words in different sentences;
- from related words, identifying the relations between sentences containing those words; and
- comparing the structure of relations among sentences to particular paragraph main idea models to generate the main idea.

The more extensive model of the skill involves a much broader range of processes, those that make up "inference" from prose. The focus is not so much on solving a set of relational constraints as on producing ideas. It is not as important that the student be able to compare and identify which of a set of ideas is the main one as that he or she be able to produce a statement of a key idea that lay more or less hidden in the passage. Generally, the skills involved are those that enable a person to understand the meaning of a text passage (or the prose-parsing transformations required for critical reading).

Courseware implications. The types of exercises one should include in courseware on a specific reading skill clearly depend on the analysis of that skill. Much of the software that has been developed has focused on drill and practice of specific skills, such as vocabulary drill. As stated by one teacher, "It is easy to develop curriculum materials, including

software, that teaches and provides drill-and-practice in recognizing letters, letter groupings, sight words, and word groupings. Much more difficult is the area of reading comprehension and concept formation." Appendix B provides a listing of some of the available software categorized by reading skill area, as taken from the Addison-Wesley Scope and Sequence chart (1982).

Rather than follow the pattern established by previous attempts to develop software to teach reading skills, it would be more profitable, from an educational standpoint, to consider exercise types that truly expand and challenge the student. Langer (1981), following a presentation of the major aspects of reading, suggests several computer-based activities that could improve reading comprehension. We endorse them as worthy guidelines.

(1) Increase student's awareness of text features a) to meet text-based needs and b) to override text-based problems.

(2) Model a metacognitive strategy in the presentation of a particular reading activity. Gradually turn the decisionmaking, and later the question-generation, over to the student.

(3) Provide concept and language awareness activities prior to reading to help students think about what they already know about a specific topic. Have them anticipate what they will read in the text.

(4) Provide activities which require decisions about ideas in the text that may or may not make sense or are not necessarily consistent with one another.

(5) Help students decide how thoroughly they must learn the textbook material based on the purpose for reading.

(6) Develop activities which ask adjunct questions to teach self-questioning before, during, and after reading.

(7) Vary audience, author, or voice to help students become aware of these shifts.

(8) Vary text clues at word, sentence, and text level and have students develop sensitivity to their varying levels of usefulness.

(9) Have students judge what they think will be easy/difficult for their classmates to understand. Why? What would make it different?

(10) Present writing-in-progress and have students determine whether the author is exercising strategies that they know (Langer, 1981, pp. 9-10).

In a later publication, Langer (1982) uses an information processing analysis focusing on the processes before, during, and after the reading act to identify relevant instructional activities. This listing, presented in Figure 5, provides some worthwhile suggestions for software developers.

Frederiksen (1983) provides another example of how an information processing analysis of reading skills can lead to the development of implications for software. He used his componential theory of reading to develop three computer games to help train students in three skill components: (1) perception of multiletter units appearing within words (the Speed game), (2) efficient phonological decoding of orthographic information in words (the Racer game), and use of context frames in accessing and integrating meanings of words read in context (the Ski-Jump game). Results showed that trainees were able to reach levels of performance in the trained skills that equalled or exceeded those of high ability readers. In addition, there was evidence for transfer of acquired skills to other functionally related reading components.

In some cases, it is useful to explore alternative definitions of the skill. An example of this approach appears in Block (1979). It involves an application of cognitive theory to the production of computer-assisted instruction in spelling. Theory predicts that spellings of words are produced in one of three ways depending upon the amount and kind of information about a specific word stored in memory. (See Simon & Simon, 1973).

(1) Direct recall is used for spelling highly overlearned and familiar words.

(2) A generate-and-test process is used for less well learned words whose spellings are only partially recallable.

(3) A direct phonemic spelling process is used for words that have not been seen before and for which visual recognition information is not present.

Stage of Reading

| Before | During | After |
|---|---|---|
| current related prior knowledge (concepts) | predicting what comes next | organization of recall (hierarchical) |
| text-related knowledge (format, text structure etc.) | integrating (constructive aspects) | organization of story (recall of structure as well as recall of details) |
| specific vocabulary knowledge | using self-questions | post-questions (textually-and-scriptually-based) |
| understanding the purpose for reading | knowing when and how to get additional information | long and short term recall of metaknowledge of task |
| familiarity with form, style, genre | keeping purpose for reading in mind | knowing when being uncertain is OK |
| knowing what one knows and needs to know | using flexible strategies at work, sentence and text level | judging if information gained is sufficient -- based on purpose |
| | monitoring inconsistencies | reacting to author's point |
| | becoming aware of author's goals | |

Figure 5. Information processing skills and activities required before, during and after reading (taken from Langer, 1982, p. 14).

Two instructional programs were developed to provide children with either a phoneme generator (SPLPAT for Spelling Patterns) or the generate-and-test procedure (TRYSPL for Try Spellings). Testing of these programs suggested that (a) children's performance on the TRYSPL program was consistent with the generate-and-test theory, (b) generating and testing is a useful and credible instructional approach that does not inhibit the learning of the correct spelling; and (c) children's generation performance improves with training on spelling patterns (using SPLPAT).

A variety of suggestions for different types of exercises were contributed by the teachers, developers, and researchers in our Idea Work Groups. The suggestions were discussed and elaborated to identify key components and concepts. The suggestions can be categorized into two major groups: (1) activities that focus on reading skill development and (2) tools that provide support for the reading process. Perhaps the most innovative of these contributions were the following:

Skill Development Activities

(1) Present simultaneous speech and text, as a kind of Rosetta Stone. The student can then make the translation from words to text, using the two forms of input.

(2) Use color to represent sounds within a phonically-based system. As children learn the sounds, the computer could gradually remove the color distinctions.

(3) Present written materials auditorily. Then students would get not only practice in the correct form of the verb, but they would also hear it being used.

(4) Provide word banks allowing the child to say a word and the computer to give the correct spelling and meaning. Alternatively, the computer could show the word and then pronounce it, helping students learn how to pronounce multisyllabic words.

(5) Present speed reading exercises that teach the skill of reading from left to right. This could be coupled with software that monitors eye-tracking skills.

(6) Have the computer highlight the main idea, but only if the student takes too long to respond.

(7) After highlighting the main idea, move that main idea to different parts of the paragraph--to the end, to the middle, to the beginning--and ask for the child's interpretation of how that changes the meaning.

(8) Create an adventure game in which the child receives instruction notes on how to get past the next obstacle. These could get more and more complicated with a sequencing of skills.

(9) Display only a piece of the text, forcing students to make inferences about the text.

Tools Supporting Reading

(1) Provide reader with the flexibility to scan the text, to move forward and backward quickly and easily. This would enable the student to learn the skills associated with scanning a textbook to identify the major headings and main ideas of a chapter.

(2) Present the background experiences needed for understanding the setting of a story. This might be in the form of a computer simulation.

(3) Present computer simulations of stories. These would get the students into actually experiencing the stories, such as the whaling experience of Moby Dick.

(4) Introduce students to a specific author, using an interactive videodisc. This would be a "meet-the-author" program that would motivate students to read books by that author.

(5) Present interactive stories, allowing the reader to manipulate characters and determine the outcome.

(6) Present interactive texts (or data bases or dictionaries) from which the reader can request supplementary information, such as a vocabulary item, a graphic display relevant to the text, or the main idea of the paragraph.

(7) Have the computer take as input the name of a favorite book and then list the names of other books by the same author.

In addition to suggestions about software to teach students particular skills or to provide support for the reading process, there is a need for software to test and monitor reading-related skills. The following are some suggestions from the Idea Work Groups:

(1) A program to test whether young children have problems in reading from the left to the right and in crossing the midline.

(2) A program to test for problems in directionality. Many children are unable to see a difference between "was" and "saw."

(3) A program to find the reading level of individual children quickly and easily.

(4) A program to monitor eye movements. This could be used to identify back-tracking and other indicators of reading difficulty.

At this point, let us return to our example of teaching how to find the main idea of a paragraph as an indication of how to analyze a reading task in order to develop courseware implications. In our less extensive model (focused on paragraph structure), the progression of exercises could begin with finding similar words in different settings using increasing complexity of the "similarity." Next, one could provide exercises that involve parsing sentence-pairs, based on overlapping words, with increasing complexity of the sentences. Finally, one could present a series of exercises that relate sentence-relations to location of the main idea. For example, one rule that might be learned is the following: if a paragraph contains three or more sentences and all but one of the sentences are special cases of the fourth, then the fourth sentence states the main idea.

In the more extensive model of "finding the main idea," a different set of variations would be more appropriate. First, one could begin with exercises consisting of increasingly complex ideas that are to be extracted from a single sentence. Next, one could present exercises in which the relation between a stated sentence and the target idea increases in complexity. Finally, one could provide exercises in which there is increasing interference from competing candidates for inference.

A key to the development of high quality courseware is the interaction of developers with teachers and educational researchers throughout the design and development phases. Thus, courseware developers should start by obtaining an outline of an instructional sequence from teachers and researchers. Then, they should map out a sequence of student interactions that the computer could support. Through repeated interplay of

instructional design expertise and knowledge of the computer's capabilities, a unique computer-supported lesson sequence would be designed. Taking as an example the "more extensive model" of skill of finding the main idea, the following presents an initial plan for courseware development.

(1) Start by identifying a large set of ideas that are typically expected to be extracted from prose passages, and determine the transformations required for extracting these ideas. This will give ideas for exercise structure and content and help to avoid stereotyped passages.

(2) Select a game format in which precise identification of ideas leads to clear positive feedback and in which typical errors (e.g., overgeneralizations) lead to informative losses of points.

(3) Insofar as possible, allow for production of ideas, not merely recognition and selection among presented ideas. For this, it may be useful to build in a process for recognizing ideas stated by the student.

(4) As an alternative to requiring overt production (e.g., typing of identified ideas), set up subsequent choice problems that require having previously extracted an idea from prose.

(5) One relevant type of exercise is to present an idea and a separate passage. Then ask for the (minimal) change to the passage so that the idea can be inferred from it.

(6) An extension of direct inference that may be useful in games is probabilistic inference. In this case, the student would be asked to indicate the likelihoods of ideas being true, given prose passages.

(7) Do not focus on algorithmic comparison of competing main ideas in a paragraph, because that is not the skill that is intended for this instructional objective. (That is the focus for the other version of this skill.)

This kind of analysis can be applied to the entire range of reading and writing objectives, and we believe that it will help developers and publishers to create high quality reading software.

## Writing

The opportunities for innovative use of computers to improve the teaching of writing are tremendous. At a time when the processes underlying expert writing are just beginning to yield to the inquiries of researchers, the very nature of "writing" is also undergoing significant change as a result of the introduction of new technology. The skills that have formed the core of the teaching of writing in schools, spelling, punctuation, grammar, and penmanship, are becoming automated; and new skills of composition, such as goal-setting, audience-modeling, heuristic search, sophisticated editing, and general information processing, promise to form the core of a new curriculum for writing. The variety of approaches for bridging the gap to a new curriculum appears unlimited, and computer-based instruction is particularly appropriate for the introduction of teachers to these approaches.

There is in 1983 a dramatic separation between the ways in which writing is taught in many traditional classrooms and the ways in which it is conceived by many educational researchers and innovative teachers. At times, it would hardly seem that the two perspectives refer to the same topic area. Most schools continue to focus on the acquisition of the rules that differentiate correct from incorrect writing. Educational researchers, on the other hand, have recently focused on the interplay of a variety of cognitive processes in writing tasks, seeing writing as a problem-solving task requiring planning, heuristic search, judgment, and so forth.

In meetings with teachers, researchers, and software developers, this conflict was repeatedly brought up, although the point was frequently made that many teachers, especially the more progressive ones, not only accept but extend the higher level approach to writing. The following excerpts from these meetings indicate the range of innovations in writing instruction that educators are looking for.

> Computers have been effective in teaching capitalization and mechanics of [illegible]. There is less evidence of effective use in teaching higher level composing skills. The problem of selling courseware for composing skills is that these skills are not those tested for in standard tests, so their value is not evident to teachers, administrators, or parents. There is a need to change the tests.

One problem for developers is that the skills being taught by the courseware may go beyond those currently being taught in the schools. This may mean that the developer has to educate the publishers and teachers to realize that children can and need to learn these skills.

Getting children to write something that is logical is difficult to teach. Children have problems in writing their thoughts in a logical fashion, with a beginning, a middle, and an end. A theory of the story-generation process is needed. Logical written communication should be seen as the end product of the composition process that includes drafting, rewriting, and proofreading.

Students need to learn how to generate "reader-oriented" prose, that clearly communicates the intended ideas.

The problem of choosing a structure for the document is attacked by analyzing the context of the communication problem: who is the audience, what do they need to know, what is my purpose in writing.

Let us assume that the ultimate goal is a logical story using all the rules of grammar and punctuation. Perhaps the computer could be used to teach logic, maybe not through writing, but through different game techniques that employ logic and sequencing. In these cases students would have to learn these specific problem solving skills and then would draw them together for the ultimate goal of writing a logical story.

It is difficult to teach children how to write poetry. We're great at teaching them how to write stories, how to write about their experiences, how to write story summaries.

Another difficult skill to teach is how to write opinions and editorials. Opinion may be difficult because, unlike the other forms of writing, you are not modeling. It is difficult for children to state their reasons for an opinion.

Spelling and grammar may be emphasized too much. In many schools these are emphasized to the exclusion of communication principles. Grammar and spelling should be seen as subservient to communication.

Another need is to reduce, or avoid increasing, "writing anxiety." One way to do this is to help students to apply the specific problem solving skills they have developed in other areas to the writing process. They can see the process as a communication problem in which they search for solutions in familiar ways. This has been tried at the college level but might also apply to younger students. These "processes" are called "design principles" or "the scientific method" by professionals in different fields.

There are a variety of reasons why writing has not been better taught. Unlike the areas of reading and mathematics, there are few "basal" methods for teaching the "composing" process, and teachers have not generally been trained to teach writing skills. Thus, teachers are confronted with a need for assistance in developing their students' compositional skills.

Edwards (1982) suggests six causes for the lack of good instruction in writing. First, standards for writing instruction are lacking in teacher training curricula. Second, possibly as a result of poor instruction, teachers themselves often view writing as a tedious chore and convey this model to children. Third, teachers, administrators, and curriculum designers fail to assign a clear place for writing instruction in the curriculum. These background factors are reinforced by, fourth, the lack of time for carrying out the essential feedback activities for writing instruction. Fifth, the "back-to-basics" movement has focused on superficial details of the writing process. (According to Emig (1978), "Much of the current talk about the basics of writing is not only confused but, even more ironic, frivolous.") Vocabulary, spelling, grammar, and mechanics have become all-important. Finally, sixth, few requirements are being placed on students to write. Very little time is spent writing passages of more than a single paragraph.

Software can help fill the need for improved writing instruction in two quite different ways, (1) by focusing on rule-learning or skill objectives, and (2) by providing support for the composition process. The latter has evoked a great deal of interest in the past year or two, in the form of word-processing and related software packages. There are also interesting developments in the area of skill identification, however. Recent research has focused on the organizational and planning skills in composition, and on the complexity of the review/revising phase of writing.

In the 1970s, and especially in the period since 1975, there has been a substantial amount of research on writing, and there have been a number of reviews, which need not be repeated here (Larson, 1978, 1979; Humes, 1982). In addition to the journals _College Composition and Communication_ and _Research in the Teaching of English_, three collections of papers are especially useful in understanding the emerging views on writing, or "composition." These are _Research on Composing: Points of Departure_ (Cooper & Odell, 1978), _Cognitive Processes in Writing_ (Gregg & Steinberg,

1980), and Writing: Policies, Problems, and Possibilities (Cronnell & Michael, 1982). Potential courseware authors should find these collections to be invaluable sources of design ideas and of means for evaluating outcomes.

Functional domains. Writing covers a broad range of tasks and skills, and any attempt to develop a curriculum for the subject must start with an identification of parts of that range that can be focused on individually. There are a variety of methods for categorizing "writing," and particular educators may be comfortable with only one or another. In order to communicate to potential users, a courseware developer must be able to place an instructional module's objectives in a familiar framework.

Three methods of categorization of tasks within the general domain of writing are familiar to many educators:

(1) developmental sequences, proceeding from concrete component rules to complex, abstract concepts;

(2) different purposes for which writing is undertaken, such as report-writing versus poetry; and

(3) different phases of the task of producing a written product, including pre-writing, writing, and revising.

A curriculum developer will increase the size of his or her audience by considering all three frameworks. A particular exercise may possess an identifiable place in one categorization but cut across levels of another.

There have been many attempts to partition writing skills into a set of developmental categories. The categories selected for earliest instruction have traditionally focused on very concrete rules that can be easily checked by student and teacher alike, such as spelling, punctuation, and handwriting. Categories that are taught later tend to be more abstract, to involve longer bodies of text, and to have less obvious criteria for measurement. Recent attempts to teach writing at very early ages, however, suggest that children who are not ready to learn concrete rules of writing may already at age 5 be able to construct meaningful stories, so the developmental sequencing of these categories is open to question. Perhaps a closer connection between oral communication skill learning and learning to write should be explored. Nevertheless, the developmental categories are

extremely familiar to most teachers and are perceived as reasonable for describing the level of children's achievement in the area of writing.

Various publishers have prepared curricula in writing based on the developmental categories, and they have described the components of their programs in terms of "scope and sequence" charts. For example, the Addison-Wesley Scope and Sequence Chart for Writing contains six major headings of objectives:

Handwriting;

Spelling and encoding;

Structure;

Usage,

including verbs, modifiers, pronouns, and common confusers;

Capitalization and punctuation; and

Written expression,

including sentence writing, paragraph writing, report writing, and proofreading.

The concept of a hierarchical ordering of components of writing has appeared in various forms. For example, Koch and Brazil (1978) proposed a priority ordering for diagnosing problems with student writing, in which the most global concerns received highest priority:

I. Global concerns:

A. Rhetorical concerns
1. Unity: logical development and flow of thought
2. Focus: staying on the topic without wandering
3. Coherence: "sticking together" of major parts of writing, use of transitions
4. Pointedness: responding pertinently to the writing topic
5. Sufficiency: saying enough to get the job done
6. Value: the quality of thought

B. Rhetorical structure
1. Introduction
2. Body
3. Conclusion

II. Other structural concerns:

A. Paragraph structure
   1. Development: "backing up" of generalizations by using details, examples, illustrations, comparisons, and so on
   2. Coherence: one sentence "fitting" with or leading to another; using transition words or phrases

B. Sentence style and structure
   1. Sentence fragments
   2. Run-on sentences
   3. Awkward sentences
   4. Wordy sentences
   5. Choppy sentences
   6. Illogical sentences
   7. Lack of sentence variety

III. Writing errors:

A. Subject-verb agreement
B. Verb tense consistency
C. Pronoun reference and case
D. Internal punctuation
E. Capitalization (for beginning of sentences only)

IV. Mechanics:

A. Spelling
B. Punctuation mechanics: use of apostrophes, hyphens, capitals, abbreviations, and numbers

V. Usage questions (e.g., hanged-hung, who-whom, can-may)

VI. Dialect features.

In this scheme, they emphasize that teachers should focus on the global aspects of the writing and battle their tendencies to mark the easily identifiable errors of types III through VI.

Although the elements of these categories are very familiar, and course-ware developers should consider them in planning their modules, they should not be considered as the sole structure underlying a curriculum for writing. There is evidence, in fact, that isolated learning of rules at one level does not automatically transfer to writing tasks encountered at other levels.

Much of what is taught beyond the simple rules of spelling and grammar is, in fact, questionable. As Applebee has pointed out (1981), the advice to begin each paragraph with a topic sentence does not appear to have been followed by great writers; nor is outlining clearly effective--it may

actually inhibit good writing. The important message in this for potential courseware developers is that the promise of CAI in the area of writing will be diluted if what is created merely mimics teaching practices already present in the classroom. There is an opportunity not only to introduce a new medium, with all the advantages that have been stated elsewhere, but also to facilitate the introduction of new ideas in the teaching of writing into the curriculum.

The second major framework concerns the purpose of the writing. According to Britton (1978), for example, the function of writing can be transactional (writing "to get something done"), poetic (writing to create a work of art, in which "every part is appropriate to each other and to the whole design"), and expressive (informal and casual writing for a variety of purposes). The skills needed for transactional writing are quite different from those needed for poetic writing or for expressive writing. Among transactional writing tasks, Britton identified two major subcategories: conative and informative. Conative writing can further be divided into regulative (compliance assured) and persuasive; and informative writing can be divided into seven levels of abstraction, described in Britton, et al. (1975).

The implication of this framework for courseware development is that separate lessons must be designed for teaching of the skills essential for different forms of writing. The teaching of some basic skills will, of course, facilitate all writing. However, teaching conative writing will not involve the same steps as teaching poetic writing; so we should not expect the one form of writing to be learned automatically when the other is learned.

A phase framework for describing writing tasks categorizes each writing task into three (or four) phases: pre-writing, writing, and post-writing, with post-writing possibly divided into reviewing and revising. In the pre-writing phase the writer plans and organizes; in the writing phase the text is generated according to production rules; and in the post-writing phase the writer evaluates the text and improves it.

Bridwell (1980) has focused on the revision process and its state of development among twelfth graders, and Nold (1979) has developed a framework

for revising. The "phase" framework has been elaborated in the information processing approach to analyzing writing. Hayes and Flower (1980) examined protocols of writers and found out that the "pre-writing" processes actually occur throughout a writing task, as do writing and review. The processing framework divides composition into "planning," "translating" or "transcribing," and "reviewing," with various subcategories. Furthermore, the skills involved in accessing other information might be considered separately from general planning skills (Gould, 1980).

In several meetings conducted in the project, suggestions were made concerning the phases of the composition process, especially the "planning" and "reviewing" phases. The following are some of the comments.

(Planning Phase)

> Currently when writing is taught, the teacher assigns a topic. But this is not really fair to the students. In many cases, students cannot even understand their own writing. They need something to guide them through the first draft. Perhaps a computer program could be written to provide them with a model as a stimulus, several suggestions for topics, and then guiding questions or prompts. A guiding question might be: What is the main character's name in your story?
>
> One approach teaches students about "plans" that appear in texts, such as the problem-solution plan, the compare-and-contrast plan, the cause-effect plan, the description plan, or the story plan. Students write according to these different plans using the same topic. As a result, they exhibit significant increases in memory and comprehension and ability to handle inference questions.
>
> There are some prototype programs that do try to give children prompts to help structure their stories. The computer gives the student the frame and the student must fill it in. So it is a collaborative writing effort with the computer and the child.

(Reviewing Phase)

> The computer can help in the drafting and revising process. It is important to convey the idea that, when you write, you don't just settle for your first draft. Rather, there is always room for improvement. Many times we don't give students the opportunity to draft, to read, and to revise.

> Perhaps you could have children begin to learn about the drafting and proofreading process by having them do it manually. That is, you would begin with having them cut-and-paste a document. Later, they could use the computer to do the same thing.
>
> A problem with writing instruction is that kids want to just sit down and do something quickly. Reviewing, editing, and revising require the development of patience and longer attention spans. Perhaps playing the "editor" role will help develop these attitudes.

Curriculum developers can be expected to generate particular lessons and exercises based on a variety of personal experiences and idiosyncratic methods. The purpose of considering existing "functional domains" is to facilitate communication with those who will evaluate, select, and use the curriculum. A common complaint about existing software is the failure of developers to bridge the gap from the computer program to classroom applications. Inclusion of examples of classroom uses (that have been tried out), couched in terms familiar to educators, will be likely to improve the product's acceptance.

Information processing skills. In addition to placing a piece of courseware in a functional domain framework, the developer should analyze the writing tasks, and courseware exercises, to determine that the skills being developed and consolidated do, in fact, match the objectives for which the courseware is designed. The information processing skills analysis is quite different from the functional domain analysis, in that the former focuses on large categories while the latter focuses on detail. For example, within the "planning" phase a variety of skills are required, and many of the same skills recur in the "reviewing" phase.

Writing, in fact, consists of a large body of skills that spans much of human cognition. Both production processes and information gathering processes play roles in writing, and a major problem for students learning to write is the need to "simultaneously" pay attention to a variety of constraints and hold a variety of ideas in memory. On the one hand, it is essential that many of the skills become automatic, so that, for example, a writer need not worry about uses of semi-colons while planning the organization of a persuasive argument. On the other hand, it essential that writers learn to organize their problem-solving tasks so as to minimize

memory overload, avoiding concern about grammar while generating ideas for a composition.

There is an underlying dichotomy of skills that affects both the types of courseware needed and the ways in which courseware is described; the two categories of skills involved in writing are:

(1) Discrete rules to be learned; and

(2) Continuous skill domains that are mastered gradually.

A comprehensive writing curriculum must cover both types of skills, and the pedagogic method must be designed to match the type of skill being learned.

In the case of rule learning, most of the rules can be explained to most students explicitly, and students can memorize the verbal expressions of the rules. The objective of instruction, however, must be to go through a series of stages with each rule: first recognition of instances of the rule, with recognition of the associated errors, then production of text in accordance with the rule upon request, then spontaneous recall and use of the rule in general writing contexts, and finally, automaticity of application of the rule. At the point of mastery of a rule, students carry out writing tasks applying the rule without imposing any incremental load on conscious processing. Bridwell (1980) has noted that the automaticity of orthography, spelling, and basic sentence construction is important to reduce the cognitive load on the writer and make room for the creative heuristics.

Some educators argue that for many students the rules are already automatic at the beginning of formal writing instruction, and no effort should be expended in making students conscious of these rules. Certainly, prior to any extensive instructional component, a diagnostic test to determine prior acquisition of a rule will improve the efficiency of instruction.

The rules can be ordered in a manner very similar to the developmental sequences. One categorization of rules is the following:

(1) Rules for forming letters of the alphabet;

(This includes printing, cursive writing, and "keyboarding".)

(2) Rules for spelling;

(3) Rules for forming grammatical sentences;

(4) Rules for use of punctuation;

(5) Rules for standard forms;

(This includes form letters, scripts, poetry, lab reports, and book reports, among others.)

(6) Rules for structure, such as good paragraph structure, good structure for arguments, and good structure for stories;

(7) Rules for identifying content that will communicate the desired ideas to the intended audience;

(8) Rules for information gathering, such as the use of a dictionary, an encyclopedia, or a library; and

(9) Rules for reviewing, such as playing the role of the reader.

Many of the rules that have been proposed by experts on writing remain controversial, and exceptions abound, so courseware intended to teach rules of writing should be carefully designed to convey the extent and limits of each rule.

Contrasted with discrete rules are the several sub-domains of gradually mastered information-processing skills. The teaching of these skills differs from the teaching of rules in that a verbal description of the skill, unlike the rule, is never sufficient for imparting the skill. Repeated practice is necessary to expand and solidify the skill.

The following categories of skills appear to be the most important for the domain of writing, based on the perspectives of teachers and educational researchers who participated in this project.

(1) Vocabulary, that is, the potential to produce the word that is needed to express one's thought, given the constraints of the writing context;

(2) Heuristic Memory Search, that is, the ability to retrieve from one's memory content appropriate to the current goal of the writing;

(3) Problem-Solving Management, that is, the ability to keep track of a variety of goals and subgoals simultaneously in order to move from one goal to another smoothly;

(4) Role-Playing, that is, the ability to create an image of the context within which readers process the written text and of the factors that affect the impact of the text on them;

(5) Spontaneous Comparison, that is, the automatic production of "ideals" against which trial expressions are compared in order to generate new subgoals for planning, writng, and revising;

(6) Organization, that is, the ability to imagine reorderings, categories, and transformations of the text, in order to explore alternative expressions;

(7) Memory, that is, the ability to keep track of previously generated text and ideas yet to be expressed, in order to create continuity in text; and

(8) Text Production Speed, that is, the ability to print, write, or type the text that expresses one's thoughts quickly.

This list of skill categories, like the prior list of rules, is a synthesis of ideas expressed by teachers, researchers, and publishers, in the course of the project's Idea Work Group meetings. In the case of the skills, numerous suggestions were made for teaching them in more general contexts, not just in the context of writing. Problem-Solving Management, for example, can be practiced in a variety of settings, and a major step to assist developing writers is to help them to generalize previously learned problem solving methods to the writing situation.

In one of the meetings in the project an attempt was made to obtain a comprehensive list of the writing skills that are needed. The following illustrates the result.

Sustaining a line of thought;

Diversifying to increase interest;

Understanding one's "voice" or role;

Audience analysis;

Observing conventions of structure (e.g., a story has a beginning, middle, and end; a paragraph has a topic sentence);

Surface conventions, like spelling and punctuation, but spelling may not be required with such rigor as the amount of content which writers must know increases and as spelling checkers become universally available;

Searching for ideas of what to write;

Information retrieval skills;

A sufficient set of facts on which to base thinking, both in one's memory and in very quickly accessible form, such as through a word-processing retrieval system;

Creative heuristics;

Critical re-reading of what one has written; and

Creating meaning for the reader.

The content of a new, comprehensive curriculum for writing might be built on skills such as these, although researchers are concerned that many of the hypotheses about the writing process are untested. Tryouts of these concepts in courseware modules are needed in order to indicate the appropriate directions for further development.

Finally, before turning to specific courseware implications, it is worth focusing specifically on several alternative approaches for teaching rules involved in the planning process, because this is an area in which there is the greatest need for skill improvement. According to Hayes and Flower (1980), planning can be divided into subtasks of generating content, organizing content, and setting goals. They found that good writers are particularly facile and flexible in setting and resetting goals, in the middle of other writing "episodes." The goals may relate to the overall purpose of the writing task or to details of the composition.

Young (1978) describes four methods for fostering "invention" in the composing task. These methods can be seen as ways of generating a rich set of goals for a writing task. According to the classical method, one first identifies the topic, then decides on a thesis or main argument, and finally uses ethos (appeal to morality), pathos (appeal to emotions), and logos (appeal to reason) to generate the content of a composition. A second approach involves an analogy with drama; Burke's (1955) pentad involves identifying an Act, a Scene, an Agent, Agency, and Purpose. This approach is a more general analysis of human behavior and motives, which happens also

to apply to writing. It serves a purpose not far different from the journalist's Who? What? Where? When? How? and Why?

The third of the four approaches is Rohman's (1965) pre-writing instruction, which is to teach heuristics for creative problem-solving, suggested by Gordon (1961), Koestler (1964), and Bruner (1965). Analogy is the primary tool for analysis and generation of ideas in this approach. Gordon, in his "Synectics" system for creative problem-solving, proposes four methods for stimulating creative thought, Personal Analogy, Direct Analogy, Fantasy Analogy, and Symbolic Analogy.

Finally, there is Pike's "tagmemic" approach, which consists of "a series of heuristic procedures for analyzing and formulating problems, for exploring problematic data in search of solutions, and for testing solutions." It is presented as an expanded linguistic theory that includes, among other things, techniques for inducing psychological change in the audience (Young, Becker, & Pike, 1970).

Software implications. There have been other recent summaries of the categories of writing courseware, with particular examples included (e.g., Collins, 1982; Bradley, 1982; Bridwell, Nancarrow, and Ross, in press), and individuals considering the development of writing software for the school market should be aware of the contents of these. In this section, we summarize the categories of courseware for teaching writng and focus on suggestions derived from the Idea Work Group meetings on how to produce high quality courseware.

Software to assist in writing instruction can be divided into two major categories:

(1) software aimed directly at particular rule learning or skill development, and

(2) software aimed at motivating, supporting, and facilitating the writing process.

There are literally hundreds of programs of the former variety on the market, focusing on spelling, vocabulary, grammar, and punctuation. Appendix B presents a listing of some of this software. Most consist of a sequence of exercises, in which the computer presents a problem, waits for the student's response, and provides some feedback, keeping records of student progress. Some programs provide practice in the context of a game

or simulation. A few of these packages are entire sequences of lessons for several grade levels, although many are sold as separate programs.

The objectives that form the basis for the skill-training modules can be drawn from the lists of rules and information-processing skills in the preceding section. A major difficulty in ensuring the effectiveness of the courseware, however, is the limitation on "transfer" of the skill from one exercise sequence to its spontaneous use in a larger text-writing context. For example, one might consider the area of spelling. There are many programs that present words in various ways to test spelling, but few if any (we have not examined them all) require the student to make the response of spelling the word correctly in translating ideas into prose.

A second difficulty is in ensuring that the exercises require practice of the skill being taught, or of its components. For example, there is a need for spelling programs that build up spelling ability by analyzing and training its component skills, which are (1) to memorize the spelling "rules" that determine the spelling of nearly all words based on their sounds and (2) to memorize the exceptions. As students in high school Latin classes have been told for countless years, the rules of English spelling make sense when understood in terms of the roots of different words in different languages. Courseware based on investigations of the specific problems that poor spellers have might have a great impact on the level of achievement in spelling.

The other category of writing courseware, software designed to motivate, support, and facilitate the writing process, uses the strengths of the computer that enable the introduction of novel teaching procedures. The most obvious members of this category are the "text editors" or "word processors," programs for facilitating text production and editing. Other types of support programs include: question generators aimed at helping the writer to plan a composition, for example by stimulating the invention processes; spelling, grammar, and style checkers; activity kits and publication systems, such as for school newspapers; information retrieval and data base systems; and electronic mail systems.

A model for the teaching of writing using software in the latter category is:

(1) having a pre-writing stimulation activity of some sort;

(2) having children write a composition on the topic;

(3) having individual teacher-child conferences to discuss problems; and

(4) having a rewrite, checking for grammar, spelling, etc., followed by final printing.

An example similar to this model appears in the work of Collins, Bruce, and Rubin (1982). Their approach is to create tools that make writing easier for children and an environment to motivate children to want to write. Some of their specific goals are:

(1) develop writing skills in the context of meaningful communication with real audiences;

(2) encourage writing for peers;

(3) provide motivation to read others' writing;

(4) encourage feedback from others;

(5) facilitate revision;

(6) help with the mechanics of writing; and

(7) enhance the thinking-writing process.

These goals have been implemented in a set of five programs, referred to jointly as QUILL: (1) a text editor for children, (2) a publication system allowing children to make printed copies of their texts, (3) a message system enabling students to send messages to students in other classes around the country, (4) an information exchange, providing an indexed storage system, and (5) an activity kit, allowing students to create games and activites that involve reading and writing.

The key to the approach suggested by Collins, et al. (1982) is to create situations in which students want to write for the same reasons that adults write. The writing that is created is not just for the teacher, but it is also for other students. Presumably, these situations will make writing become intrinsically motivating. Furthermore, by providing the tools that make both writing and revising easy, the student will feel less inhibited to write and to revise.

Possibly the most important form of impact of computers on the teaching of writing is the enhancement of the motivation of students to write. Computers help the teacher to establish real audiences for writing and they remove the substantial barrier of effort that limits the revision process. Studies have shown that students will write more, and enjoy the writing process more, when they are writing for specified, real audiences and purposes, and when they can use computers to produce professional-looking copy. The following are some of the comments from the Idea Work Groups concerning the problem of motivating students to write.

> One principal mentioned the advantages of word processing. Students come into the computer lab after school to use the word processor for writing their themes. The students are much more willing to work on editing, refining, and revising. This may be because the final production of papers is made easier; students do not have to go through the laborious task of manually rewriting their papers. Or perhaps this happens because the students are able to obtain a nicely printed version of their paper.
>
> Another advantage of the word-processor is that it reinforces the printed images necessary for reading. As the student's handwriting deteriorates, the student's image of the word deteriorates. If the student's handwriting is bad, he or she may be constantly imaging a poor replica of the word. By using a word-processor, the student will always receive a good image of the word and will be receiving repeated experience with that image; this ought to provide a better model for storage and retrieval of that word.
>
> So many of the reading, writing, and communication skills can occur in other areas of the curriculum. Reading and writing become more meaningful, more interesting, and more relevant to students when you bring in all the other areas. Children will start reading beyond their level. They have a real desire to read and write at a high level because they want to be part of a space flight, for example.

Electronic mail, or message, systems offer particularly powerful uses of computers to improve writing instruction because they provide the medium for real, not simulated, communication, and that communication must be in the written mode.

The major gaps in the range of courseware presently available to teach writing appear to be threefold:

(1) software based on new research results on the composing process;

(2) software that plays an audience role and responds appropriately to errors in writing to guide improvement; and

(3) software that manages the presentation of exercises for drill and practice which is demonstrably effective.

The first of these needs is greatest in the area of "prewriting" or "planning" skills. Two notable attempts to provide instruction in planning are (1) the program by Burns (1982), which uses various "tagmemic" heuristics for evoking appropriate ideas from students as they design compositions, and (2) the Storymaker program, by Collins, Bruce and Rubin (1982), which asks students to make decisions among various alternative sentence wordings while composing text.

The role of the computer in filling the first need is especially promising. Research hypotheses can be tested in classroom conditions without the expensive cost of training teachers in new methods that may need further revision after tryout. Furthermore, software "authoring" systems can facilitate the translation of research hypotheses into classroom exercises.

The second need is for courseware that can evaluate student writing and provide valid feedback. Skill practice programs can evaluate performance on a series of short-response exercises; however, modern theory on writing instruction holds that the entire planning/composing/review process must be taught, not merely the mechanics of writing.

The problem lies in the fact that writing is largely a divergent production task, in which a writer may generate a variety of valid responses. To evaluate these responses and give appropriate feedback, the computer program must contain, or at least mimic, a high level natural language parser (that is, "meaning comprehender"). Until this barrier is overcome, it is not likely that computers will eliminate the need for a teacher's tedious tasks of acting as an "audience" for all the students in the classroom and of grading each paper.

In the third area of need, the problem is to separate the truly effective skill practice programs from the mundane reproductions of paper-and-pencil exercises. Beyond the well-known characteristics of user-friendliness and technical adequacy, the major differentiations concern

(1) the holding power of the lessons, in terms of student interest, and (2) the match between the objectives to be achieved and the skills actually required for the responses the student is to make.

Based on the 17 Idea Work Group meetings, we have abstracted a list of a dozen suggestions for improving the quality of courseware. Some of these apply to other areas as well as to the teaching of writing.

1. Involve good writers in the design of the courseware, as subject matter experts. One thing that should be worked on is the building in of models of "expert" writers into software. These expert models could be matched to a student's writing to identify particular skill needs.

2. Become aware of and take into account the variation that occurs between teachers and between classrooms. This variation will greatly affect the use of courseware, and by preparing for alternatives, the range of classrooms in which the courseware will be found effective will be increased. The following is taken from one of the project meetings.

> There are great differences in the opinions that different teachers have about the same software. This relates to different teaching styles. The differences in styles include:
>
> teaching by objectives vs. teaching without explicit goals;
>
> varying approaches to discipline in the classroom; and
>
> different environments, such as: class size, time period, noise level (open vs. closed classroom), visual surroundings, nooks and stations vs. rows of desks, and movement of children (which depends on teacher's comfort with situation).

3. Pay particular attention to the preparation of documentation, especially including examples of the use of the courseware in a teacher's manual. A major complaint has been that courseware publishers don't really communicate to classroom teachers the way to use the courseware.

4. Especially for writing, include very flexible record-keeping, because teachers vary greatly in the kinds of detail they use to capture the level of writing ability of students. Some want very detailed records of component skills while others feel that actually interferes with teaching the writing process.

5. In developing software to teach component rules or skills, be aware of the need to be able to demonstrate that the learning generalizes to real writing contexts.

6. In describing the objectives and methods of the courseware, relate your concepts to traditional terms in order to communicate successfully. Traditional objectives were described earlier. For a set of concepts for describing the instructional methods, we recommend Judith Smith's A Technology of Reading and Writing (1978). Although this work does not focus on specific writing skills so much as reading skills, it provides many ideas about the design of both testing and practice materials to ensure that they do, in fact, teach the skill that is sought:

providing verbal mediation,
including demonstration of examples,
providing opportunity for practice,
allowing the student to complete examples begun by the instructor,
shaping the student's behavior gradually,
conforming to natural developmental sequences,
using proven paradigms where possible.

7. Include content-valid tests with the courseware, if it is designed to teach a skill. Teachers are especially happy to be able to communicate the results of instruction to parents in terms of concrete gains. Moreover, this is an important step in evaluating the effectiveness of the courseware. Of course, tests are not meaningful for some types of software.

8. Make use of the power of the computer to interact with the student in "private," thus encouraging risk-taking and reducing social pressure on performance.

9. Make use of the ability of the computer to produce aesthetically pleasing text, and at the same time consider the response mode carefully. Studies have shown that children learn to use a keyboard quickly and in fact learn other aspects of writing at an earlier age when a keyboard can eliminate the necessity for the fine motor control involved in writing by hand. However, Emig (1978) has pointed out some potential ways in which the "hand" may be important in writing. It "mobilizes" the body to perform; there is aesthetic value in highly developed handwriting; and handwriting forces a slow, linear organization which reinforces and enables particular thought patterns.

10. In order to broaden the student population and age group for which a module is appropriate, avoid unnecessary prerequisites that would exclude either very young children or handicapped students.

11. Identify weaknesses of the CRT display, in comparison with paper, in order to avoid incidentally increasing the difficulty of some aspects of writing. For example, unless a word-processor is well-designed, it is much easier to flip through the pages of a draft paper to supplement one's memory than to flip through the screens of a draft word-processor file.

12. Finally, in addition to ensuring that all the programming bugs have been exterminated from a piece of courseware, make sure that the written text and the answers to the courseware exercises both use proper English.

In the project's Idea Work Groups, many ideas for software to assist in the teaching of writing were offered, ranging from global principles to specific programs. We conclude our discussion of writing with a few of the most creative suggestions for software from these meetings.

> Here's an idea for a program. First, you let children create their own planet. This allows them to create and expand beyond what we know now. Then, you bring in information about the existing planets, presenting the facts as we know them now. Here is where you could use the computer. You could let the children do some problem-solving or do some simulations. Then, after that, it would be interesting to see what they would recreate based on initial creation and later experiences.
>
> Another possibility would be to have the students create an adventure game to go to different worlds where different rules apply and different things happen. The adventure games allow branching, decision-making, and problem-solving. Using an adventure game could be like having students read science fiction that includes some science. So many of the current adventure games have little content and limited vocabularies. But, it needn't be that way. For example, with Dungeons and Dragons, kids are fascinated to learn about Medusa.
>
> A program could be written that included some "canned" prompts to guide students' writing of a story: What is your setting; what is your initiating event; what is the goal? This might result in the child producing a well formed story, but it might prevent the child from doing any problem-solving or any creating of the story.
>
> You would need to have some notions as to development beyond the initial "canned" production, notions that are usually provided by a teacher. But, one teacher responded that she was not worried, because she was certain that there is probably some creative programmer who could design a responsive system (given the right inputs from a teacher).
>
> Word-processing is a wonderful tool for all ages. Hopefully, it will get down to the younger ages. When you see how much children labor over making the letters, it seems much better to give them word processing. Even if they have to hunt and peck, at least a "d" looks like a "d." Writing becomes a real success. Children's manual dexterity gets in the way of their writing and thinking about their writing. It frees their creativity if they do not have to worry about actually

producing the word. Also, it makes it easier for the teacher to read and understand what the student has written.

We could do this by having a big map. Rather than having the screen scroll or go away, there are little doorways that you can go to. So you could make the story spatial. You could color code the parts of speech. For a young child you could have a room where they can go to pick up a word for their story. In the room, there may be a "cat" and a "ball" and a "bike." And all of the nouns would be red, and the verbs would be green. So the children would see the structure of the sentences visually.

There will be super-dictionary-thesauruses, in which one can access the resource with very simple and vague inputs and browse to find the word you want, with its spelling. The tip-of-the-tongue problem may be solved. Communication can be made more exact because we can rely on the computer to prompt us with just the right word to convey our meaning ... and spelled correctly.
Grammar correctors are needed, too. This is not so easy because grammar overlaps into semantics and requires a processor that can recognize semantic as well as syntactic transformations.

One idea is to generate whole writing lessons, but with a clear emphasis on particular grammatical skills. Students are told to write a composition but that their grade will depend in large part (or completely?) on the avoidance of specified types of errors and the inclusion of specified types of structures.

An alternative is to employ grammar games. This is more efficient than the inclusion of a skill in a much larger task in which its occurrence will be just a small part of the whole and in which it will not be repeated in a variety of contexts.

Another idea is to teach the concept of "grammatical rules" by providing a context in which students create their own rules for some game. Grammatical rules become the object of thinking and problem-solving. There is some question, however, about the extent of transfer of skills from any "games" to actual writing.

One solution to the problem of good writing instruction is to get students to read more good literature, more examples of well-written documents. An example of software might be to take some well-written passage; put all the needed information on a diskette; give the student the communication task which the writer confronted; and compare what they produce to the model essay. As a small example, one might leave out single sentences, to be filled in.

## 2. Developers' Incentives

Schools must recognize that high quality courseware development is costly. Anybody with a few hours training in BASIC can quickly write a program that will present a list of correctly spelled and misspelled words and give students scores and feedback for correct identification of the misspellings. Developing a program to teach spelling skills well is an entirely different project, however, involving analysis of cognitive skills, drafting programs to teach the identified skills, tryouts with children, revision and debugging, and preparation of supplementary materials. These are costly procedures, and neither publishers nor individual developers can be expected to expend the effort needed for high quality professional reading and writing courseware without the expectation of a reasonable return on their investment.

Computation of the expected return on investment depends heavily upon the size of the market. But this computation is complicated by several factors. The basic characteristics of the school market are an unknown to most small developers and publishers; and even when these characteristics are known, they are certain to be in a rapid state of change due to hardware and software developments and the unpredictability of teacher acceptance. In addition, even when the size and characteristics of the school market are known, determination of the return on investment is complicated by the problem of software piracy. The following sections of this report will attempt to deal with the major components affecting developer's incentives: knowledge of the school market and the problem of software piracy.

### School Market

Determination of the return on investment depends heavily upon the size of the market. But, for developers and publishers considering work in the educational market, other factors are also critical. Generally, the following create problems for developers and publishers:

(1) lack of knowledge about the future of microcomputers in the schools;

(2) existence of a wide range of computers that cannot use each other's software; and

(3) lack of market guides that identify and describe actual software purchasers.

For these reasons, as well as the market size, some developers and publishers are focusing on the business and the home markets while they wait to see what happens in the schools. As was stated by one Idea Work Group participant:

> There has not been enough unanimity in education. Therefore, we have invested in business software. If the schools could come to some consensus about what they would like, then we might be willing to make some investments in software developments for public school systems. This is no doubt true for other developers and publishers.

Future of microcomputers in schools. Quantitative estimates of the current and future size of the school market have been made to a certain extent by the Technology-Assisted Learning Market Information Service (TALMIS), which reported $11 million in educational software sales in 1981, for roughly 130,000 computers in the schools. TALMIS predicted that by 1985 there will be at least 450,000 computers in the schools; and it is reasonable to expect that nearly every schoolchild in America will soon have some form of access to microcomputers or a computer terminal. Some industry analysts expect sales of educational software to reach $8.7 billion by 1990. Yet education remains a tiny fraction of the total software market.

Johns Hopkins University recently conducted a study entitled the National Survey of School Uses of Microcomputers. The study provided national estimates based on a stratified probability sample of 2209 public, private, and parochial elementary and secondary schools in the United States. The survey revealed that, by January 1983, 53% of all schools in the United States had at least one microcomputer for use in instructing students. Furthermore, there is a high rate of increase in ownership. Figure 6 shows the percentage of elementary and secondary schools with microcomputers over the period from 1980 to 1983. As can be seen from this figure, secondary schools are more likely to have a microcomputer than are elementary schools. By January 1983, 85% of all high schools, 77% of all junior-senior combinations, and 68% of all middle and junior high schools

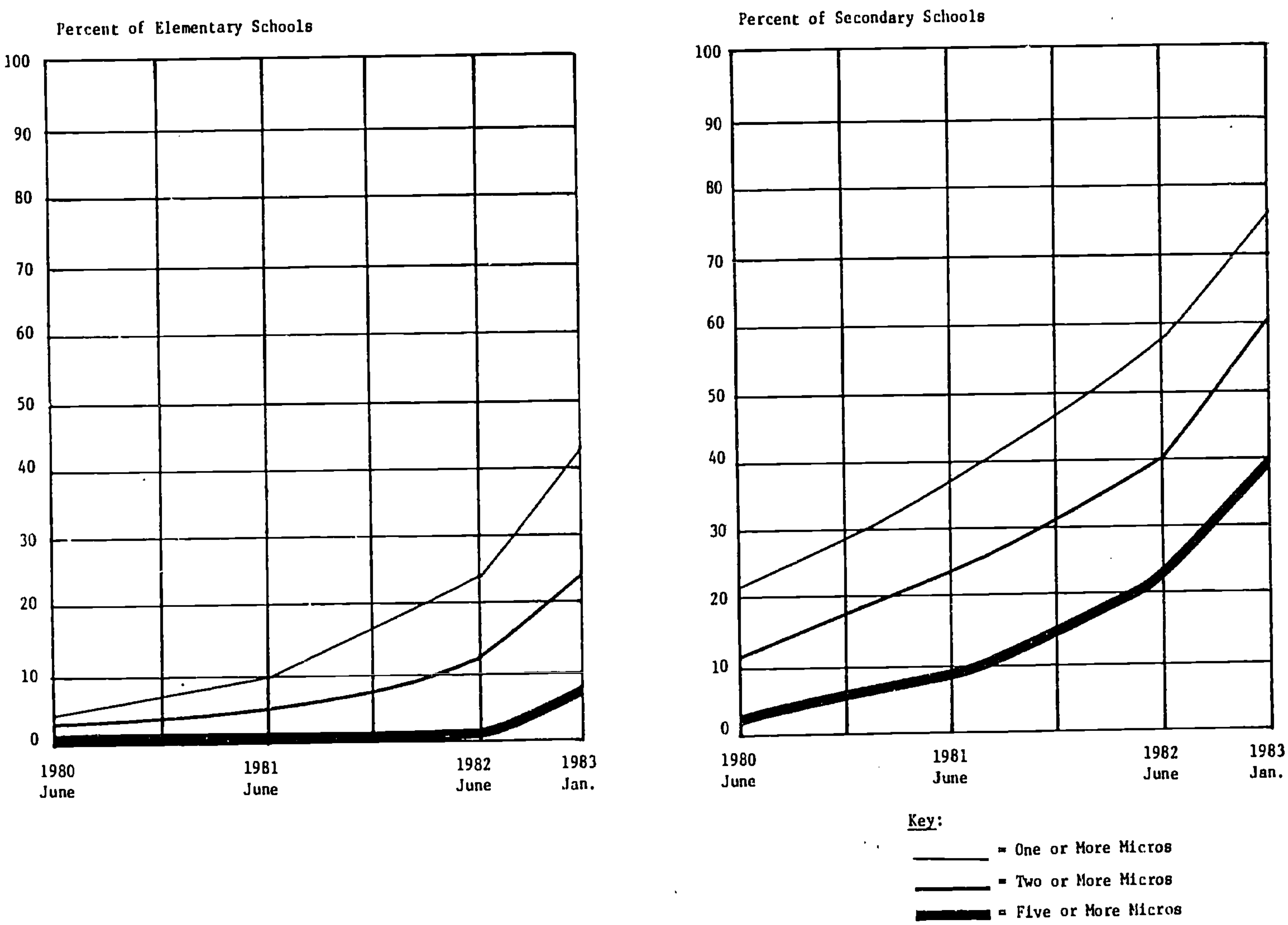


Figure 6. Microcomputers in Schools: 1980-1983
(Taken from School Uses of Microcomputers, 1983.)

had one or more microcomputers. In contrast, only 42% of all elementary schools had a microcomputer.

The typical microcomputer-owning elementary school has two microcomputers. Each one is used for about 11 hours of instruction per week for an average of 22 hours of use per week. An average of 62 students share these 22 hours of use, resulting in about 20 minutes per user per week. Approximately 40% of all instructional time is spent by having students use computer programs for practicing math and language facts, spelling drills, and various other memorization tasks. Approximately one-third of the instructional time is spent having students copy, write, and test computer programs. The remaining time is spent playing games under the direction or approval of the teacher, with many of these being "learning" games.

The typical microcomputer-owning secondary school has five microcomputers. Each is used about 13 hours of instruction per week for an average of 65 hours of use. An average of 80 students share this use, resulting in about 45 minutes per user. Two-thirds of the instructional time is spent on programming and computer literacy activities. Another 18% focuses on "drill-and-practice" activities. The remaining time is devoted to "learning games," various advanced applications such as word processing, science lab work, and business courses, and other activities.

Compatibility. The size of the courseware market does not depend solely upon the number of microcomputers in the schools. There are an increasing variety of microcomputers on the market, and this variety is finding its way into the schools. Schools with more than one computer often have more than one type. One important consideration for such schools in making software purchases is whether the software will run on all, some, or none of its computers. Many leading computer-using educators have recommended that schools make software purchase plans first and then buy the appropriate hardware. However, once purchased, the hardware can be used for a wide variety of software, and the availability of that hardware will affect future software purchases.

The increasing variety of computers in the schools means that, for the developer/publisher, programs written for one microcomputer can only sell to a portion of the market. There are two approaches to this problem: (1) to use portable (or universal) languages for software development, or (2) to select a single computer for software development. Although software can be

designed in transportable languages, such as BASIC and Pascal, this approach limits use of the specific powers of each computer. The other approach to this problem is to determine the predominant machine and then create software for that machine. Assuming some return on that first piece of software, then work can be devoted to translating the software for other machines. Given this approach, the developer/publisher should focus efforts on the most popular computers in the schools.

According to the Johns Hopkins University National Survey of School Uses of Computers, the computers that are predominant in the school are the following: Apple, TRS-80, Commodore Pet, Atari, and TI. A recent study of computers in education (Sherman, 1983) indicated the following percentage distribution of computers in schools: Apple (49.5%), TRS-80 (14.4%), Commodore Pet (14.2%), terminals to mini-computers (12.9%), TI-99 (3.7%), Atari (1.8%), and other (3.6%). Clearly, for the current school market, the developer should consider development of software compatible with the Apple.

However, other computers are gaining popularity in both the home and the school markets. For the greatest coverage within both markets, a piece of software needs to be compatible with more than one computer. So, the wise developer will devote efforts to translating the software to make it usable with a number of machines. This is not always a simple matter, but in most cases, the translation does not require as much effort as the production of the original software. One developer, who participated in an Idea Work Group, estimated that 80% of the original development effort is saved in translating to a new computer.

From the standpoint of both developers and schools, the prospects of having software developed for different machines is not very appealing. The developer, already wary of the school market, may not be interested in devoting efforts to translate programs for different machines. The schools are not interested in having to purchase several different versions of the same software merely to run on different machines. Also, teachers and administrators identify other benefits in having compatibility in different computer systems.

> Perhaps communication with other computers will become even easier in the future, so that it will not matter whether a student has an Apple, a Commodore, or a TRS-80. The different

types of microcomputers may become even compatible with each other in the future.

Also, all of these microcomputers may be able to download off an IBM mainframe computer, and then students will be able to communicate with each other.

Thus, there is a need for software to be compatible, so that it can run on all types of microcomputers. This would solve the market-size problem for developers and would allow the communication possibilities desired by school personnel. One solution suggested in an Idea Work Group is to obtain a standardization of hardware. With continued competition among hardware manufacturers, this is not very likely in the near future. Another solution is in the development of some other mechanism such as "emulators." These mechanisms function by allowing the user's computer, for example the Commodore 64, to act as if it were a different computer, such as the Apple IIe. If the concept were fully functional, emulators would allow the owners of any computer to run software developed for any other computer, after purchasing an appropriate "emulator." Another type of solution would be to develop an authoring language that can be used on any computer. Current efforts are underway on such a project; however, it must be recognized that this solution may suffer from the same problems as that of the "universal" languages, in that the language may place limits on the developers' use of the computer. Any or all of these solutions would provide the developer/publisher with a larger school market, with presumably a greater return on investment.

Knowledge of market channels. Another problem for small developers/publishers and those who have never dealt with the schools is that of how to approach the school market. The developers and publishers who participated in the Idea Work Groups and in the private discussions provided several suggestions for those considering the school market.

"First, the developer/publisher much decide on the target audience: the primary audience and the secondary markets." The school market is multifaceted, and the developer/publisher must be cognizant of the various segments. Consider, for example, a piece of software designed to teach some vocabulary concepts. The developer must be able to identify the ages and grade levels for which the package is appropriate, and whether it is appropriate for special populations of students such as the gifted or those

in special education. In addition, the developer should determine what specific skills will be learned by students using the package and how both the student and teacher will benefit from having the computer teach these skills. These need to be phrased in terms that are familiar to teachers and school administrators and that indicate ways in which the software is relevant to skills currently being taught in the schools.

As part of the effort to define the target audience, the developer should initiate a "dialogue" or interchange with the schools. The developer might begin with a market survey, contacting teachers and reviewing relevant articles in the computer magazines to determine what schools need. Next, the developer should create a prototype version of a good idea and try it out with students. At a later stage, the developer should test a refined version of the courseware in some selected schools. As stated in one Idea Work Group,

> The development of the Bank Street Writer gives us an example of the process. In this case, a key factor was the involvement of teachers from the very start. There was a clear objective: to provide a word-processor for young children. There were also educational assumptions. For example, they assumed that the writing process and the editing process should be kept separate.

The example of the Bank Street Writer indicates that the involvement of teachers resulted in two distinct outcomes: the development of needed educational software and the creation of a market for that software.

Another part of defining the market involves finding out where the hardware is concentrated. Here are some suggestions for finding out about available hardware:

Contact state boards of education,

Contact the large school districts, like New York, Chicago, and Los Angeles, and

Contact the Department of Defense schools. (As one developer noted, "you can consider them to be one of the big cities.")

The purpose of this contact is two-fold. First, the developer can determine what type of hardware prevails in these locations. For example, currently,

eleven states have a statewide buying agreement with Apple and, the Department of Defense schools have Atari equipment. A developer planning to create software for the Apple or the Atari should target these locations to identify software needs. A second purpose of the contact is to promote sales of current or future software. Large school districts or state agencies may provide a guarantee for a reasonable market. As stated by one developer, the large districts "may be the most difficult to sell to, but once they are sold on your product, you have it made. And once they are sold, your product will filter down the system."

For small developers who are not interested in handling the distribution of their software, the next step involves locating an interested publisher. Some software publishers offer the developer a flat fee for the product, while many other offer royalties. The Software Writer's Market (1982) provides a listing of most of the software publishers. It describes the market that they serve and indicates the type of arrangements that they usually make with developers and authors. Edwards (1983) suggests several reasons why royalties can be attractive for both the publisher and the author. For the publisher, they:

(1) lower the initial cashflow, and therefore, the risk of bringing a product to market;

(2) enable the publisher to publish a greater number and wider variety of programs;

(3) allow the publisher to pick and choose its authors; and

(4) are a known, controllable expense.

For the author, royalties provide:

(1) the possibility of large economic returns;

(2) freedom to pursue the projects he or she wants to, without the need to accept fulltime employment;

(3) incentive to create an excellent product in a timely manner; and

(4) an easily understood, equitable compensation arrangement.

There are problems with the royalty arrangement. Because the publisher cannot control the developer's activities, it becomes more difficult to

create a "family" of related products. The author, on the other hand, receives little security. If the product does not sell, he or she will realize no income.

At this point, there are no standard industry-wide contracts, although the royalty agreements resemble those found in book or record publishing. The major components of these agreements include (1) the royalty percentage, (2) a sign-up bonus (available from some publishers), (3) the timing of the distribution and marketing rights, (4) the publisher's responsibilities, and (5) the author's responsibilities. In addition, the developer and publisher must agree separetely on the rights for the following: (a) the right to make copies, (b) the right to distribute copies, (c) the right to develop derivative works, (d) the right to display the software, and (e) the right to perform the software. The final agreement will depend on such factors as the negotiating skills of the publisher and the author, the type of product, the amount of follow-up support from the author, the amount of marketing and technical support from the publisher, and the reputation of the author and the publisher.

For developers interested in capturing all of their program's revenues through their own distribution and marketing, the next step in dealing with the school market is to identify the distribution channels into the schools. Several different channels exist, and the developer/publisher should evaluate their potential for his or her product. One possibility is to work directly with retail computer, book, or educational supply stores. But, as one developer stated, "working with the retail stores is very frustrating because of the turn-over in staff." Another possibility is to sell directly to the users. This may work for specialized markets, but dealing with numerous schools may overwhelm the developer. One developer, participating in an Idea Work Group, tries to discourage direct sales by suggesting that the school/user go to a local dealer. Instead of going to schools or to retail stores directly, the developer can use the services of companies that specialize in distributing software for schools.

The software distributors provide services to the retail stores, in order to facilitate getting a developer's software into the stores. As stated by one developer:

> The distributors 'nourish' the retail stores; they have a tollfree number and talk to them every day; they have a 'hot list' of products; they sell to the dealers by credit.

Other developers made some specific suggestions regarding the use of software distributors:

> Our company works with 14 distributors in the country. The largest is Softsel with 2400 dealers.
>
> Another outlet is through the school distributors. Bell and Howell, for example, used to have marketing rights for the Apple to schools. They have regional training centers. They are interested in selling hardware; so in some cases, they will place ads about software in order to sell the hardware.

Rather than going to distributors, the developer may prefer to use market services directed to the schools. For example, Katz Marketing are specialized school marketers.

> They are specialized sales reps for schools, which Apple and other manufacturers have throughout the country. They have a cadre of trainers, and they are making a package for those interested in training and hardware. They are doing things like point-of-sales displays in the stores for the 'Learning Center': and having workshops. They are promoting the product and providing the training with it.

However, the developer should realize that having sales representatives and marketers limits the use of distributors.

> The sales reps have their territory. Even if a store in Los Altos sells the software, using Softsel, the sales rep will take a commission. So, you have to go either with the sales reps or with the distributors.

In addition to the distributors or sales representatives, the developer needs to focus on publications. These publications include software catalogues as well as magazines, newsletters, and journals that are targeted to the schools.

> Ideal School Supply, for example, supplies 50 to 100 catalogues; they sell to Creative Publications, Lake Shore, and other mail-order houses.

We (a developer) target the magazines that go to the people with purchasing power in the districts; these people are not necessarily the teachers. The magazines include Electronic Learning and Curriculum Products Reviews. These magazines go to the people who are buying in the schools; these are read by about 200,000 buyers in the schools.

Developers can use all these publications for advertising software. The magazines, newsletters, and journals have the added advantage of allowing developers to submit articles to increase their visibility within the schools.

Having identified the distribution channels, the next problem is that of "making your product stand out." It was estimated that Softsel, for example, probably receives 300 products a week. Since it costs them to list a product and to answer questions about it, they want to be certain that the product will sell. The same is true of publishers who may be considering licensing the rights to a developer's software. The following are suggestions for attracting notice from potential publishers or distributors:

(1) Use superb packaging. "Something in a little plastic baggy just will not sell."

(2) Use theme characters. Then the developer can design the next products using those characters.

(3) Create a series of products from easy to hard. Then the developer will have a client base for future products.

This last suggestion is important for developers to use in planning a product line. To maximize the return on investment, the developer should identify follow-up products for the customer or client to buy after the first product. As stated in one Idea Work Group,

> Don't think of a software sale as a one-shot purchase. Plan on extended versions and altered versions to be purchased in a series. Every sale is of an edition that is going to be improved upon.

The developer should use promotional efforts to make the customer aware that the first product is part of a family. It is just the first in a series, and the customer should be looking for the after-products. This approach tells the customer that the developer will be around for a while and that this is a company that will have something even better next year.

Software Piracy

There are roughly 3,000,000 children at each grade level in the 100,000 elementary and secondary schools in the country. At one rosy extreme, where each courseware unit is usable by one student alone, a 10% penetration of the potential market would yield roughly 300,000 sales per year; this would allow inexpensive pricing for courseware requiring even hundreds of thousands of dollars to develop. If a single copy of a courseware unit is bought by each school and copied or networked, however, the corresponding sales estimate (over all years!) for that unit are reduced to fewer than 10,000; this would require expensive pricing to recover several hundred thousand dollars of development costs, when added onto per-unit production costs. As schools' income is proportional to their enrollment, very large schools would be able to purchase even expensive courseware, while smaller schools would be pushed out of the market by high unit prices. To the extent that prices can be related to number of pupils using the courseware, a larger market will develop, and greater incentives can be provided to developers.

Most software today is copyrighted. The copyright law was amended in August, 1980, to protect computer programs from copying in much the same way in which the law previously protected other works of authorship. The thrust of the amendment to the copyright law is to put limits on the circumstances under which a purchaser may make a copy or adaptation of a purchased computer program (Marshall, 1980; Gore, 1980); for a thorough summary of the legal issues and the recommendations of the National Commission on Technological Uses of Copyrighted Works (CONTU), see Olmstead (1980). Thus, copying of software beyond a single backup copy violates the law. (See the legal opinion summarized in "Fair is fair," InfoWorld, 1981.) The matter is not straightforward, however. As MacLennan (1980a), points out, copyrights are "quick, simple, dirt cheap to obtain and require little monitoring. Here the good news stops.

> ...Copyrights advance the social interest but retard the economic rights of the software developer because of limited protection. Sometimes even the program itself is not protected."

In a follow-up article, MacLennan (1980b) presents the opinion that "substantial uncertainties exist as to just what law applies to software, and to what extent." He offers advice to software companies and programmers to "...review their legal position, carefully minimize liabilities, and simultaneously attempt to provide maximum customer-oriented documentation."

The problems raised by illegal copying were discussed at several of the Idea Work Groups. Most developers and publishers felt strongly that it was an important problem that must be solved if the educational software industry is to continue to exist, but a few felt that the problem is not so severe or that it will diminish soon. A few of the comments are listed below.

> The perception is that all schools copy.
>
> The high cost of software encourages software piracy.
>
> Many developers, however, are reticent to make great investments to develop a good piece of software only to have it stolen.
>
> Software developers, in fact, may be moving away from the educational market because of the widespread piracy problem. They know that teachers are looking for good products and lots of technical support, but they are not willing to pay what it costs to get it. They also know that teachers are not adverse to piracy.
>
> In general, publishers have not yet developed any mechanism for handling the needs of schools for multiple copies, and this reflects the fact that most of them do not see the schools as a big market in comparison with business and with games. Teachers are left with no guidance and the opportunity to copy.
>
> Copying poses less of a problem for software developed for businesses. Businesses are in competition, while there is some cooperation among schools.
>
> The school market is different from the business market. First, competing companies are not likely to share with each other. Second, even within a company there is likely to be a greater understanding of the meaning of the development costs than in a school.
>
> Why do these problems not apply to the home market? There is less of a network of users in homes than in the schools. It is

a quantitative difference. If you could sell 30,000 copies of a piece of software, then you are less concerned about the copying question.

Thus, with copy protection less than assured, and with alarming tales in the public media of pirating engaged in by school children, sometimes encouraged by adults (see, for example, Doyle, 1982), publishers are understandably reluctant to circulate preview copies to schools (most commercial software for microcomputers is not copy-locked at the time of this writing). To copy an unlocked diskette costs about $2 (for a blank diskette) and about 10 minutes time.

The entry of local networking into schools adds another dimension to the problem. In a network arrangement, a number of computers are linked to a mass storage device and other peripherals. Upwards of 30 computers linked to a central memory can thus share devices such as printers. In addition, these computers can share software. The advantages of a local network to the classroom teacher or computer lab instructor are measured in terms of setup time and convenience; down-loading a software image electronically requires less effort than inserting multiple tape cassettes or diskettes manually. Another important advantage of the network is a reduced investment in software. In this situation, the software is not being copied onto diskettes, and the position of the copyright laws is not clear. However, the net impact on the software producer or publisher is equivalent--namely, a great reduction in the anticipated number of copies of a single program or disk which vendors expect to sell to the schools.

Illegal copying and the presence of networking reduces developers' incentives to expend the effort to enhance the quality of their educational software. For unauthorized copying (otherwise known as "piracy" and "theft"), there appear to be three broad categories of solutions:

(1) prevention of copying, by convincing teachers, by storage in non-copyable form, or by including associated materials that are not as inexpensively copied;

(2) subsidizing the development of courseware; and

(3) institutional purchase/lease or license agreements;

Solutions 2 and 3 also apply to the networking problem. Each category of solution has its own problems. Let us consider each.

Prevention of copying. This approach, if possible, would be ideal, and it is its failure to date that has created the existing problem. Copyright laws exist but copying persists (Bayer, 1981; Edlin, 1982; Gilder, 1981; Morgan, 1981; Sturdevant, 1982; Wollman, 1982). Basically, as long as courseware is stored and transmitted on floppy disks and cassette tapes, the only absolute barrier to extremely inexpensive copying is the use of nonstandard formatting. However, sophisticated "lock-picking" programs are likely to be continually improved to defeat the use of nonstandard formatting. Lock-picking programs have raised a controversy of their own. Many industry periodicals have refused to accept advertising from their producers. However, their introduction into the market has also pushed software producers and publishers to offer low-cost or free backup copies of their programs to purchasers, and to respond in other ways (most notably, improved post-sales support) to the threat posed by bit-copy programs. Alternatives to the use of nonstandard formatting of diskettes that might be considered for preventing copying include:

- convincing teachers not to use illegal copies;
- storing courseware on programmable read-only memory (PROM) or other less copyable media;
- providing (a) supplemental material with the courseware (such as a workbook or a keyboard overlay) that is either "used-up" or at least not so cheaply copied and (b) special services that are responsive to teacher concerns.

The first approach to prevent copying, convincing teachers not to make illegal copies, may be enhanced in various ways but must always be considered only to be a partial solution. One software manufacturer has implemented this method by including an advertisement of a reward for turning in "pirates" with each disk it sells (Electronic Learning, 1982).

Since 1980, publications directed toward the classroom teacher have carried numerous articles raising the issues and problems associated with illegal copying and pointing out the implications to future software development for the school market. (See, for example, Elman, 1982; Finkel, 1981; Letellier, 1981; Weiner, 1982.) The dialogue prompted by this

education process has begun to penetrate the school community and to produce some beneficial side-effects. Primarily, it has begun to make producers of software for the school market aware of and responsive to certain characteristics of the school market. For example, a survey of 68 Apple software producers conducted in late 1981 and early 1982 (Hoover & Gould, 1982) revealed that nearly three-fourths of those responding do NOT provide backup copies at all; and a full three-fourths said that the purchaser could not preview the software prior to purchase. Such wide spread lack of understanding of schools' needs virtually invites illegal copying within the school community. On the other hand, the same survey revealed that 83% of the software producers were willing to negotiate a special price with a school district or work out a licensing arrangement for multiple copies. With the arrival of local-area networks on the school scene, such licensing arrangements may become commonplace. Since 50% of respondents identified illegal duplication as a serious or moderate threat to profits, software producers may become more open to steps that they can take to make alternatives to copying more appealing to the education community.

Convincing teachers not to make illegal copies can only be effective if software developers acknowledge that teachers have some unique concerns which vendor sales policies must address. Making backup copies readily available at low cost and allowing educators to preview software before purchase may be effective solutions to the copying problem in schools. The following are some comments from the Idea work Group participants:

> There is some defense for the user who wants to make some copies. If, for example, the teacher has based the lessons on some diskette and something goes wrong with the diskette, he or she doesn't want to wait two or three weeks to receive a back-up copy from the publisher.
>
> A question was posed as to whether it would be sufficient for the teacher to be able to call the publisher and order a replacement to be sent immediately by overnight mail. One teacher responded that he had waited for six weeks for a replacement diskette. Perhaps, if there were enough commitment to make it work, it would work. However, after the first negative experience, the teacher would not rely on the responsiveness of the publisher.

Computer-Using Educators (CUE) in California have suggested that back-up rights should be part of the purchase agreement. But, not all publishers agree with this stance: (1) with the back-up rights comes the potential for making several copies, and (2) users feel that they should not be charged for receiving more than one copy, while publishers do not agree.

Another problem involves getting software on an evaluation basis. Some companies are resistant to allowing schools to do that. If developers and publishers don't trust teachers, then they won't be able to make the sales.

Teachers as a group generally feel a moral obligation, because of their position, to do what's right. Having been informed that copying software is wrong, illegal, and likely to have adverse consequences, they may be more willing to comply if the software producers establish marketing policies that show that they are sensitive to teachers' concerns. Software in a classroom of 30 fifth-graders is far more likely to be damaged than is a single copy used by an adult for business or recreation purposes. The fact that so few vendors offer backups can be frustrating to a teacher with a tiny budget for materials and many students to serve. Similarly, schools should not be expected to purchase courseware for classroom use without first previewing the materials. At least one publisher has demonstrated responsiveness to the school market by announcing the following policy:

- 30-day preview privileges with full refund;
- $5 per disk for backup copies (at the time of purchase);
- $10 per disk replacement fee (return old disk, no questions asked);
- demonstration disks available on a loan (free) basis;
- all software copylocked; and
- availability on Corvus and other hard disks under consideration.

The Idea Work Group discussions indicated that teachers are beginning to realize the negative consequences of software piracy. And, they are beginning to take some actions to reduce the occurrences of that piracy.

The schools are beginning to realize that there is a moral issue here. Many school districts are making policies that all software must be purchased. In one school district, for example, they are almost making it a matter of a person's job. If there is any illegal copying, the principal will problably lose his or her job. So, the principal is going to make sure that the teachers are not making illegal copies.

Copying is a severe problem for publishers. One solution is to put a minimal "lock" on the diskette to eliminate casual copying. This would eliminate the vast majority of cases. The expert who wants to rip you off can do it in any case. Most school districts do not condone piracy explicitly, and what is needed is to make them aware that turning the other way allows illegal copying. Merely putting a sign on the wall saying "don't copy" will not stop copying. There must also be some expectation of sanctions for illegal copying. Nevertheless, the publisher must take some steps.

The prevention of copying cannot work without the voluntary compliance of schools. The legal approach of surveillance and enforcement is costly to all parties concerned. In addition, most publishers, although acknowledging that illegal copying persists, refuse to enforce the copyright because of "bad publicity."

There is, however, a recent example of a lawsuit against a copying company and some university professors concerning illegal copying of book material for classroom use. It should be noted that this lawsuit was filed by the Association of American Publishers, which represented nine publishing companies in court. Such a major lawsuit against a "pirate for profit" (see, for example, Hayman, 1981) may carry the weight of example for schools that engage in a lot of illegal copying; however, it is an unrealistic solution against the large number of "small-time," amateur copiers which some say are responsible for the large majority of piracy activities.

The second approach to prevent copying, storage of software on PROMs, suffers primarily from the cost of unit production. A floppy diskette is quite inexpensive and easy to copy onto, so it is an especially efficient medium for storing and selling courseware. However, only a small crucial part of any courseware need be uncopyable; so, storage of that part on a chip or a peripheral card may be both sufficient and inexpensive enough to warrant serious consideration.

Several different hardware scenarios were discussed by the participants in the project's Idea Work Group meetings.

Piracy is a critical problem. An advantage of the videodisc is that it prevents the pirating that is easily accomplished with videotape. When software is available only through a databank or on a videodisc, then the market will really open. There is an analogy to the record and cassette industry. Similarly with the copying machines, now no one thinks twice about duplicating papers and books. This is really stealing someone's ideas and not paying them for all their hard work. This means that the developers and publishers are not getting fair return on their investment.

Consider the scenario of the game parlor applied to the schools. A program that is on a dedicated, stand-alone will be more immune to piracy. This is the format that has been used in the past by Computer Curriculum. Here you are selling a service and not a product.

Package the software into special inexpensive machines.

Hardware encryption--a device costing about $100 that would allow the user to make copies of software for that particular computer unit, but those copies would not run on any other computer unit. This would enable the publisher to set a reasonable price for the software. It requires a commitment by hardware manufacturers to put these devices into the machines and to perform the overhead functions of keeping track of serial numbers and so forth. Until the sales of hardware become software-driven, there is no incentive for the hardware manufacturers to incorporate these devices.

Another possibility would be to create a system in which it would be necessary to insert a quarter each time the software is used, or downloaded from a master file. This could work in the home or in the school. A ROM chip could be included in the computer that would count the number of uses of a piece of soft-ware, for billing purposes. This is like the way jukeboxes record plays for determining royalties to performers.

The third approach to prevent copying, embedding the software in a package of materials and services, is a viable solution only if the supplementary material (1) is really necessary for using the courseware and (2) cannot be shared by several students. Informal conversations with large numbers of educators over the past two years have led us to the conclusion that this avenue deserves serious consideration by the software industry. The present relative abundance of school-directed software has made teachers hungry for computer-based curriculum materials. Such packages may include teacher's manuals, student manipulables, or other relevant materials as well as diskettes or cassettes themselves; in addition, they may be keyed to the scope and sequence which each school district adopts and to existing texts already in use in schools. Curriculum packages integrating the computer

activities into the existing school curriculum would, we believe, be well received, and would provide a pedagogical basis for the development and marketing of consumable materials as part of a school software package.

The courseware package may also include special services provided by the publisher to the schools. These services would only be available to those who had purchased the software. In several Idea Work Groups, participants agreed that the developer or publisher should focus not so much on the diskette as on the service provided along with the software. Here are some excerpts from the meetings.

> Perhaps what should be offered are services, not just programs on diskettes. Series of programs that are continuously updated and upgraded may provide a cure for piracy, because they will be available only to bona fide buyers. IBM has had a long-term policy on mainframe computers of forcing acquisitions of updates in systems, by not supporting old versions.
>
> Another approach would be to have some sort of fee or subscription system. Perhaps this could be a cooperative or a lending library. By paying this initial fee or subscription, you would be allowed to make the back-up copies that you needed. This would permit you to examine the software and would remove the conflict of wanting the software but not being able to acquire it. In addition, you want to keep updating your own library so that you are not stuck with the old version of the program.
>
> Another approach is to make a special school package. For one developer, this package includes a back-up, task cards, objectives for the teacher, a description of the processing done in the program. So, there is a school version that gives the schools what they need--how to manage the computer in the classroom, how to extend the concepts into the curriculum. This package will be sold at a higher price than for the same program sold to the home market.
>
> Another approach is to use a licensing agreement. In this way, if a publisher finds that someone has been making illegal copies, they can demand that the software be returned.

Subsidizing development. A second major approach for increasing developers' incentives to create high quality courseware in the face of illegal copying exists. Rather than attempting to prevent copying, this alternative focuses on breaking the link between copying and the incentive to develop software. One way to do this is to subsidize courseware

development. Subsidizing development involves paying for the courseware development "up front," rather than through royalties. Its primary disadvantage is that the free marketplace cannot operate to select the most effective courseware on the basis of performance. Choices must be made based on past performance and stated plans. Furthermore, little incentive exists to improve the courseware beyond initially accepted specifications. And, who will do the subsidizing? Many in our society have long held that the federal government should not be in the position of funding the development of (and thus "dictating") curricula. Current courseware development does not require a large hardware capitalization; and although a single teacher cannot create a full-blown reading program in a month, a group of teachers and programmers may create one over a year or two in their spare time. A free market stimulates such innovative efforts, while central control, in the form of funding subsidies, may stifle these efforts.

In considering this solution, then, we ask the following questions. Can the benefits of the free market be realized when there are development subsidies? Can certain parts of development be subsidized, or can efforts contributing to the development of quality software be subsidized, in much the way that this project was solicited? Are collaborative endeavors between industry and education, "seeded" by private funds, a viable option for developing quality courseware? Is there a reasonable source for subsidies, or can such a source be created for this purpose?

The possibility of federal and state support for software development was discussed at several of the meetings. Selected excerpts from these discussions follow.

> The Terrapin experience with LOGO was that most of the development undertaken by Bolt Beranek and Newman and the Massachusettes Institute of Technology was supported by grants from the Office of Naval Research and from the National Science Foundation. That opportunity is not possible for private developers relying on sales to consumers.

> In the sixties, publishers got things handed to them because of the federal investment in educational innovation. Without this kind of federal investment, how can we expect publishers to do more than make simple adaptations of existing textbooks?

> An important federal contribution would be to support a grants competition for the development of good software for

> education. This would provide the opportunity to spend the needed developmental effort to get a substantial range of excellent software into the schools. On the other hand, education must compete with other social services for a dwindling federal budget for human services; and the number of people with children in schools is also declining.

Further discussion of the federal role in supporting software development appears in the final section of this paper.

Institutional purchase agreements. This third approach to increasing developers' incentives attempts to create a guaranteed market for the courseware. If sales can be made in sufficiently large blocks to state education agencies or large local education agencies, these courseware sales can include provisions that can serve to dissipate the effects of copying on the market, such as the licensing of multiple copying of disks for a fee. An example was provided in one of the Idea Work Groups:

> Richmond Public School district is a very large district. The purchase of a single piece of software for all the schools within the district would be prohibitively expensive. This situation can lead to software piracy. A company could reduce the piracy problem by offering a package deal to the school or by offering a certain number of copies at a reduced rate or by allowing copying for a certain fee.

Indeed, with an expected rise in the use of local-area networks to link large numbers of microcomputers together within a school or district, institutional lease/purchase or license agreements may become the preferred solution.

Cable TV stations are exploring the feasibility of delivering computer programs to their subscribers. Delivery of software via cable may, in the long run, offer the most satisfactory solution of the copy-protection problem in schools, requiring as it does a subscription fee for the use of the service. Institutional subscription fees could be established at a level which ensured the software producer a reasonable return on investment, while at the same time guaranteeing widespread availability to the user at a reasonable cost.

An example of this kind of arrangement can be found in the state of Virginia. Recently, the Virginia Department of Education signed a contract with a telecommunications company to study how to set up a system for

transmitting computer software via radio airwaves. With this system, Virginia schools would be able to transmit computer programs and other information over National Public Radio subcarrier airwaves without the usual telephone modem. Rather then a modem, users would lease a kind of radio receiver to receive the signals and copy them onto computer diskettes. The cost of leasing would be about $25 per month, plus royalty charges for the commercial software used.

There are many questions to be explored with respect to this type of solution, however. What proportion of sales for use in schools will be to districts large enough to make institutional licensing agreements attractive, rather than to individual teachers? Would a system like that being explored in Virginia result in an acceleration of the home as an educational institution rivaling the school? Is most of the effect of copying within or between schools, districts, or states? Will large purchasers insist on the right to copy? Will discrimination between large and small purchasers lead to an imbalance in availability of courseware to students? What form of lease/purchase or license agreement can simultaneously protect the industry's right to make a profit on its products and protect children's right to equal access to a quality education?

As a final observation, it should be pointed out that not everyone agrees that piracy is a crucial dilemma. One software producer suggests that piracy costs to the industry are predictable and should be factored into the price of software (Freiberger, 1982). A software dealer points out that some popular programs which he sells are copyable, yet this doesn't appear to have affected sales (Easterling, 1982). And a publisher's representative reminds us that the educational software industry has several other substantive issues to surmount before the educational market can become profitable for vendor, teacher, and student (McMillan, 1982). The quality of sound and graphics in educational software must be improved substantially before it can be considered an adequate medium for various educational applications, for example, spelling, reading acquisition, and foreign language instruction. Also, considerable effort needs to be invested in exploiting the unique characteristics of the computer as a medium of instruction, instead of using the computer merely as an expensive page-turner.

## 3. Teachers' Incentives

Just as schools must become aware of the costs and risks that developers encounter, developers can increase the acceptance of their product by understanding the needs of teachers. A major barrier to the implementation of computers in instruction, particularly in reading and writing, is that of teacher resistance. This resistance stems from several different fears about the technology. One teacher identified several of these fears. "There is the fear of the unknown. This includes fear of mechanical problems. It includes fear of the costs of the equipment and software. It includes fear that there is too much to learn."

Fear of being replaced by the computer is a fear that has received some attention. As Pogrow (1982) has pointed out, there are two levels of implementation of computer-based education: (1) low-leverage, in which the computer is a tool to be used by teachers in a setting not very different from the traditional classroom and (2) high-leverage, in which the computer replaces the teacher in either a home or school setting. Studies of the implementation of computer-assisted instruction have indicated that one of the greatest obstacles to this implementation is that of teacher acceptance (Rockart & Morton, 1975). One cannot expect teachers to be enthusiastic, however, about a technology that threatens to replace them.

Teachers recognize that it is impossible for them to respond simultaneously to individual student needs when working with 25 or more children in a class, and one answer is for technology to provide some assistance. The computer can assume part of the role of gatherer, analyzer, and user of information about individual students; and although introduction of computers is not likely to result in teacher replacement, it provides an impetus to review and refine the role of the teacher in the classroom.

The introduction of computers can lead to changes in the classroom, and changes in a stable situation can be expected to create anxieties for many teachers. According to one teacher, "the real barrier is the teacher's concern about changing. The teacher has been doing the same thing successfully for many years. Now, we are suggesting changes; these changes will be met with resistance. By suggesting something different, we are suggesting that it can be done better."

Closely related to the problems created by changes in the classroom is that of a "lack of time." This is particularly true of new teachers. One developer/researcher mentioned the results of a study examining the factors affecting the implementation of new approaches in the classroom. "During the first two years of teaching, teachers are merely trying to survive; they are really not concerned with the students. If you ask them to do something new, they will respond with, 'I don't want to talk to you; I just want to get through until Tuesday.'" Although the problem of lack of time may be exacerbated for new teachers, these sentiments are shared by others. Computers are perceived as something else to be done in an already crowded curriculum. In the words of one teacher, "the elementary teacher has a very limited time for instruction in any one subject area. Besides being responsible for covering math, science, social studies, literature, reading, writing, art, and music, there are numerous administrative tasks." In addition, teachers are being held accountable for student results in certain areas; and, not surprisingly, these areas tend to become overemphasized. This leaves little time for the consideration of a new technology and a new approach to teaching and learning.

Furthermore, among teachers in the humanities may be a fear that expanding technology signifies progressive dehumanization (Holmes, 1982; Wilson, 1981). "Some language arts teachers are resisting the use of computers. There is a feeling that computers are just fine--for math and science teachers. They feel that they are people-oriented and that the computer is a cold, heartless machine." Grossnickle, Laird, Cutter, and Tefft (1982), for example, reported typical comments of teachers who failed to use available microcomputers:

> I know nothing about computers--if they cannot grade my papers, I have no use for them.... -- English Department
>
> I have other priorities.... -- Art Department
>
> I don't know capabilities or their value in our subject area.... -- Driver Education. (p. 18)

(It should be noted, however, that this fear of the destructive nature of technology has afflicted us for some time: Socrates is reputed to have feared that written documents would result in the decline of memory (Gerard, 1969)).

Even if there is not a fear of the technology, there may be the perception that the computer is not appropriate for teaching reading and writing skills. Many teachers believe that there is a lack of software that fits into and expands the current curriculum: "The lack of software is a barrier. There is a perception that there isn't anything that you can do on a computer in reading. Actually, it is not so much a lack of software but a lack of quality software." The following are some examples of problems that teachers have noted:

> The software was supposed to teach blends, but it included examples that were not blends. Whoever was writing the materials did not know enough about phonics to know the difference between a blend, a digraph, and a vowel sound.
>
> The program was to teach predicting, but the item had the answer included in the passage.
>
> Another problem related to that of quality was that of a lack of prerequisites. Some programs include key concepts in the testing that were not a part of the student's schema or background.
>
> Another problem appears in the materials directed toward comprehension. The questions tend to focus on testing general knowledge rathar than knowledge of the content of the passage. In many cases, you can give the students those questions without the passage, and they can answer them.

This problem results in a vicious circle. Teachers are reluctant to make investments in software for reading and writing, because the software does not appear to be very exciting. Developers do not invest their resources to producing creative, high-quality software for reading and writing, because there does not appear to be a good market.

To foster teacher enthusiasm, guidance must be made available to educators interested in promoting computer use among their colleagues and to potential courseware developers concerned about structuring their products to foster teacher enthusiasm. There are four broad categories of strategies to consider in promoting teacher acceptance of courseware: (1) teacher training, (2) inclusion of positive teacher roles in the lessons, (3) care in ensuring user friendliness and (4) development and use of evaluation standards that address teachers' needs.

Teacher training and retraining must be undertaken to introduce teachers to computers and to convey the message that the computer is a tool that can be used to help both teachers and students. Including positive teacher roles in courseware requires a level of exploration of the human-computer teaching team that has heretofore not been undertaken. Although Wall and Taylor (1982) have developed plans for writing instruction that involve the team concept (the computer acts as a word-processing tool to facilitate the teacher's feedback and the student's editing of compositions), a systematic exploration of the possible ways in which courseware can benefit from teacher inputs and control is needed. User friendliness has been discussed by various courseware evaluators. In this particular case, we will focus on the match between the skills and tasks with which teachers of reading and writing are familiar and that are useful for implementing computer-based instruction and on tactics for raising the likelihood that these teachers will like the courseware. Critical to the implementation of teacher training, the incorporation of teacher roles, and the focus on user friendliness is the development of evaluation standards for courseware. These can be used by developers in designing courseware to meet teachers' needs, as well as by teachers in reviewing available software in order to make wise purchase and use decisions.

## Teacher Training

Increased knowledge and information about microcomputers may dispel certain teacher fears. Boyd (1981), Petruso (1981), Rose (1982), and Townsend and Hale (1981), in discussing strategies for overcoming resistance to microcomputers, suggested the need for teachers to be informed about computers and their potential use in the classroom. A common suggestion is to hold inservice programs with workshops and presentations to demonstrate the microcomputer's capabilities. Inservice training should be an important component of each school district's plan for infusing courseware into the classroom setting. It must be recognized that such inservice training must be conducted at several different levels, depending on the needs and experiences of the teachers.

Inservice training for "beginners." Inservice training for beginners needs to start with a focus on the fears and reasons leading teachers to

avoid computers. As one teacher stated, "there is a need for a 'therapy' session in which philosophical concerns are brought up." And teachers need to "realize that they can plug in the computer and play with it without worrying that they are going to break it." In addition, these sessions need to give teachers hands-on experiences with computers. They need to be shown how to run a program and given the opportunity to try it themselves. Allowing teachers complete access to the machines will encourage them to have a sense of ownership.

Morrissey (1982) provides a list of principles to aid in developing a feeling of competence among teachers:

(1) Teach the students to think of the computer as a general purpose machine, not a brain.

(2) Provide successful introductory experiences.

(3) Emphasize the importance of good instructional design in effective computer-assisted instruction.

(4) Avoid programming at the beginning (p. 50).

Similarly, Rawitsch (1981) lists three principles to be applied in teaching educators about computing:

(1) Ease into the technicalities of computing through the use of concepts familiar to the learner's previous knowledge.

(2) Don't distract the learner's grasp of fundamental concepts by introducing extraneous technical information.

(3) Use an organizing model to give the learner the "big picture" before discussing the detailed pieces (p. 31).

In addition, Spero (1982) described a successful approach to teacher training that allowed teachers to keep a personal computer at home during the 10 weeks of the workshop. Such training can reduce initial anxieties, alter teacher attitudes toward the new technology, provide situations in which teachers can feel comfortable and capable of handling microcomputers, and assist them in developing strategies for incorporating microcomputers in their classrooms.

Anadam and Kelley (1982) recommended a guiding principle for teacher training--humans first and computers second. In most cases when discussing computers in instruction, the enthusiastic advocate glorifies computer capabilities to such an extent that human qualities appear as insignificant. This approach will rarely encourage teachers to incorporate computer technology into their classroom activities. A critical message that must be conveyed to teachers is that it is a human being who conceives, develops, implements, and evaluates educational activities, even when these are presented by a computer. "Advanced computer capabilities haven't reduced the intellectual effort required to construct and test instructional models....They haven't alleviated the need for artistry and creativity in the preparation of effective and motivating instructional experiences" (Allen, 1978, p. 61).

As for specific topics to be included in such sessions, Nansen (1982) recommended designing a course with five main objectives: (1) to help teachers feel comfortable using the computer; (2) to familiarize teachers with good quality educational software; (3) to show teachers how to copy both single programs and entire (unlocked) disks; (4) to teach them how to set up a computer from scratch; and (5) to introduce them to a variety of professional or "utility" programs. In addition, based on the experiences of several school districts, such inservice training for beginners should cover:

(1) simple trouble-shooting rules, such as checking whether the computer is plugged in;

(2) simple commands, such as LIST, RUN, and CATALOG;

(3) procedures for software purchases;

(4) sources for supplies, equipment, and repairs; and

(5) administration policies on copyright and software piracy.

These topics will introduce teachers to the use of the computer and provide them with practical tips on how to begin to use the machines. This level of training is insufficient in itself to promote effective use of computers in the classroom, however. For effective use of the computer in instruction, teachers need to be given a second level of training--training for users.

Inservice training for "users." At this level, the teachers have overcome their initial fears of the computer and are actually interested in some level of implementation in the classroom. The training at this level must deal with (1) options for computer use, (2) software evaluation procedures, and (3) classroom management.

Teachers need to be shown that different options exist for the use of the computer in the classroom. Options exist as to the placement of the hardware: in the classroom, in a separate computer lab, or on movable stands. Demonstrations of the outcomes of these various options can be presented through videotapes. In addition, different options exist as to the types of software that may be employed within any of these settings, from drill-and-practice programs, to games and simulations, to general instructional support. Awareness of these options can be provided through live classroom demonstrations or through the use of videotapes. Options also exist for the pattern of student use: individually, on a scheduled or free-time basis, or in groups. Most important, options exist for the subject matter to be taught using computers. Teachers need to be made aware that the purpose of computers in the classroom is not merely to teach students about computers or to extend math and science curricula.

An important component of preparing teachers to use computers in the classroom for instructional purposes involves providing them with procedures for making decisions about the appropriate pieces of software to purchase and use. One might think that decisions about software purchases can be made by using the published reviews of software. Unfortunately, these reviews sometimes do not agree with one another. One teacher suggested a strategy analogous to using reviews of movies. "You have to read several of them and read between the lines to decide if this is for you. No review really covers all the bases."

So, in addition to examining the published reviews, teachers need to be given guidance in personally examining the software. They will benefit greatly by being shown examples of software of different types and quality levels. One suggestion might be for the district or region to provide a list of courseware that some teachers have found effective. This can help teachers to avoid excessive reliance on the marketing techniques of publishers, distributors, and retailers. Such a list should not substitute, however, for careful, personal evaluations. Teachers should be allowed time

and access to "play with" alternative courseware packages, much as they would examine alternative textbooks.

Teacher training must also address issues of classroom management and the expected changes in the teacher's role. Anadam and Kelley (1982) stress the importance of sy c planning on the part of the teacher. They identify a three-lev el of teaching as an integrated activity. Planning, including th elopment of objectives and procedures, constitutes the first le Execution, a component following planning that cuts across the objectiv d procedures, represents the second level. The third level includes two pa llel components: (1) teacher-student interactive activities, and (2) student independent study. It is within these two components that computer technology can be introduced to the students in the classroom. It is important that teachers recognize that the computer activities shoul not be isolated from other activities at each level.

Certain other aspects of the teacher's role, as well as the arrangement of classroom furniture may also need to be considered. Classrooms are designed for group instruction of 20 to 40 children, and they operate within a fixed time frame. Computers can offer individualized instruction for a diverse student population; thus, most educational software tends to assume an individual student as the user. Individualized instruction and students learning at their own rate appear to be incompatible with the traditional classroom setting--providing limited access to a computer within a fixed time frame. Students do not begin at the same time, do not proceed at the same rate, do not end at the same time, and do not reach the same levels of performance. A more flexible modular classroom is required to take advantage of the computer's power as a tool for individualized instruction. One model suggested in our meetings was the "learning stations" approach, in which students move between learning environments, one or more of which may involve a computer. The teacher's role in "execution" becomes transformed from that of a lecturer to that of a supervisor of activities for individual learners and small groups of learners.

Without the computer, the teacher's role includes such activities as assembling and disseminating information; constructing, administering, and grading tests; identifying student errors and providing some remediation; and evaluating and revising curriculum materials and the teaching approach.

Frequently, the "execution" aspects of teaching consume much of the available time, leaving relatively little time for needed "planning" activities. The use of computers in the classroom should provide a delivery mechanism that reduces the burden of many of the teaching functions. This will enable teachers to devote more time and energy to developmental efforts, particularly those focusing on the socialization and human interaction skills.

It is not clear how much time is required for the first two levels of inservice training. One or two days may be sufficient to get most teachers past the hurdle of initial use. However, Slesnick (1983) describes a 90-hour teacher inservice course in computer education provided at the Lawrence Hall of Science in Berkeley, California that captures aspects of the inservice for beginners and for users. The course has three main goals:

(1) To give teachers a broad base of knowledge in computer education rather than to provide mastery in any specific area of computer science;

(2) To motivate teachers to initiate a computer education program in their schools; and

(3) To provide teachers with activities that can be used in their classrooms.

The course includes programming (35% of the course), graphics (19% of the course), off-line activities related to computer work (19% of the course), examination of software (8% of the course), and examination of courseware (18% of the course). Because programming ability is not really required for teachers to make effective use of computers to teach reading, writing, or other topics, a total of 50 hours may be a good estimate of the time needed for inservice training.

<u>Inservice training for teacher/developers</u>. Criticisms have been raised concerning the focus of many computer training programs for teachers. Many times the instructor is a computer specialist who focuses the instruction on computer programming. But, most teachers do not want to learn programming; they want to learn how to use the hardware and the software. There are, however, a few teachers who become so interested in computers that they want

to go beyond the initial training for "beginners" and for "users." These teachers need a course to help them learn how to design and implement their ideas through software.

In the initial stages, this training can focus on the use of authoring systems and higher level languages. The availability of higher quality authoring languages is an important ingredient of this step. Eventually, some teachers will want to go beyond the limitations of these languages. In both cases, teachers need to be given methods (e.g., phone numbers) to gain access to experts to provide answers to their specific problems. An important concept that emerged from our Idea Work Groups was the powerful potential to be realized from forming courseware development teams. Teachers who want to develop courseware need to be brought together with both computer experts and educational researchers. This is a formula that will lead to practical, yet creative innovations in the use of computers in schools.

## Teacher Roles

Although teachers can learn how to select courseware that will roughly fit their needs, it is still necessary for courseware designers to plan for teachers' use of the materials. The essence of the concept of including a role for the teacher in a courseware unit is that the courseware can be implemented in different ways by different teachers, and that the success of the implementation, in terms of student outcomes, is at least in part due to teacher variation. In planning their courses, some teachers consider their role to include that of modifying the available curriculum material for the students in their classroom. Many basal programs fail to prescribe exactly what the teacher should do with the materials. Furthermore, such programs typically include practice on a particular skill once every few months. Such designs force the teacher to revise the materials and the presentation sequence to focus on what he or she considers to be important. In addition to such "required" modifications in curriculum materials, some individual teachers feel the need to manage and direct the learning situation as completely as possible (Purdy, 1975; Rose, 1982). The following are some comments from teachers who participated in the Idea Work Group sessions:

> What is needed in software is to give the teacher more control. The software that will sell the best will be the software that gives the teacher control over the program without that teacher having to be a programmer.
>
> Schools are not so much interested in buying a curriculum as in buying materials that can be adapted to fit their curriculum. An authoring system is not necessary. What is necessary is the ability to make modifications in the program so that the teacher can individualize it and make it relevant to his or her class.
>
> What is needed is a wide range of options for the teacher. There needs to be a basic program that can be run by the teacher who does not want to touch the computer. In addition, there needs to be a component that allows the teacher to modify the program.

There are two approaches to the inclusion of potential for teacher variation: (1) including provisions for teachers to make choices in the design of the lesson, and (2) including provisions for teachers to provide crucial input during the lesson (including overnight inputs in multi-day lessons). Provisions for teachers to alter the design of lessons vary along a continuum from making small alterations to designing the software. At one end, programs can be designed to allow teachers to set a few parameters, like presentation speed or amount of practice. At the next level, teachers can modify programs by entering content, such as spelling or vocabulary lists to be presented by the computer. At the next level teachers can select software to enable them to design entire instructional sequences. In the latter extreme, the "courseware" is essentially a computer language, such as PILOT or PASS, designed for the writing of instructional sequences. These authoring languages can allow teachers to use their special knowledge about subject matter, pedagogy, and children to design a software program without having to endure the intricacies and tedium of computer programming (Wagner, 1981). In our Idea Work Groups there was a general consensus, however, that the value of authoring languages, at least in their present form, is limited. Authoring languages must be made more flexible, in order to allow teachers' creative ideas to be implemented.

The second type of provision, allowing for teacher inputs during the lesson, presents a very challenging problem; on the surface, it appears to

defeat one of the strengths of the computer--its ability to attend to one child completely while the teacher focuses on the remainder of the class. It seems clear that the form of the teacher's input should be designed so that it can be adapted by the teacher to the classroom situation. There should be alternatives so that the instruction is not required in real time but, rather, can be made when the teacher chooses. For example, a word processor might assist the teacher in writing instruction by indicating spelling or grammar errors to a student directly. The problem of indicating the need for better transitions, however, would be left to the teacher, who would examine a composition while the student is engaged in other activities.

## User Friendliness

User friendly software is critical for teacher acceptance. And, what does user friendly mean? "A program is user friendly if a fifth grade teacher can try it out successfully without the necessity of struggling through a set of elaborate instructions." User friendly software is self-directing. It provides a self-explanatory menu that directs the user to the available options. The documentation, if needed at all, is completely understandable; it avoids computer jargon. And, above all, the program is dependable.

Further recommendations can be found in the literature concerning the critical aspects to include in a program to make it user friendly. For example, School CourseWare Journal suggests the following as questions to use in evaluating software packages for their user friendliness:

Does it always use paging (not scrolling)?

Can the user control the speed of presentation?

Can the user control going back to a previous frame or leaving the program at any desired time?

Is the screen output easily readable?

Is the reading level correct for the target student?

Does it use correct grammar, spelling, hyphenation, and punctuation?

Does it accept abbreviations for NO (N), YES (Y), etc?

Is it individualized, e.g., does it change to meet individuals' differing needs?

Is the dialogue personalized?

Can the student get help?

Can the program be easily modified to suit individual teacher and student needs?

Can you get to the program "guts" to examine program coding?

Does the program never ask for more than one data item at a time?

In considering this listing of components, there would appear to be three basic approaches to user friendliness: (1) avoiding requirements that the teacher or student perform actions that he or she finds difficult or annoying; (2) providing byproducts that eliminate chores teachers traditionally find annoying; and (3) including components in the courseware that correspond to relevant preferences of teachers.

Some of the requirements that teachers find annoying, and that lead to avoidance of the computer, are:

(1) the need to type;

(2) the need to program;

(3) the need to rely on a technology that can break down;

(4) the need to schedule students for individual sessions;

(5) the need to find a place in the classroom to put the computer, or the need to walk down halls and/or find keys to computer rooms;

(6) the need to read operating instructions written by computer programmers; and

(7) the need to give up some of the authority for providing feedback to students.

Each of these applies only to some teachers and to some courseware, of

course, but the problem for promoters of courseware is to reduce all of these annoyances to the point that very few teachers find any unfriendly requirement of the computer.

Various articles on the development of educational software have recommended inclusion of components in the design that relieve the teacher from some activities. The most frequently suggested activities are recordkeeping with the accompanying report generation, and the control of practice sessions with the slowest students, who would otherwise use up a disproportionate share of the teacher's attention.

Finally, there are possibilities for enhancing the image of courseware specifically for reading and writing instructors. For example, it would be appropriate to include lists of conversation-starting questions, or interesting quotations from good literature, or information on useful resources.

## Evaluation Standards

In order to increase the level of acceptance of courseware by teachers, developers must design courseware that meets evaluation standards based on teachers' needs. Numerous evaluations of software can be found in the literature. At least four different organizations focus on courseware evaluations -- Courseware Report Card, EPIE, MicroSIFT, and School Microware Reviews. MicroSIFT, for example, was created by the Northwest Regional Educational Laboratory. It includes the following basic information on each instructional package:

| | |
|---|---|
| Version | Instructional purpose |
| Producer | Instructional techniques |
| Evaluation completed | Documentation available |
| Cost | Instructional objectives |
| Ability level | Instructional prerequisites |
| Subject | Content and structure |
| Medium of transfer | Potential user |
| Required hardware | Major strengths |
| Required software | Major weaknesses |

In addition, the package is evaluated by three or more reviewers who are representative of potential users of the courseware package. These reviewers use the following dimensions for their review:

Content is accurate.

Content has educational value.

Content is free of stereotypes.

Purpose of package is well defined.

Package achieves defined purpose.

Content presentation is clear and logical.

Difficulty level is appropriate to audience.

Graphics/sound/color are used appropriately.

Use of package is motivational.

Student creativity is effectively stimulated.

Feedback is effectively employed.

Learner controls rate and sequence.

Instruction integrates with prior learning.

Learning can be generalized.

User support materials are comprehensive.

Use support materials are effective.

Information displays are effective.

Users can operate easily and independently.

Teachers can employ package easily.

Computer capabilities are used appropriately.

Program is reliable in normal use.

The published review provides a summary rating on these dimensions using a five point scale: strongly agree, agree, disagree, strongly disagree, and not applicable.

Some of the reviews appearing in journals and magazines use the dimensions developed by MicroSIFT. For example, School Microware Reviews, produced by Dresden Asociates, publishes evaluations of educational software for the APPLE, ATARI, PET, and TRS-80. (The reviews are created from evaluation forms shown in Appendix C.) The summary rating, "relating to objective items on the form and not to the evaluator's subjective comments," is similar to that of MicroSIFT.

Other journals use different criteria and different procedures for the review. For example, School CourseWare Journal published a list of questions to be asked when evaluating and comparing software packages. The following are the major categories from that list.

Is the program demonstrably "user friendly"?

Is the program "user proof"?

Does the program do best what the computer does best?

Does it provide the printed documentation required for effective use in the classroom?

Is additional documentation given so the program can be used to teach programming?

Is quality assurance given?

(The full set of questions can be found in Appendix D.) Another example of guidelines for courseware evaluations appears in The Apple Journal of Courseware Review. For these reviews, evaluators must consider the following:

Instructional design;

User level;

Documentation;

Support of educational process;

User interest; and

Programming quality.

In addition, the evaluators are asked to respond to certain questions. How critical is this subject matter? Is this an innovative use of microcomputer technology? Would you recommend this program? What improvements, if any, could you suggest to strengthen this package in content, program, or documentation?

Taking a slightly different approach, Lathrop (1982) presented ten major reasons for automatically rejecting a software program.

1. Audible response to student errors.
2. Rewarding failure.
3. Any sound that cannot be controlled.
4. Technical problems.
5. Uncontrolled screen advance.
6. Inadequate instructions.
7. Errors of any kind.
8. Insults, sarcasm, and derogatory remarks.
9. Poor documentation.
10. Denial of a back-up copy.

In our meetings of teachers, developers, researchers, and publishers, some suggestions were made as to characteristics to be avoided in software. These are listed below:

1. Software that duplicates a workbook.
2. Software that rejects logical attempts by students. It should say, for example, "'KT' is a good way to try to spell 'cat.'"
3. Software that overreacts to either right or wrong answers.
4. Software that includes outright errors; for example, a program that identifies predicate adjectives as verbs.
5. Software that simulates a personal conversation with the computer. The conversation is really with the programmer, and the programmer should construct the conservation to reflect this.

This brief review indicates the variety of sources, as well as criteria, for evaluations of educational software. Nevertheless, both teachers and developers indicate the need for some standards in evaluations of software to ensure that high-quality software will be identified. The following are some comments from participants in the Idea Work Groups:

> Teachers are extremely eager to find good-quality software, but they are not sure how to evaluate the software or even how to judge the quality of the software evaluations produced by others.
>
> Almost every school district in the country that is actively looking for software, however, has put together its own evaluation instrument.
>
> Having standards for the evaluation of software is not only good for the users, but it is also beneficial for the developers. The comments received from credible reviewers could then be taken by the developer to signify meaningful reactions from the education community. The developer could use this information to improve or refine all future development efforts.

An important contribution to the whole field would be the development of some standards for software evaluation.

Based on the inputs from the Idea Work Group participants and from the National Advisory Panel, we can provide some suggestions. An important point to emphasize is that software can be used in many ways in different settings. Thus, a review by a single person from a single perspective is not likely to cover the range of applications adequately; it should be evaluated by more than one person from more than one perspective. An initial screening should be done to determine whether the software runs properly and whether it is factually correct. If it passes this initial screening, then it should be tried on students at the appropriate grade level to examine children's reaction to the software. The evaluation with children could focus on whether a particular skill objective was achieved. However, such a focus might limit the development of software to those types in which gains can be easily measured using preand post-tests. Instead of or in addition to providing data on gains, a review should identify actual uses of the software in classrooms where teachers have considered it

effective. This could be accomplished through a local, state, or national software directory.

The following provides an outline for the components of a "Consumers' Guide to Software."

1. Purpose--a description of the purpose, making it clear as to whether it is skill practice, tutorial, etc. This would include information as to the target audience and prerequisite skills.

2. Effectiveness--a statement as to whether the teachers' objectives were accomplished. The data could include field tests, testimonials, evaluation studies, etc. In any case, they should be qualified by descriptions of the settings in which the courseware was tried out.

3. Usability of the Software--answering auch questions as: do students find it appealing; and can teachers adapt it to their classroom easily?

4. Quality of the Documentation--considering three types: a teachers' guide, a tutorial, and a reference manual.

5. User Friendliness--including reported "bugs" in the program as well as annoying limitations.

6. Hardware Requirements--indicating needed machines and peripherals, as well as whether it can be used in a network.

7. Price--including provision for back-up copies.

Using these dimensions, evaluators could rate individual software packages, as well as compare several packages that purport to focus on the same skills. This would provide the potential user with an evaluation report similar to that appearing in Consumer Reports. As this system became used, additional dimensions could be considered for inclusion. Only those dimensions that significantly contributed to teachers' or administrators' purchase or use decisions should be included.

## RECOMMENDATIONS

### Federal, State, and Local Support

Technological advances have placed the United States in the forefront of the world market and world power. These advances have been achieved as the result of earlier commitments to education for American children. These commitments came not only from local school districts but also from federal agencies, state agencies, and business and industry. The implementation of technology in the classroom must receive support from all these sources. The following paragraphs describe the role of each of these parties.

### Federal Role

The federal government must be involved in accessing the magnitude of the need for action at all levels. For example, one publisher noted that most of the problems facing education in the U.S. are already being addressed by other countries. Concern was expressed that, if the United States does not attend to these problems, we may fall behind in the marketplace. In addition, the federal government needs to be involved, because there needs to be some consistency in policies and in their implementation. Computers are expensive. That means that the gap between the "haves" and the "have-nots" may get bigger and bigger. If left to individual states, some places will move ahead, and others will have nothing. Thus, equal access to technology for all students is critical and is an issue that must be addressed by federal and state policymakers. Furthermore, members of Congress, judges, and any other decisionmakers in the legal process must be educated in order for them to be able to make informed decisions about issues in technology, such as those involving software piracy.

Leadership. The federal government can provide the needed leadership. "Setting technological priorities to maintain our role in the world market and to remain in competition will take real leadership at the national level." This leadership need not necessarily be in the form of large funding commitments, rather it can take the form of crystallizing opinion, focusing the direction of future developments, and disseminating

information. "We need the federal government to provide some national goals and emphasis." "The federal government should make a national commitment...This could be similar to the commitment made at the time of Sputnik."

Research. A major role for the federal government to play in supporting the introduction of technology into the school is that of funding basic and applied research. Numerous research questions abound whose answers can stimulate and shape software development and implementation in the classroom. The following are a listing of some of the research questions that have been raised:

What are the cognitive demands and consequences of computer learning? Are there certain prerequisites? What advantages result from using computers to teach reading and writing skills?

What changes occur in the composing process when students use a word processor as compared with paper and pencil?

What effects do these new technologies have on learning styles?

What are the long-term effects of computers? What happens to children who begin to work on computers in kindergarten?

From a developmental standpoint, when is the best time to introduce children to a computer?

What are the optimal models for using the computer to teach which skills? When should children be alone on the computer and when should there be cooperative learning situations?

What are the motivational consequences of using the computer? Do students become more interested in writing with the word processor than with paper and pencil?

Will reading comprehension skills learned on the computer transfer to books?

What are the social consequences of using computers in the classroom?

Software Development. "The government should go beyond funding research. Funds should include support for development of materials for the classroom and for implementation of those materials." The marketplace is currently handling some software development. However, as one publisher noted, "The publishing business is not very good right now, and it probably

will not get much better." It is estimated that the amount of money that schools spend on instructional materials has declined, in the past 10 years, from over two percent to less than one percent of their budgets. In this climate, publishers are unwilling to take risks on pioneering developments. Quality software takes much time and money to develop. What is needed is funding from the federal government for the development of such pioneering efforts. One possibility might be to provide guaranteed loans, to be paid back from future sales of successful courseware.

One developer commented that the area of reading and writing is an unknown to their staff. This may be the major reason that quality software does not exist in these areas. The formation of teams to work on this development might be the most successful model--teams that include creative software designers, classroom teachers, and leading researchers. There is, however, a problem with public domain software: the developer reaps no benefit from its dissemination, and publishing companies are reluctant to market it. Society will benefit through the publication and distribution of creative, meaningful software; so to attract creative developers, the federal government should allow the private distribution and marketing of the software. Perhaps in addition, funded developers might be required to "pay back" society through a certain number of training workshops.

Evaluation. Support is needed not only for courseware development. Evaluation of available software is also critical. Numerous packages currently exist on the market, and these are accompanied by numerous reviews. Several different approaches could be taken to assisting schools with the problems of knowing what is available and worthy of consideration. One model would be for the federal government to develop standards or criteria for evaluating software. Perhaps this might even result in the development of some consistency or compatibility among systems, as has occurred in audio-visual equipment. Another model would have the federal government serving as a central clearinghouse. At the national level, a central source could develop a listing of the available software, categorized by subject area, with references to the published reviews and evaluations. Another model would be to establish a mechanism similar to the National Diffusion Network to recognize and distribute good software, particularly teacher-developed software. Another model would be to have the

federal government supporting state or regional resource centers. These centers could assist schools in evaluating and implementing software.

Training. Teacher training presents a major problem for the incorporation of computers into the schools. The federal government could make a valuable contribution by supporting some of the teacher training costs. In the immediate future, funds are needed for inservice training. Federal funds could be used to support universities or even local school districts to provide needed training. According to one school principal, the analogy here would be with the science institutes that were funded in the sixties.

Funds are also needed to support preservice teacher training. As a step in this direction, the federal government could sponsor a conference for deans in schools of education; the purpose would be to encourage these schools to require computer literacy among their students.

Hardware Support. Many teachers and developers agree that "the Computer Education Act, known as the 'Apple Bill,' should be passed by both Houses of Congress. A lot of schools are waiting around for the outcome of the Bill before they get into computers." In addition to assisting schools to acquire computers, the federal government could become involved in trying to encourage cable systems to link the home and the school. Other countries, like England, France, Germany, and Japan, are moving in this direction.

## State Role

In addition to a national commitment to encouraging computers in education, there need to be individual state commitments. "States must deploy their own resources and set their own priorities." One example comes to mind immediately. Minnesota's leadership in educational technology is due in large part to the contribution of the Minnesota Educational Computing Consortium (MECC).

A state commitment to technology in education may, in fact, result in encouraging and stimulating the growth of high technology industries within a state. As reported by Charles Minshall of Battelle Institute at the annual meeting of the Southern Regional Education Board (SREB) held in June 1983, high technology industry tends to blossom in areas where people like to live and areas with cultural attractions, good neighborhoods, and good

schools. So, encouraging technology in the schools may contribute to the economic, as well as educational, growth of a state.

Evaluation/software adoptions. Currently several states provide some support to local schools, and incidentally to publishers, through statewide textbook adoptions. Teachers, developers, and publishers recognize the benefits and the problems with such adoptions. Statewide adoptions of software would provide some assurance of a reasonable product to schools and of a reasonable market to developers and publishers. On the other hand, adoption standards tend to be "way behind the more progressive school districts" and may prohibit some creative developments. Nevertheless, there seems to be consensus for state agencies to take the lead in identifying good courseware and in pushing for changes to improve courseware quality.

Resource Centers. As part of a federal effort at evaluation and dissemination of educational software, states should establish statewide or regional resource centers. These centers could provide special services to school districts, such as establishing a courseware library or assisting in teacher training. In addition, these centers could facilitate the broad dissemination of software developed in the schools through the creation of electronic mail networks.

Training. As part of the federal assistance in teacher training, states should provide funding for teacher training efforts. This might be viewed as part of ongoing state programs to attract high technology industries. (See the congressional Office of Technology Assessment description of such efforts in "Technology, Innovation, and Regional Economic Development.") As one example of a state investment in training, the Virginia legislature has allocated $120,000 statewide for teacher inservice training. The resource centers mentioned above can also assist in this teacher training effort. In addition, states must focus on the incoming teachers. "States must set teacher certification standards in the area of computer literacy and technology."

Hardware support. States can supplement the federal initiative to provide schools with support in acquiring software. Some states, such as California and Virginia, have already passed their own versions of the "Apple Bill." These states provide a tax write-off to companies that donate computers to the schools.

## Local Role

The local school district is the place where implementation of computers in the classroom will either succeed or fail. The development of a long-term plan for the introduction and implementation of technology into the schools provides a mechanism that can lead to success. As stated by one school administrator, "we must have a long-term plan to implement computers. And we must watch for mistakes; they will happen and are part of the process."

What are the components of a long-term plan for technology in the schools? The first component should be an assessment of the needs of the district, focusing on the needs of the students and the community. Based on this needs assessment, goals and objectives for the program can then be developed. As part of this statement of goals and objectives, decisions must be made as to skills needed by students, level and inclusiveness of student, of teacher, and of parent involvement, time schedule for implementation, policy regarding illegal copying, and management of the system. Evaluation must also be a component of this long-term plan. The evaluation will examine both the process and the outcome of the implementation. Thus, the system will require a continual evaluation that will lead to modifications of the long-term plan.

## Role of Business and Industry

The previous discussion has emphasized the need for a national, state, and local commitment. In addition, there needs to be a commitment from business and industry. "Business and industry clamors for schools to teach students the skills needed when they get into the work world."

Communication. Many educators, characteristic of the above quote, believe that business people are extremely critical of the schools. Educators should improve communications by holding meetings or discussions with business groups. Such sessions can help to identify what it is that business and industry want from the schools. In addition, it may be possible to identify points for mutual benefit--"things that businesses can do for the schools and things that schools can do for businesses."

Hardware and software donations. Business and industry can assist schools by donating computer hardware and software to the schools. For example, at a recent National Computer Conference, Apple Computer, Inc. launched a program to donate computers to more than 9,000 public and private schools throughout California. The program, called Kids Can't Wait, is estimated to cost about $21 million. This program followed passage of California state legislation making donations of computer equipment eligible for tax credits of up to 25% of its market value. Clearly, businesses will be more encouraged to make such donations if they receive some tax benefits, due to federal or state policies.

Internships/training. For both students and teachers, business and industry can provide additional learning environments. Through internships, students and teachers can learn the practical implications of their educational experiences.

## Summary of Recommendations

In summary, we have identified the following 10 major recommendations for improving the quality of courseware to teach reading, writing, and communication skills.

1. The federal government should develop a leadership position, in order to focus attention of software developers on the school market and in order to ensure excellence and equity in the use of computers in education.

2. The federal government should fund projects aimed at the development of high quality software, based on careful research. Development costs are high, and the school market is otherwise not sufficiently large to attract the needed developers' efforts.

3. The federal government should support efforts in the areas of developing (a) standardization in hardware and in software languages, (b) software evaluation standards, and (c) regional resource centers, in order to increase schools' abilities to select and acquire high quality software.

4. The federal and state governments should support hardware acquisition by schools, such as through tax credits to manufacturers. The size of the hardware base is critical for the development of high quality software.

5. States should establish standards for computer literacy in the preparation of new teachers, in order to alleviate the need for massive in-service training.

6. Schools should develop long-term plans for use of computers, including hardware and software acquisition procedures and in-service training at three different levels of expertise: first-time users, continuing users, and potential developers.

7. Local teams of teachers, programmers, publishers, and researchers should be formed in order to generate high quality, relevant courseware. The responsibility for forming these teams might lie with any of the constituents, but the value of forming the teams is unquestionable.

8. Courseware developers should make use of research on reading and writing in order to generate exercises and activities that require practice of appropriate skills; and they should describe courseware in terms familiar to teachers.

9. Schools should require that courseware be of high quality, with (a) documentation that includes suggested uses, (b) mechanisms for adaptation to local classroom variations, and (c) accurate use of English.

10. Schools should take strong positions opposed to the use of illegally copied software by students and staff. Software piracy is viewed by publishers as a significant deterrent to investment in the development of high quality courseware.

REFERENCES

Adams, M. J. Failures to comprehend and levels of processing in reading. In R. J. Spiro, B. C. Bruce, & W. F. Brewer (Eds.), Theoretical issues in reading comprehension. Hillsdale, NJ: Lawrence Erlbaum Associates, 1980.

Adams, S. Say yoho. Softside, February 1981, 3(5), 12.

Allen, H. W. The art of science: Professionalism in CBI (Or standing on each other's feet). Journal of Computer-Based Instruction, 1978, 4, 59-63.

Applebee, A. N. Looking at writing. Educational Leadership, March 1981, 38 (6), 458-462.

Anadam, K., & Kelley, J. T. Teaching and technology: Closing the gap. T.H.E. Journal, November 1982, 10(2), 84-90.

Barrett, T. Taxonomy of reading comprehension. In R. Smith, & T. C. Barrett, Teaching reading in the middle grades. Reading, MA: Addison-Wesley, 1976.

Bayer, B. D. No truce in sight in copy-protection war. InfoWorld, May 25, 1981, 3(10), 17-19.

Becker, S. A. Legal protection for computer hardware and software. BYTE magazine, May 1981, 6(5), 140-145.

Billings, K. Computers (Chapter 3). In Best bet. New York: Columbia University, 1981.

Block, K. K. Cognitive theory, CAI, and spelling improvement. Journal of Computer-Based Instruction, 1979, 5, 86-95.

Bobrow, D. G., & Norman, D. A. Some principles of memory schemata. In D. G. Bobrow & A. M. Collins (Eds.), Representation and understanding: Studies in cognitive science. New York: Academic Press, 1975.

Boyd, R. Attacking the in-service education problem. The Computing Teacher, March 1981, 8(7), 50-51.

Bradley, V. N. Improving students' writing with microcomputers. Language Arts, 1982, 59, 732-743.

Bridwell, L. S. Revising strategies in twelfth grade student's transactional writing. Research in the teaching of English, October 1980, 74(3), 197-222.

Bridwell, L., Nancarrow, P. R., & Ross, D. The writing process and the writing machine: Current research on word processors relevant to the teaching of composition. In R. Beach & L. Bridwell (Eds.), New directions in composition research. New York: Guilford Press, forthcoming.

Britton, J. The composing processes and the function of writing. In C. R. Cooper & L. Odell (Eds.), Research on composing: Points of departure. Urbana, IL: National Council of Teachers of English, 1978.

Britton, J., Burgess, T., Martin, N., McLeod, A., & Rosen, H. The development of writing abilities (11-18). London: Macmillan Education, 1975.

Brown, A. Metacognitive development and reading. In R. J. Spiro, B. C. Bruce, & W. F. Brewer (Eds.), Theoretical issues in reading comprehension. Hillsdale, NJ: Lawrence Erlbaum Associates, 1980.

Brown, J. S., & VanLehn, K. Repair theory: A generative theory of bugs in procedural skills. Palo Alto, CA: Xerox, Palo Alto Research Center, 1980.

Bruner, J. S. On knowing: Essays on the left hand. New York: Atheneum, 1965.

Burke, L. A grammar of motives. New York: George Braziller, 1955.

Burns, H. Computer-assited pre-writing activities: Harmonics for invention. In J. Lawlor (Ed.), Computers in composition instruction. Los Alamitos, CA: SWRL Educational Research and Development, 1982.

Burton, R. R. Diagnosing bugs in a simple procedural skill. Palo Alto, CA: Xerox, Palo Alto Research Center, 1981.

Caldwell, R. M. Guidelines for developing basic skills instructional materials for use with microcomputer technology. Educational Technology, October 1980, 10(10), 7-12.

Caldwell, R. M., & Rizza, P. J. A computer-based system of reading instruction for adult non-readers. AEDS Journal, Summer 1979.

California Testing Bureau. CTBS Forms U & B Test Coordinator's Handbook: Preliminary Edition. Monterey, CA: CTB/McGraw-Hill, 1982.

Caplan, N. Social research and national policy: What gets used, by whom, for what purposes, and with what effects. Evaluation Studies Review Annual, 1977, 2, 351-359.

Carroll, J. B., & Freedle, R. O. (Eds.). Language comprehension and the acquisition of knowledge. Washington, DC: V. H. Winston & Sons, 1972

Cohen, V. L. B. Evaluating instructional software for the microcomputer. Paper presented at the AERA Convention, New York, 1982.

Collins, A. Teaching reading and writing with personal computers. Cambridge, MA: Bolt Beranek and Newman, Inc. 1981.

Collins, A., Bruce, B. C., & Rubin, A. Microcomputer-based writing activities for the upper elementary grades. In the Proceedings of the Fourth International Congress and Exposition of the Society for Applied Learning Technology, Orlando, Florida, February 1982.

Cooper, C. R., & Odell, L. (Eds.). Research on composing: Points of departure. Urbana, IL: National Council of Teachers of English, 1978.

Cronnell, B., & Michael, J. (Eds.). Writing: Policies, problems, and possibilities. Los Alamitos, CA: SWRL Educational Research and Development, 1982.

Doyle, M. Palo Alto Weekly, November 17th, 1982, pp. 15-17.

Easterling, B. The issue is really consumers' rights, not piracy. InfoWorld, March 22, 4(11), 1982, 56-57.

Edlin, J. Kinky pirates stir sea of controversy with teaspoons, not shovels. InfoWorld, January 11, 1982, 4(1), 24-25.

Educational software directory, Apple II edition. Manchaca, TX: Sterling Swift, 1981.

Edwards, K. Problems related to the teaching of writing in the public schools. In B. Cronnell & J. Michael (Eds.), Writing: Policies, problems, and possibilities. Los Alamitos, CA: SWRL Educational Research and Development, 1982.

Edwards, M. H. Selling thy software. Software News, January 1983.

Elman, A. The courseware dilemma. NICE Newsletter, Fall 1982, 2(1), 1.

Emig, J. Hand, eye, brain: Some "basics" in the writing process. In C. R. Cooper and L. Odell (Eds.), Research on composing: Points of departure. Urbana, IL: National Council of Teachers of English, 1978.

Fair is fair. InfoWorld, June 8, 1981, 3(11), 30.

Finkel, L. Computer using educators position paper on commercial software pricing policies. COMPUTE!, June 1981, 3(6), 16-18.

Flower, L., & Hayes, J. R. The cognitition of discovery: Defining a rhetorical problem. College Composition and Communication, February 1980, 31(1), 21-32.

Fraser, R., Wells, C., & Burkill, S. Designing material for the microcomputer and teacher partnership in the classroom. T.H.E. Journal, September 1982, 9(1), 55-57.

Frederiksen, J. R. A componential theory of reading skills and their interactions. In R. J. Sternberg (Ed.), Advances in the psychology of human intelligence. Hillsdale, NJ: Lawrence Erlbaum Associates, 1982.

Freedle R. O. (Ed.). New directions in discourse processing (Vol. II); Advances in discourse processes. Norwood, NJ: Ablex Publishing Corporation, 1979.

Freiberger, P. Pirates bedevil angry and frustrated software vendors. InfoWorld, March 22, 1982, 4(11), 31-38.

French, J. W. The description of aptitude and achievement tests in terms of rotated factors. Psychometric Monographs, No. 5.

Gagné, R. M. The conditions of learning. New York: Holt, Rinehart, and Winston, 1965.

Gagné, R. M., Wager, W., & Rojas, A. Planning and authoring computer-assisted instruction lessons. Educational Technology, September 1981, 11(9), 1-26.

Gerard, R. W. Shaping the mind: Computers in education. In R. C. Atkinson & H. C. Wilson (Eds.), Computer-assisted instruction. New York: Academic Press, 1969, 15-39.

Gibson, E. J., & Levin, H. The psychology of reading. Cambridge, MA: The MIT Press, 1976.

Gilder, J. H. Software piracy: The problem is getting worse. Personal Computing, April 1981, 5(4), 6.

Glaser, R. Components of a psychology of instruction: Toward a science of design. Review of Educational Research, 1976, 46(1), 1-24.

Goodman, Y., & Burke, C. L. Reading miscue inventory: Procedure for diagnosis and education. New York: Macmillan, 1972.

Gordon, W. J. J. Synectics: The development of creative capacity. New York: Collier, 1961.

Gore, M. R. The wonderful world of copyright confusion. Creative Computing, August 1980, 6(8), 140-141.

Gould, J. D. Experiments on composing letters: Some facts, some myths, and some observations. In L. W. Gregg & E. R. Steinberg (Eds.), Cognitive processes in writing. Hillsdale, NJ: Lawrence Erlbaum Associates, 1980.

Greeno, J. G. Cognitive objectives of instruction: Theory of knowledge for solving problems and answering questions. In D. Klahr (Ed.), Cognition and instruction. New York: Halsted Press, 1976.

Gregg, L. W., & Steinberg, E. R. (Eds.). Cognitive processes in writing. Hillsdale, NJ: Lawrence Erlbaum Associates, 1980.

Grossnickle, D. R., Laird, B. A., Cutter, T. W., & Tefft, J. A. Profile of change in education: A high school faculty adopts/rejects microcomputers. Educational Technology, June 1982, 12(6), 17-19.

Guilford, J. P. The nature of human intelligence. New York: McGraw-Hill, 1967.

Hartman, J. A systematic approach to the design of computer assisted instruction materials. T.H.E. Journal, February 1981, 8(2), 43-45.

Hayes, J. R., & Flower, L. S. Identifying the organization of writing processes. In L. Gregg & E. Steinberg (Eds.), Cognitive processes in writing: An interdisciplinary approach. Hillsdale, NJ: Lawrence Erlbaum Associates, 1980.

Hayman, M. Software protection in the United Kingdom. BYTE magazine, October 1981, 6(10), 126-139.

Heilman, A. W. Principles and practice of teaching reading. Columbus, OH: Charles E. Merrill, 1967.

Holmes, G. Computer-assisted instruction: A discussion of some of the issues for would-be implementors. Educational Technology, September 1982, 12(9), 7-13.

Holzman, T. G. Process training as a test of computer simulation theory. Unpublished master's thesis. Pittsburgh, PA: University of Pittsburgh, 1975.

Hoover, T., & Gould, S. The pirating of computer programs: A survey of software producers. Educational Technology, October 1982, 22(10), 23-26.

Humes, A. Research on the composing process. In B. Cronnell & J. Michael (Eds.), Writing: Policies, problems, and possibilities. Los Alamitos, CA: SWRL Educational Research and Development, 1982.

Isaacson, D. What makes educational software great. School CourseWare Journal, 1982.

Jenkins, J. R., & Pany, D. Teaching reading comprehension in the middle grades. In R. J. Spiro, B. C. Bruce, & W. F. Brewer (Eds.), Theoretical issues in reading comprehension. Hillsdale, NJ: Lawrence Erlbaum Associates, 1980.

Jenkins, J. R., & Pany, D. Instructional variables in reading comprehension. In J. T. Guthrie (Ed.), Comprehension and teaching: Research reviews. Newark, DE: International Reading Association, 1981.

Kock, C., & Brazil, J. M. Strategies for teaching the composition process. Urbana, IL: National Council of Teachers of English, 1978.

Koestler, A. The act of creation. London: Hutchinson, 1964.

Langer, J. A. Computer technology and reading instruction: Perspectives and directions. University of California, Berkeley, 1982. (ERIC Document Reproduction Service No. ED 214131)

Langer, J. A. Reading, writing and the use of computers in schools. Paper prepared for the National Institution of Education as part of a task force monograph on "The Use of Computers to Teach Reading and Writing", 1981.

Larson, R. L. Selected bibliography of research and writing about the teaching of composition. College Composition and Communication, 1978, 29, 181-194.

Larson, R. L. Selected bibliography of research and writing about the teaching of composition. College Composition and Communication, 1979, 30, 196-213.

Lathrop, A. SOFTSWAP. The Computing Teacher, April 1982, 9(8), 16-31.

Lau, C. The software writer's market. Duxbury, MA: Kern Publications, 1982.

Leichtman, K. Group formed to fight software bugs. 80 Microcomputing, March 1982, 380.

Leonard, G. Education and ecstasy. New York: Dell, 1969.

Letellier, L. T. Copying software: Crime in the classroom? Electronic Learning, January 1982, 1(3), 42-51.

Lock, R. Backing-up: Software copyrights revisited. COMPUTE!, July 1981, 3(7), 4.

Lock, R. Software copying revisited, or who's paying the bills? COMPUTE!, 3(3), March 1981, 4-9.

MacLennan, E. The Computer Software Protection Act. Interface Age, September 1980, 5(9), 17. (a)

MacLennan, E. Legal liability and protection for software companies. Interface Age, November 1980, 5(11), 17-18. (b)

Malone, T. W. Toward a theory of intrinsically motivating instruction. Cognitive Science, 1981, 4, 333-369.

Manufacturer offers reward for names of software pirates. Electronic Learning, March 1982, 1(14), 12.

Marshall, D. Software Copyright Law. 80 Microcomputing, November 1980, 48-55.

Mason, G. E. Computerized reading instruction: A review. Educational Technology, October 1980, 10(10), 18-22.

McLaughlin, D. H. Coordination and evaluation of videodisc and microcomputer programs across 45 sites. Paper presented at the Fourth International Learning Technology Congress and Exposition sponsored by the Society for Applied Learning Technology, Orlando, FL, February 1982.

McMillan, N. Personal communication, November 17, 1982.

MICROSIFT: An implementation plan for resources in computer education. Portland, OR: Northwest Regional Educational Laboratory, 1981.

Moffett, J., & Wagner, B. M., Student-centered language arts and reading, k-13: A handbook for teacher (2nd Ed.). Boston: Houghton Mifflin Company, 1976.

Morgan, C. How can we stop software piracy? BYTE Magazine, May 1981, 6(5), 6-10.

Morris, W. (Ed.). The American heritage dictionary of the English language. New York: American Heritage Publishing Company, 1969.

Morrissey, W. J., Jr. Overcoming educators' fears about classroom computers. Computing Teacher, October 1980, 8(2), 50-51.

Nansen, C. Teaching computer use--not programming: An outline for a five-session course. Electronic Learning, November 1982, 2(3), 24, 31.

National Assessment of Educational Progress. Reading, thinking, and writing: Results from the 1979-80 national assessment of reading and literature. Denver, CO: Education Commission of the States, 1981.

Nold, E. Revising: Toward a theory. Paper presented at Conference on College Composition and Communciation, Minneapolis, Minnesota, March 1979.

Norman, D. A. What goes on in the mind of the learner. In W. J. McKeachie (Ed.), New directions for teaching and learning. New York: Jossey-Bass, 1980.

Office of Technology Assessment. Technology, innovation, and regional economic development. Washington, D.C.: U.S. Government Printing Office, 1983.

Olmstead, R. The legal protection of computer programs. Creative Computing, December 1980, 6(12), 114-116.

Pearson, P. D., & Johnson, D. D. Teaching reading comprehension. New York: Holt, Rinehart, and Winston, 1978.

Petruso, S. A commitment to computer education: Introducing computers into a district. T.H.E. Journal, November 1981, 8(6), 53.

Pogrow, S. Technological changes: Policy implications for funding and delivering educational service in the 80s. University of Arizona. Prepared for the National Institute of Education School Finance Project, 1982.

Purdy, L. Community college instructors and use of new media: Why some do and others don't. Educational Technology, March 1975, 5(3), 9-12.

Ragosta, M., Holland, P. W., & Jamison, D. T. Computer-assisted instruction and compensatory education; The ETS/LAUDS STUDY (Project Report No. 20). Princeton: Educational Testing Service, 1982.

Rapaport, P., & Savard, W. G. Computer-assisted instruction. Reprinted in Microcomputers in today's schools: A conference. An administrator's handbook. Portland, OR: Northwest Regional Educational Laboratory, 1981.

Rawitsch, D. G. Teaching educators about computing: A different ball game. The Computing Teacher, December 1981, 9(4), 27-32.

Roberts, W. K. Preparing instructional objectives: Usefulness revisited. Educational Technology, July 1982, 12(7), 15-19.

Rockart, J. F., & Morton, S. Computers and the learning process in higher education. New York: McGraw-Hill, 1975.

Rohman, D. G. Pre-writing: The stage of discovery in the writing process. College Composition and Communication, 1965, 19, 106-112.

Rose, S. N. Barriers to the use of educational technologies and recommendations to promote and increase their use. Educational Technology, December 1982, 12(12), 12-15.

Rosenshine, B. V. Skill hierarchies in reading comprehension. In R. J. Spiro, B. C. Bruce, & W. F. Brewer (Eds.), Theoretical issues in reading comprehension. Hillsdale, NJ: Lawrence Erlbaum Associates, 1980, 535-554.

Rumelhart, D. E. Notes on a schema for stories. In D. G. Bobrow & A. M. Collins (Eds.), Representation and understanding: Studies in cognitive science. New York: Academic Press, 1975.

Rumelhart, D. E. Schemata: The building blocks of cognition. In J. T. Guthrie (Ed.), Comprehension and teaching: Research reviews. Newark, DE: International Reading Association, 1981, 3-36.

Russ-Eft, D. F., McLaughlin, D. H., & Elman, A. Issues for the development of reading and writing software. Palo Alto, CA: American Institutes for Research, 1983.

Schank, R. C., & Abelson, R. P. Scripts, plans, goals, and understanding. Hillsdale, NJ: Lawrence Erlbaum Associates, 1977.

Scope and sequence: The Addison-Wesley reading program. Menlo Park, CA: Addison-Wesley, 1982.

Sheingold, K., Kane, J., Endreweit, M., & Billings, K. Study of issues related to implementation of computer technology in schools. New York: Bank Street College of Education, 1981.

Sherman, M. T. Computers in education: A report - recommendations and resources. Concord, MA: Bates Publishing, 1983.

Shostak, R. F. Computers and teaching English: Bits 'n' pieces. The Computing Teacher, 1980-1982.

Simon, H. A., & Chase, W. G. Skill in chess. American Scientist, 1973, 61, 394-403.

Simon, H. A. Alternative uses of phonenemic information in spelling. Review of Educational research, 1973, 43, 115-137.

Slesnick, T. Teacher inservice in computer education. Educational Computer Magazine, 1983, 3(2), 16-8.

Smith, J. M. A technology of reading and writing (Vol. 4): Designing instructional tasks. New York: Academic Press, 1978.

Software directory: A guide to 200 producers of educational programs. Electronic Learning, May/June 1982, 1(5), 1A-11A.

Software protection: The law. Personal Computing, February 1982, 6(2), 16, 98.

Spero, S. Let them have micros: An approach to computer literacy for teachers. T.H.E. Journal, 1982, 10(1), 127-129.

Steffin, S. A software publisher's position on software pricing and service policies. COMPUTE!, October 1981, 3(10), 36-38.

Sturdevant, R. Microcomputers and copyright in education. Phi Delta Kappa, 1982.

Thompson, B. J. Computers in reading: A review of applications and implications. Educational Technology, August 1980, 10(8), 38-41.

Townsend, B., & Hale, D. Coping strategies for resistance to microcomputers. T.H.E. Journal, November 1981, 8(6), 49-52.

Toffler, A. The third wave. New York: Bantam Books, Inc., 1980.

Thurstone, L. L. Primary mental abilities. Psychometric Monographs, 1938, No. 1.

Thurstone, L. L., & Thurstone, T. G. Factorial studies of intelligence. Psychometric Monographs, 1941, No. 2.

Tyndall, J. Faraday as a discoverer. London: Longmans, Green, 1868.

VanLehn, K., & Brown, J. S. Planning nets: A representation for formalizing analogies and semantic models of procedural skills. In R. W. Snow, P. A. Frederico, & W. E. Montague (Eds.), Aptitude learning and instruction (Vol 2): Cognitive process analyses of learning and problem solving. Hillsdale, NJ: Lawrence Erlbaum Associates, 1980.

Wagner, W. J. Author languages: Instruction without programming. Classroom Computer News, July 1981, 1(6), 42-43.

Wall, S. M., & Taylor, N. E. Using interactive computer programs in teaching higher conceptual skills: An approach to instruction in writing. Educational Technology, February 1982, 12(2), 13-17.

Watson, J. D. *The double helix*. New York: Signet Books, 1968.

Weiner, C. The hidden cost of copyright infringement. *Classroom Computer News*, July 1981, *1*(6), 38.

Wertheimer, M. (Ed.). *Productive thinking*. New York: Harper and Row, 1959.

Wholey, J. S. *Evaluation: Promise and performance*. Washington, DC: The Urban Institute, 1979.

Wilson, K. G. English teachers: Keys to computer literacy. *English Journal*, September 1981, 50-53.

Wollman, J. Are you breaking the law? *Popular Computing*, 1982, *1*(6), 99-106.

Woods, W. A. Multiple theory formation in speech and reading. In R. J. Spiro, B. C. Bruce, & W. F. Brewer (Eds.), *Theoretical isssues in reading comprehension*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1980.

Young, R. E. Paradigms and problems: Needed research in rhetorical invention. In C. R. Cooper & L. Odell (Eds.), *Research on composing: Points of departure*. Urbana, IL: National Council of Teachers of English, 1978.

Young, R. E., Becker, A. L., & Pike, K. L. *Rhetoric: Discovery and change*. New York: Harcourt, Brace, & World, 1970.

## APPENDIX A

### Idea Work Group Participants

Adams, Bob
EISI-Educational Instructional
Systems, Inc.
Los Altos, California

Allen, Ann
Richmond Public Schools
Richmond, Virginia

Allen, Karen
Lesley College
Cambridge, Massachusetts

Arkin, Jaime
Richmond Public Schools
Richmond, Virginia

Aylsworth, Gayle
Glen Grove Public Schools
Glenview, Illinois

Bagan, Beverly
Richmond Public Schools
Richmond, Virginia

Baker, Lois
Creek Valley Elementary School
Edina, Minnesota

Banner, Ronnie
Hurst-Euless-Bedford
Independent School District
Bedford, Texas

Barlow, Louise
Society for Visual Education, Inc.
Chicago, Illinois

Becht, Paul
P. K. Yonge Laboratory School
Gainesville, Florida

Becht, Sally
P. K. Yonge Laboratory School
Gainesville, Florida

Bergen, Victoria
Texas Education Agency
Austin, Texas

Birhle, Karen
Osseo Public Schools
Osseo, Minnesota

Bjork, William
Richmond Public Schools
Richmond, Virginia

Blythe, Barbara
Bell and Howell
Chicago, Illinois

Branch, Muriel
Richmond Public Schools
Richmond, Virginia

Bridwell Lillian
University of Minnesota
Minneapolis, Minnesota

Brudenell, Gerry
Florida State University
Tallahassee, Florida

Butler-Nalin, Kay
Stanford University
Stanford, California

Capes, Bruce
Heath Zenith Educational Systems
Benton Harbor, Michigan

Chipman, Susan
National Institute of Education
Washington, D.C.

Choate, Jon
Groton School
Groton, Massachusetts

Clark, Bob
Florida State University
Tallahassee, Florida

Collins, Allan
Bolt, Beranek and Newman
Cambridge, Massachusetts

Colvin, Carol
Hurst-Euless-Bedford
Independent School District
Bedford, Texas

Cory, Beverly
Dale Seymour Publications
Menlo Park, California

Cubillos, Enrique
University of Iowa
Iowa City, Iowa

Dawson, George
Florida State University
Tallahassee, Florida

Dawson, Joel
Astoria Park School
Tallahassee, Florida

Denton, William
Dallas Independent School District
Dallas, Texas

Dibblee, Larry
Institute of Education Research
Batavia, Illinois

Ditty, Rudolph
Educational Activities
Ocala, Florida

Doig, Steve
Miami Herald
Tallahassee, Florida

Doyle, Brian
Osseo Public Schools
Osseo, Minnesota

Eckert, Lyn
Education Service Center Region 10
Richardson, Texas

Ehr, Carolyn
University of Florida
Gainesville, Florida

Emerick, Blanche
Control Data Corporation
Minneapolis, Minnesota

Esbensen, Thorwald
Micro-Ed Industries
Minneapolis, Minnesota

Filliman, Paula
Glenview Public Schools
Glenview, Illinois

Fortune, Ron
Computer Curriculum Corporation
Palo Alto, California

Frantz, Ed
Micro Power and Light
Dallas, Texas

Frederiksen, John
Bolt, Beranek and Newman
Cambridge, Massachusetts

Fultz, Lionel
IBM Corporation
Rockville, Maryland

Gagné, Ellen
Stanford University
Stanford, California

Gayles, Ruth
Richmond Public Schools
Richmond, Virginia

Haggard, Gerry
Plano School District
Plano, Texas

Haynes, Amos
Convergent Systems, Inc.
St. Paul, Minnesota

Head Douglas
Edina Public Schools
Edina, Minnesota

Hegarty, Carol
Learning Line
Berkeley, California

Hoetker, Jim
Florida State University
Tallahassee, Florida

Holland, Fred
Radio Shack Education Division
Fort Worth, Texas

Honaker, Linda
Hurst-Euless-Bedford
Independent School District
Bedford, Texas

Irons, Dennis
Texas Christian University
Fort Worth, Texas

Johnson, Jan
Education Media Services
Minneapolis, Minnesota

Johnson, Tom
Plano School District
Plano, Texas

Kantrov, Ilene
Education Development Center
Newton, Massachusetts

Keller, Barbara
Corte Madera Elementary School
Portola Valley, California

Kosloski, Coleen
Minneapolis Public Schools
Minneapolis, Minnesota

Kozen, Nancy
Minnesota Educational Computing Consortium
St. Paul, Minnesota

Latham, Marilyn
Scott, Foresman & Co.
Burlingame, California

Latzke, Laura
St. Paul Lutheran School
Melrose Park, Illinois

LeFrenz, Dale
Scott-Foresman and Company
Glenview, Illinois

Leonard, Jenelle
Northern Virginia Community College
Washington, D.C.

Lepper, Mark
Stanford University
Stanford, California

Levinsky, Jeff
Interactive Sciences
Palo Alto, California

Lobello, Anthony, J.
Electronic Communications, Inc.
Tallahassee, Florida

Lobello, Sharon T.
Electronic Education
Tallahassee, Florida

Lowd, Beth
English Teacher
Newton, Massachussets

Maddox, Rachel
Richmond Public Schools
Richmond, Virginia

Madison, John
Gallaudet College
Washington, D.C.

McCauley, James
Santa Clara County
Office of Education
San Jose, California

McClees, J. Sheridan
Terrapin Inc.
Cambridge, Massachusetts

McGee, Max
Darien District No. 61
Fairview, Illinois

McMillan, Nina
Addison-Wesley Publishing Co.
Menlo Park, California

Meyer, Bonnie
Arizona State University
Lake Forest, Illinois

Moffett, Jim
Gainesville Schools
Gainesville, Florida

Mumford, Jenise
Dallas Independent School District
Dallas, Texas

Murphy, Susanne
Eastern Jr. High School
Riverside, Connecticut

Murray, Betty
Lexington Public Schools
Woburn, Massachusetts

Naiman, Adeline
HRM Software - Human Relations Media
Cambridge, Massachusetts

O'Toole, Anne
Richmond Public Schools
Richmond, Virginia

Orasanu, Judith
U. S. Army Research Institute
Alexandria, Virginia

Palmer, Nancy
Palo Alto Unified School District
Palo Alto, California

Parker, Ralph
Learning Line
Menlo Park, California

Pasta, Betty
University of the District
of Columbia
Washington, D.C.

Peck, Lorna
Hewlett Packard
Palo Alto, California

Perl, Teri
The Learning Co.
Portola Valley, California

Peters, David
Electronic Courseware Systems
Champaign, Illinois

Piestrup, Ann
The Learning Co.
Portola Valley, California

Powell, Francis
Langdon School
Washington, D.C.

Pretlow, Dolores
Richmond Public Schools
Richmond, Virginia

Pugh, Richard
Cupertino Union School District
Cupertino, California

Redish, Ginny
American Institutes for Research
Washington, D.C.

Reed, Margaret
Minneapolis Public Schools
Minneapolis, Minnesota

Reese, Phil
University of California
Berkeley, California

Roberts, Linda
U. S. Department of Education
Washington, D.C.

Robinett, Warren
The Learning Co.
Portola Valley, California

Rogers, Ralph
Atari
Sunnyvale, California

Ross Don
University of Minnesota
Minneapolis, Minnesota

Sack, Kay
Minneapolis Public Schools
Minneapolis, Minnesota

Saecker, Peter
Science Research Associates
Chicago, Illinois

Schragg, Bill
Osseo Public Schools
Osseo, Minnesota

Simmons, Doug
Simpac Education Systems
Gainesville, Florida

Snider, Robert
National Education Association
Washington, D.C.

Sutherland, Hilda
Commonweath of Virginia
Department of Education
Richmond, Virginia

Stauffer, Carolyn
Apple Computers
San Jose, California

Stolurow, Lawrence
University of Iowa
Iowa City, Iowa

Strassberg, Ruth
Peirce School
Arlington, Massachusetts

Sulzby, Elizabeth
Northwestern University
Evanston, Illinois

Swift, Sterling
Sterling Swift Publishing Co.
Austin, Texas

Targ, Joan
Palo Alto Unified School District
Palo Alto, California

Thomas, Bill
Thomas Jefferson Elementary School
Falls Church, Virginia

Thompson, Charles
Education Development Center
Newton, Massahcusetts

Tingey, Richard
Sequoia Union High School District
Belmont, California

Tinker, Robert
Technical Education Research Centers
Cambridge, Massachusetts

Turrentine, Bonnie
J. L. Hammet Co., Inc.
Braintree, Massachusetts

Tweedt, Joy
Dallas Independent School District
Dallas, Texas

Ueltchi, Barbara
American Institutes for Research
Washington, D.C.

Valdez, Gil
Minnesota State Department
of Education
St. Paul, Minnesota

Wallat, Cynthia
Florida State University
Tallahassee, Florida

Webster, William
Dallas Independent School District

Winkler, Victoria
University of Minnesota
Minneapolis, Minnesota

Zammit, Joanne
Eastern Jr. High School
Riverside, Connecticut

## LIST OF IDEA WORK GROUP PARTICIPANTS BY LOCATION

Palo Alto, California

Bob Adams
Kay Butler-Nalin
Beverly Cory
Ron Fortune
Ellen Gagné
Carol Hegarty
Barbara Keller
Marilyn Latham
Mark Lepper
Jeff Levinsky
James McCauley
Nina McMillan
Nancy Palmer
Ralph Parker
Lorna Peck
Teri Perl
Ann Piestrup
Richard Pugh
Phil Reese
Ralph Rogers
Carolyn Stauffer
Joan Targ
Richard Tingey
Warren Robinett

Gainesville, Florida

Paul Becht
Sally Becht
Rudolph Ditty
Carolyn Ehr
Jim Moffet
Doug Simmons

Tallahassee, Florida

Gerry Brudenell
Bob Clark
George Dawson
Joel Dawusopn
Steve Doig
Jim Hoetker
Sharon Lobello
Anthony Lobello
Cynthia Wallat

Glenview, Illinois

Gayle Aylsworth
Barbara Blythe
Bruce Capes
Paula Filliman
Max McGee
Bonnie Meyer
Peter Saecker
Elizabeth Sulzby

Melrose Park, Illinois

Louise Barlow
Enrique Cubillos
Larry Dibblee
Laura Latzke
Dale LeFrenz
David Peters
Lawrence Stolurow

Bedford, Texas

Ronnie Banner
Carol Colvin
Gerry Haggard
Fred Holland
Linda Honaker
Dennis Irons
Tom Johnson

Dallas, Texas

Victoria Bergen
William Denton
Lyn Eckert
Ed Frantz
Jenise Mumford
Sterling Swift
Joy Tweedt

Minneapolis, Minnesota

Lois Baker
Karen Birhle
Lillian Bridwell
Brian Doyle
Blanche Emerick
Thorwald Esbensen
Amos Haynes
Douglas Head
Jan Johnson
Coleen Kosloski
Nancy Kozen
Margaret Reed
Don Ross
Kay Sack
Bill Schragg
Gil Valdez
Victoria Winkler

Cambridge, Massachusetts

Karen Allen
'on Choate
llan Collins
John Frederiksen
Ilene Kantrov
Beth Lowd
J. Sheridan McClees
Susanne Murphy
Betty Murray
Adeline Naiman
Ruth Strassberg
Bonnie Turrentine
Charles Thompson
Robert Tinker
Joanne Zammit

Richmond, Virginia

Ann Allen
Jamie Arkin
Beverly Bagan
William Bjork
Muriel Branch
Ruth Gayles
Rachel Maddox
Ann O'Toole
Dolores Pretlow
Hilda Sutherland

Washington, D.C.

Susan Chipman
Jenelle Leonard
Linda Roberts
Bill Thomas
Robert Snider
Barbara Ueltchi
Lionel Fultz
John Madison
Judith Orasanu
Betty Pasta
Francis Powell
Ginny Redish
Barbara Ueltchi

APPENDIX B

## Listing of Courseware by Language Skills Area

### Language Skills--Reading Readiness

| Relevant Software[1] | Publisher[2] |
|---|---|
| Letterman (A, IBM, C, VIC20) | Behavioral Engineering |
| Alphakey (A32) | Bertamax |
| Alpha Letter Drop (T) | Bertamax |
| Match It (T, AT) | Bertamax (CT--Vol.10, No.9, May 1983) |
| Letters and First Words (A48) | C & C Software |
| Hangbug (C64, VIC20, PETO | Comm-Data Systems |
| Toddler Tutor & Tutor Alphabet (C64, VIC20, PET) | Comm-Data Systems |
| Let's Alphabetize (T) | Computer Learning Center 4 Children |
| Reading Readiness (A48, T) | Computer Courseware Services |
| Alphabetize (T) | Course Ware |
| Reverse | Course Ware |
| Letter | Dr. Daley |
| Big Letter | Dr. Daley |
| Hodge Podge | Dynacomp, Inc. (CT--Vol.10, No.9, May 1983) |
| Compu-Read: Character Recognition | Edu-Ware Services, Inc. |
| My First Alphabet | Fernando Herrera |
| Alphabetize (A48) | Fullmer & Assoc. |
| Letter Recognition (A48) | Hartley Software |
| Alphabetizing (A48) | Learning Well |

---

[1]The microcomputers compatible with the software, if known, are identified by the following abbreviations within parentheses following the name of the software. Some of the Apple software also specifies the required memory.

| | |
|---|---|
| A | Apple |
| AT | Atari |
| C | Commodore (model not specified) |
| Pet | Commodore Pet |
| VIC20 | Commodore VIC20 |
| T | TRS-80 |
| CBM | Commodore CBM |
| IBM | IBM |

[2]Known journal evaluations of the software are identified within parentheses following the name of the publisher using the following abbreviations

| | |
|---|---|
| CC | Creative Computer |
| CT | Classroom Teacher |
| EC | Educational Computer |
| ET | Educational Technology |
| SMW | School Microware Reviews |
| MJ | Macul Journal |
| EL | Electronic Learning |
| CCN | Classroom Computer News |
| JEM | JEM Reference Manual |
| EPIE | EPIE Reports |
| EE | Electronic Education |

| | |
|---|---|
| Alphabetizing (T) | Little Bee |
| Missing Letter (T) | Little Bee |
| Upper/Lowercase Matching (T) | Little Bee |
| ABC Sequence (T) | Little Bee |
| ABC (A32) | Merry Bee |
| Alphabetize (A32) | Merry Bee |
| Earyly Words (A32) | Merry Bee |
| Letter Recognition and Alphabetization (A48) | Milliken Publishing |
| Apple Flash | OMNICO |
| Preschool, IQ Builder | Program Design, Inc. |
| My ABC's | Softagon |
| Now I can Rhyme | Softagon |
| The Reading Machine (A) | SouthWest EdPsych Services, Inc. |
| Alphabet | Steketee Educational Software |
| Hard and Soft C | Micro-Ed |
| Hard and Soft G | Micro-Ed |
| Make a Sentence (PET) | Micro-Ed |
| Matching Small with Capital Letters (V20) | Micro-Ed |
| Matching Capital Letters (PET) | Micro-Ed |
| Which Letter Comes Next (PET) | Micro-Ed |
| ABC | MECC |
| Alphabet | MECC |
| Sequence and Alphabetizing (A, AT, T) | Random House |
| Early Reading (TI) | Scott Foresman/Texas Inst. (EC--9-10/81) |
| The Reading Connection | Scott Foresman |
| Juggles Rainbow (A48, T, AT) | The Learning Company (EPIE & Consumers Union, CT--Vol.10, No.9, May 1983) |
| Rocky's Boots | The Learning Company |
| Bumble Plot | The Learning Company |
| Gertrude's Puzzles | The Learning Company |
| Secrets | The Learning Compnay |
| Moptown | The Learning Company |
| Letters and Numbers Program (CBM, PET) | Teaching Tools |
| Alphabet Antics (see Softswap) | |
| Big Letter (check Softswap) | |

## Reading Skills--Phonetic Analysis

| Relevant Software | Publisher |
|---|---|
| Vowel Kong (C64, VIC20, PET) | Comm-Data Systems |
| Rhyme Time (T) | Computer Learning Center 4 Children |
| Scramble | Course Ware |
| Using Phonics in Context | Educational Activities, Inc. |
| Anagrams (LAS-2) (T) | Educational Media Associates |
| Short Vowel Exercise (A32) | Edutek |
| Phonic Series 2 | Msss D. |

| | |
|---|---|
| Consonants (A48) | Hartley |
| Vowels (A48) | Hartley (CT--1/82) |
| Vowels Tutorial (A48) | Hartley |
| Reading Series (A, IBM) | Houghton-Mifflin |
| Hard and Soft C | Micro-Ed |
| Hard and Soft G | Micro-Ed |
| Anagram Word Game Series (T) | Microphys |
| Phonics and Word Attack (A32, AT, T) | Random House (EPIE & Consumers Union) |
| Phonics (AT 800) (EC--9-10/82) | Science Research Assoc. |
| Phonet (A32) System | Total Information Education |
| Vowel Search (see Softswap) | |

## Reading Skills--Vocabulary

| Relevant Software | Publisher |
|---|---|
| Jabbertalky (A) | Automated Simulations |
| Prefix Study and Quiz (T) | Basics and Beyond, Inc. |
| Definder (C64, VIC20, PET) | Comm-Data Systems |
| Word Fun (C64, VIC20, PET) | Comm-Data Systems |
| Game Show (A, IBM, C) | Computer Advanced Ideas |
| Wizard of Words (A, IBM, C) | Computer Advanced Ideas |
| Funbunch 1 and Computer Doodle (T) | Computer Learning Center 4 Children |
| Funbunch 2 and 3 (T) | Computer Learning Center 4 Children |
| Funbunch 4 and 5 (T) | Computer Learning Center 4 Children |
| Scramble--Superheroes and Farm Animals (T) | Computer Learning Center 4 Children |
| Scramble--Weekdays and Numbers (T) | Computer Learning Center 4 Children |
| Synonyms and Antonyms | Course Ware |
| Crosswords | Creative Discount Software: Edusoft |
| Teach me Words (K-2) | Creative Programmers |
| Word Attack (A, IBM) | Davidson & Associates |
| Synonym | Dr. Daley |
| Crossword Generator Gr. 3-8 (T) | Dynatek Information Systems, Inc. |
| Concepts in Language Arts (A, C, T) | Educational Activities, Inc. (EPIE & Consumers Union) |
| Anagrams (LAS-2) Homonyms I and II Synonyms and Antonyms I and II Contractions I and II | Educational Media Associates |
| Scramble-Grams (LAS-1) (T) | Educational Media Associates |
| PSAT/SAT Word Attack Skills | Edu-Ware |
| Word-Pak (T) | Edu-Ware East |
| Synoyms and Antonyms | Edu-Ware Services, Inc. |
| Vocab Drill #4 | Hartley Software |
| Reading | Houghton-Mifflin |

3

| | |
|---|---|
| Vocabulary Prompter/Super Prompter | Jagdstaffel Software |
| Word Games (A32) | Merry Bee |
| Visual Vocabulary (A32) | Merry Bee |
| Antonym Machine (PET) | Micro-Ed |
| Homonym Machine (PET) | Micro-Ed |
| Synonym Series (PET) | Micro-Ed |
| Word Demon Series (PET) | Micro-Ed |
| Junior/Senior High Vocabulary Program Series I-III (T, PET, CBM, A) Note: This is a series of 75 programs, available individually or by groups. | Microphys |
| Vocabulary Skills: Content Clues (A48) | Milton Bradley (EL--10/82) |
| Vocabulary Skills: Prefixes, Suffixes, and Root Words (A48) | Milton Bradley (EL--10/82) |
| Crossword | Minnesota Educational Computing Consortium (MECC) |
| Word Find | MECC |
| Word Game | MECC |
| Wordmaster (T). | NTS Software |
| Dr. Jekell and Mr. Hyde | Pendulum Press |
| Minicrossword (A, AT, T) | Program Design, Inc. |
| Story Builder/Word Master (A32, AT) | Program Design, Inc. (MJ--V3#1) |
| Vocabulary 1: Beginning (A) | Program Design, Inc. |
| Vocabulary 2: Advanced (A) | Program Design, Inc. |
| Word Meanings | Program Design, Inc. |
| Fundamental Word Focus (A, T) | Random House (EPIE & Consumers Union) |
| The Hibrow Vocabulary (T) | Ravagraph Company (CT--11/81) |
| Vocabulary Builder | Silwa enterprises |
| Language/Reading Development Program | Software Technology for Computers |
| Improve Your Vocabulary, I, II | Teach Yourself by Computer Software |
| Working with Synoyms and Antonyms | Teach Yourself by Computer Software |
| Working with Contractions and Homonyms | Teach Yourself by Computer Software |
| Hangman | Total Information Education Systems (TIES) |
| Scramble | TIES |
| Worder | TIES |
| Cushman (see Softswap) | |
| Hangman 1, 2, 3, 4 (see Softswap) | |
| Qs and Zs (see Softswap) | |
| Swap New Rom (see Softswap) | |
| Word Hunt (see Softswap) | |

Reading Skills--Recognizing Memory Words

| Relevant Software | Publisher |
|---|---|
| Remember (C64, VIC20, PET) | Comm-Data Systems |
| Master Match (A, IBM, C) | Computer Advanced Ideas |

4

| | |
|---|---|
| What Was That Word? (A32) | Edutek |
| High Speed Word Recognition (A48, AT) (MJ--W/81; SMW--S/81) | Edu-Ware Services, Inc. |
| Speechzapper (T) | Edu-Ware East |
| Alphabetize | Fullmer & Assoc. |
| Word-a-TACH (A) | Hartley Software |
| Word Flash (A48) | Ideatech |
| Word Memory (A48) | Instructional Communications Technology |
| Wordmatch (A) | Jadee Enterprises |
| Catch the Rhyming Fish | Micro-Ed |
| Matching Words | Micro-Ed |
| Word Search | Micro-Ed |
| Tachistoscope (PET) | Micro-Ed |
| Cross Clues (A48, IBM) | SRA, Inc. (ET--April 1983) |
| Language/Reading Development Program | Software Technology for Computers |
| Word Power (T) | The Teaching Assistant |
| Word Recognition | The Teaching Assistant |

## Reading Skills--Speed Reading

| Relevant Software | Publisher |
|---|---|
| Speed Reading and Comprehension (PET) | Abbott Educational Software (SMW--S/81) |
| Speed Reading (C64, VIC20, PET) | Comm-Data Systems |
| The Speed Reader (IBM) | Davidson and Associates |
| Tachistoscope (T) | Micro-Ed |
| Speed Read (see Softswap) | |
| Speed Read 2 (see Softswap) | |

## Reading Skills--Study Skills

| Relevant Software | Publisher |
|---|---|
| Great Classics (T) | Dorsett Educational Systems, Inc. |
| Readings in Literature (T) | George Earl |
| Microcourse Series (A, IBM) | Houghton-Mifflin |
| Reading | Houghton-Mifflin |
| Proof-It (A32) | Merry Bee |
| Library Skills (A32) | Micro Power and Light Co. |
| Memory Builder | Program Design, Inc. |
| Reading is Fun Series (T) | Radio Shack |
| Trickster Coyote (A48, AT48, IBM) | Reader's Digest Services, Inc. (EL--Vol.2, No.7, April 1983) |
| Comprehension Power Program (A48) | Taylor Associates (EL--JanFeb 82; CT--Feb 82) |
| Mind-Memory Improvement-Level I and II (T) | Teach Yourself by Computer |
| Reference Balloon (see Softswap) | |

5

Reading Skill—Comprehension

| Relevant Software | Publisher |
|---|---|
| Critical Reading (A48) | Borg Warner (ET--9/81; CCN--3-4/82; EPIE Report 81) |
| Four Basic Reading Skills--Unit 1 | Brain Box, Inc. |
| Past Tense/Present Tense (A, T) | Computer Courseware Services |
| Steps to Advanced Reading | Creative Curriculum, Inc. (EE--May/June 1983) |
| Alpine Skier (A48, T)<br>Determining Fact and Opinion<br>Categorizing Words and Phrases<br>Getting Sentence Meanings<br>Seeing Cause and Effect | Data Command |
| Big Door Deal (A48, T)<br>Making Analogies<br>Recognizing Figurative Language<br>Sequencing Events<br>Using Context Clues | Data Command |
| Diascriptive Reading (A, PET, T) | Educational Activities, Inc. |
| How to Read in the Content Areas (A, PET, T)<br>Mathematics, Science, Social Studies, Literature, | Educational Activities, Inc. (EPIE & Consumers Union) |
| Our Weird and Wacky World | Educational Activities, Inc. |
| Our Wild and Crazy World | Educational Activities, Inc. |
| Science Content Area Reading | Educational Activities, Inc. |
| Skilldrill #5, #7, #11 | Hartley Sofware |
| Comprehension Power Program (A) | I/CT & Milliken (EPIE & Consumers Union) |
| Questions and Stories (A48) | Ideatech |
| Cloze Plus (A48) | Instructional Communications Technology |
| Comprehension Power Reading Program (A48) | Instructional Communications Technology |
| Processing Power (A48) | Instructional Communications Technology |
| Context Clues | Learning Well |
| Drawing Conclusions | Learning Well |
| Following Directions | Learning Well |
| Getting the Main Idea | Learning Well |
| Inference | Learning Well |
| Reading for Detail | Learning Well |
| Identifying Complete Sentences (PET) | Micro-Ed |
| Reading Racer One (PET) | Micro-Ed |
| Advertising Techniques (A32) | Micro Power and Light Co. |
| Cloze Plus (A48) | Milliken Publishing Co. |
| Reading Comprehension (A48) | Milliken Publishing Co. (EL--1-2/82) |
| Reading Comprehension | Msss D., Inc. |
| The Subject at Hand (T) | NTS Software |
| Watch Your Language! (T) | NTS Software |
| Reader | OMNICO |

| | |
|---|---|
| Stories | OMNICO |
| Dr. Jekell and Mr. Hyde | Pendulum Press |
| Ask Me Why (A) | Pentar Software |
| Analogies (A) | Program Design, Inc. |
| Code Breaker (A) | Program Design, Inc. |
| What's Different (A, AT, PET, T) | Program Design, Inc. (SMW--S/82) |
| Fundamental Comprehension (A32, AT, T) | Random House |
| Reading Comprehension (A48, AT, T) | Random House |
| Understanding What You Read (A48, AT, T) | Random House |
| Word Blaster (A32, AT, T) | Random House |
| Reading Skills Courseware Series K-6 | Scott Foresman |
| Analogy | Silwa Enterprises |
| Sentence Completion (A) | Silwa Enterprises |

Reading Skills--Structural Analysis

| Relevant Software | Publisher |
|---|---|
| English Usage Exercises (T)<br>Vol I: Subject-Predicate Relationship<br>Vol II: Verbs, Nouns, and Prepositions<br>Vol III: Adjectives, Articles, Conjunctions, Pronouns, and Future Tense | 3R Software |
| Prefix Study and Quiz (T) | Basics and Beyond, Inc. (ET--3/82) |
| Antonym Match (C64, VIC20, PET) | Comm-Data Systems |
| Homonym Match (C64, VIC20, PET) | Comm-Data Systems |
| Synonyms (C64, VIC20, PET) | Comm-Data Systems |
| Dragon Game Series for Language Arts (A, AT, PET, T) | Educational Activities, Inc. |
| English Basics (T, A, C)<br>Vol 1: Nouns, Pronouns, Verbs, Adjectives, Adverbs<br>Vol 2: Homonyms, Synonyms, Antonyms and Contractions | Educational Activities, Inc. (EPIE & Consumers Union) |
| Pik-Pek-Put (A48, T)<br>Base Words<br>Contractions<br>Plurals<br>Possessives<br>Prefixes; Part I, Part II<br>Suffixes; Part I, Part II | Data Command |
| Riddle Me This<br>Contractions<br>Homonyms<br>Possessives | Data Command |

| | |
|---|---|
| Tank Tactics (A48, T) | Data Command |
| Conactions | |
| Base Words | |
| Homonyms | |
| Plurals | |
| Prefixes | |
| Suffixes; Part I, Part II | |
| Tennis Anyone! (A, T) | Data Command |
| Base Words | |
| Contractions | |
| Homonyms | |
| Plurals | |
| Prefixes | |
| Suffixes | |
| Antonyms/Synonyms (A) | Hartley (EPIE & Consumers Union) |
| Capitalization (A) | Hartley |
| Roots/Affixes (A) | Hartley |
| Word Families (A) | Hartley |
| Reading | Houghton-Mifflin |
| Sequence (A48) | Learning Well |
| Contractions | Micro-Ed |
| Compound Words | Micro-Ed |
| Dropping the Final E | Micro-Ed |
| Prefix Series | Micro-Ed |
| Suffix Series | Micro-Ed |
| Grammar Package I (A, T) | Micro Learningware |
| Adjective Recognition | |
| Adverb Recognition | |
| Noun Recognition | |
| Verb Recognition | |
| Pronoun Recognition | |
| Person, Place or Thing | |
| Grammar Package II (A, T) | Micro Learningware |
| Contractions | |
| Possessive Case | |
| Prefix/Suffix | |
| Spelling Rules | |
| Subject-Verb Agreement | |
| Vocabulary Skills | Milton Bradley |
| Structural Analysis | Msss D., Inc. |
| Prefixes, Suffixes, Root Words | |
| Snookies--Preposition Identification (T) | NTS Software |
| Word Skills 1, 2: Prefixes | Program Design, Inc. |
| Word Skills 3: Suffixes | Program Design, Inc. |
| Jabberwocky (T) | Wise Owl Workshop |

Comprehensive Language Arts Packages

| Relevent Software | Publisher |
|---|---|
| Micro-Read Levels 1-8 (A48) (Includes 73 diskettes, student materials, user's guide and skills index, and story card. Covers word analysis and vocabulary skills, comprehension skills, study skills and more.) | American Educational Computer, Inc. |
| Houghton-Mifflin Microcourse in Language Arts Levels 1-8 (A48, IBM) (Includes 440 skills on 136 diskettes user's guide, folders and binders.) | Houghton-Mifflin |

## Listing of Courseware by Language Skills Area

### Writing Skills--Written Expression

| Relevant Software | Publisher |
|---|---|
| Jabbertalky (A) | Automated Simulations |
| Beginning and Advanced Composition (A, IBM, C, VIC20) | Behavioral Engineering |
| At the Zoo | Bertamax |
| Story Mix 1 (A32, T) | Bertamax (CT--Vol.10, No.9, May 1983) |
| Wordy: Common Forms of Wordiness | COMpress |
| Language Arts Topics: Letter Writing; Subject-Verb Agreement; Sentence Structure | Computer Curriculum Corporation |
| Writing Competency Program (A48, PET, T, AT) | Educational Activities |
| Language Arts | Houghton-Mifflin |
| Story Starter (T) | Random House |
| Story Builder (T) | Random House |
| Story Machine (A, AT) | Spinnaker |
| Compupoem (South Coast Writing Project) | Stephen Marcus |
| Cinquain (see Softswap) | |
| Computer Haiku (see Softswap) | |
| Computer Poetry (see Softswap) | |

### Writing Skills—Capitalization

| Relevant Software | Publisher |
|---|---|
| Punctuation and Capitalization | Bertamax |
| Capitalization | Educational Activities |
| Capitalization (A) | Hartley |
| Language Arts | Houghton-Mifflin |
| Nouns | Program Design, Inc. |
| Capitalization (CBM, PET) | Teacher's Pet |

---

| | |
|---|---|
| A | Apple |
| AT | Atari |
| C | Commodore (model not specified) |
| Pet | Commodore Pet |
| VIC20 | Commodore VIC20 |
| T | TRS-80 |
| CBM | Commodore CBM |
| IBM | IBM |

| | |
|---|---|
| CC | Creative Computer |
| CT | Classroom Teacher |
| EC | Educational Computer |
| ET | Educational Technology |
| SMW | School Microware Reviews |
| MJ | Macul Journal |
| EL | Electronic Learning |
| CCN | Classroom Computer News |
| JEM | JEM Reference Manual |
| EPIE | EPIE Reports |
| EE | Electronic Education |

## Writing Skills--Usage

| Relevant Software | Publisher |
|---|---|
| English Usage Exercises (T)<br>Vol I: Subject-Predicate relationships<br>Vol II: Verbs, Nouns, and Prepositions<br>Vol III: Adjectives, Articles, Conjunctions, Pronouns, and Future Tense | 3R Software (CC--10/81) |
| Ask25 English-Verbs, Present Tense Gr. 4-6 (T) | Ask Enterprises |
| Ask26 English-Nouns, Plural/ Singular Gr. 4-6 (T) | Ask Enterprises |
| Ask6 Spelling-Contractions Gr. 4-6 (T) | Ask Enterprises |
| Ask9 English-Kinds of Sentences Gr. 4-6 (T) | Ask Enterprises |
| Effective Writing (AT) | ATARI Educational System (EPIE & Consumers Union) |
| Sentence Diagramming (A48) | Avant Garde Creations (CT--V8#7; CC--10/81) |
| Prefix Study and Quiz | Basics and Beyond |
| Antonym Match (C64, VIC20, PET) | Comm-Data Systems |
| Homonym Match (C64, VIC20, PET) | Comm-Data Systems |
| Synonyms (C64, VIC20, PET) | Comm-Data Systems |
| Parts of Speech (C64, VIC20, PET) | Comm-Data Systems |
| Language Arts Topics:<br>Contractions<br>Modifiers<br>Negatives<br>Possessives<br>Principal Parts of Verbs<br>Pronoun Usage<br>Verb Usage | Computer Curriculum Corporation |
| Dragon Game Series Gr. 3-6 (A, T, PET)<br>The Dragon of Adjecives<br>The Dragon of Antonyms<br>The Dragon of Contractions<br>The Dragon of Nouns<br>The Dragon of Synonyms<br>The Dragon of Verbs | Educational Activities |
| English Basics: Parts of Speech (A, T, PET)<br>Adjectives I and II<br>Adverbs I and II<br>Nouns I and II<br>Pronouns I and II<br>Verbs I and II | Educational Activities (EPIE & Consumers Union) |

| | |
|---|---|
| Writing--Skills Development Packages-IMAP (T) | Educational Media Associates |
| Apple Grammar Professional | Educational Software |
| Crazy Sentences (A) | Edutek |
| Parts of Speechzapper | Edu-ware East |
| Scandura Microtutor II Language Arts Levels I-III (Word Structure) (A48) | Entelek |
| Nouns/Pronouns (A) | Hartley |
| Verbs (A) | Hartley |
| Verb Usage 1 & 2 (A) | Hartley |
| Who, What, Where, When, Why (A48) | Hartley |
| Language Arts | Houghton-Mifflin |
| English Achievement I-V (A48, PET) | Microcomputer Workshops (EL--9/82) |
| Agreement of Subject and Verb (PET) | Micro-Ed (SMW--S/81) |
| The Apostrophe | Micro-Ed |
| Grammar Series<br>Agreement of Subject and Verb<br>Make a Sentence (PET)<br>The Verb<br>The Adjective<br>The Adverb<br>The Noun<br>Subject and Predicate | Micro-Ed |
| Troublesome Pronouns | Micro-Ed |
| Usage Boners (A, T, PET, VIC) | Micro-Ed |
| Grammar Package (A, T)<br>Vol I: Adjective, Adverb, Noun, Verb, Pronoun, Place or Thing<br>Vol II: Contractions, Possessive, Prefix/Suffix, Spelling Rules, Subject-Verb Agreement Sentences | Micro Learningware |
| (A32) | Micro Power and Light Co. |
| Milliken Language Arts Programs (A48)<br>Grammar Problems for Practice: Homonyms<br>Grammar Problems for Practice: Modifiers<br>Grammar Problems for Practice: Pronouns<br>Grammar Problems for Practice: Verbs | Milliken |
| Homonyms (T) | NTS Software |
| The Subject at Hand (T) | NTS Software |
| Watch Your Language! (T) | NTS Software |
| Story Builder/Word Master (A) | Program Design, Inc. |
| Homonyms in Context (A48, T) | Random House (EL--9/82 & 5-6/82) |
| Language/Reading Development Program (A) | Software Technology for Computers |
| English Grammar<br>Working with Adverbs<br>Working with Adjectives<br>Working with Pronouns and Propositions<br>Working with Subjects and Nouns | Teach Yourself by Computer Software |
| Verbs (CBM, PET)<br>Teacher's PET<br>A or An (see Softswap)<br>Ad-Libs | |

143

Ask (see Softswap)
Subjects (see Softswap)
Verbs (transitive, intransitive, and being) (see Softswap)

## Writing Skills--Punctuation

| Relevant Software | Publisher |
|---|---|
| Ask27 English-Using the Comma Gr. 4-6 (T) | Ask Enterprises |
| Punctuation and Capitalization | Bertamax |
| Language Arts Topics: Punctuation, Capitalization | Computer Curriculum Corporation |
| Capitalization | Educational Activities |
| Punctuation I: The Period, Question Mark, (A48, T) and Exclamation Point | Educational Activities (EL--9/82) |
| Punctuation II: The Comma (A48, T) | Educational Activities (EL--9/82) |
| Quotation Marks | Educational Activities |
| Writing--Skills Development Packages-IMAP (T) | Educational Media Associates |
| The Apostrophe (PET) | Micro-Ed |
| End Punctuation (PET) | Micro-Ed |
| Punctuation Series (A, PET, T) | Micro-Ed |
| Run-On Sentences (PET) | Micro-Ed |
| Commas (A) | Micro Power and Light |
| Sentence Combining (A48) | Milliken (EL--May/June 1983) |
| Punctuation Skills: Commas (A48) | Milton Bradley (EPIE & Consumers Union, ET--April 1983) |
| Punctuation Skills: End Marks, Semicolon, and Colon (A48) | Milton Bradley |
| Fundamental Punctuation Practice (A48, T) | Random House |
| Commas (CBM, PET) | Teacher's Pet |

## Structure Skills--Spelling and Encoding

| Relevant Software | Publisher |
|---|---|
| Spellbound--Your Boggle Partner (T) | The Alternate Source |
| Ask6 Spelling-Contractions Gr. 4-6 (T) | Ask Enterprises |
| Word Scrambler/Super Spelr (A48) | Avant Garde Creations |
| Spelling Strategy (A, IBM) | Behavioral Engineering |
| Spelling in Context Levels I-VII (A, AT, T) | Bertamax |
| Spelling Teacher Software | Computer Corner |
| Education II | CompuSo Co. |
| Drill | Cook's Computer Company |
| Scramble | Course Ware |
| Spell-N-Time (T) | Course Ware |
| Spelling Test (A) | Creative Computing |

| | |
|---|---|
| Crosswords | Creative Discount Software: Edusoft |
| Scramble | Creative Discount Software: Edusoft |
| Dieting Dinosaur (Elementary & Middle) (T) | Curriculum Applications |
| SpellIIC and SpellIID: Remedial Spelling (T) | D & M Software |
| Spell | Dr. Daley |
| Hangman | Dr. Daley |
| Flash Spelling | Educational Activities |
| Scrambled Letters (A, PET, T) | Educational Activities (CC--9/80; CT--V8#2) |
| Spelltronics (A, PET, T) | Educational Activities |
| Anagrams (LAS-2) | Educational Media Associates |
| Spelling Sorcery (A) | Edu-Ware East |
| Compu-Spell (A48) | Edu-Ware Services, Inc. (ET--10/81; SMW--S/81) |
| Spelling Bee with Reading Primer (A48) | Edu-Ware Services, Inc. (EC--3-4/83) |
| Alphabetize | Fullmer & Associates |
| Horrible Homonyms | George Earl |
| Hangman (T) | George Earl |
| Skilldrill #2, #6 | Hartley Software |
| Spelling Test #3 | Hartley Software |
| Colorguess (A) | Ideatech |
| Word Demon (A) | Ideatech |
| Word Flash (A) | Ideatech |
| Vocabulary Baseball (A) | J & S Software |
| Learn to Spell (T) | Jensen Software (JEM Reference Manual |
| Word Completions (T) | Jensen Software |
| Magic Spells (A48) | Learning Company |
| Spelling Bee (A48) | Mary Bee |
| Spelling Bee Games (A) | Mary Bee |
| Spelling I, II (A32, T) | Mentor Software |
| Dropping the Final E | Micro-Ed |
| Extra Practice Spelling Series (T) | Micro-Ed |
| Words in Context Spelling Series (A, PET, T) | Micro-Ed |
| Anagram Word Game Series (A) | Microphys |
| Wheel-of-Fortune Word Game (T) | Microphys |
| Junior/Senior High Spelling Program (A, PET, CBM, T) Note: This is a series of 30 programs available individually or by group. | Microphys |
| Elementary Package III (T) | Micro Learningware |
| Spelling Rules (A32) | Micro Power and Light Co. (SMW--W/82 |
| Mixup | MECC |
| Spell | MECC |
| Spellcaster/Spellbuilder (T) | NTS Software |
| Words are Fun: Sets 1 and 2 (T) | NTS Software |
| Do It Yourself: Spelling (A) | Program Design, Inc. |
| Let's Spell | Program Design, Inc. |
| Spelling Builder (A, AT, T) | Program Design, Inc. |
| Customized Flash Spelling (T) | Random House |

| | |
|---|---|
| Fundamental Spelling in Context (A, T) | Random House (EPIE & Consumers Union) |
| Spelling Demons/Computer Spelldown (T) | Random House |
| The Spelling Machine (A48) | SouthWest Ed Psych (SMW--W/82) |
| Whole Branin Spelling (A) | SubLOGIC Communications Corp. |
| Language/Reading Development Program | Software Technology for Computers |
| Words for the Wise (T) | TYC Software |
| Spelling Package (with audio cassette) (PET) | Teaching Tools (EL--9/82; SMW--W/82) |
| Spelling | Teach Yourself by Computer Software |
| Spelling One Drill and Practice for One (T) | Terra 80 Software |
| Spelling Two Drill and Practice for Many (T) | Terra 80 Software |
| Spell 1, 2 | TIES |
| Magic Spells | The Learning Tree |
| Spellbound (see Softswap) (T) | |
| Spell Endings (see Softswap) | |
| The Word Market (see Softswap) | |

## Listening Skills--Auditory Discrimination

| Relevant Software | Publisher |
|---|---|
| Skilldrill #1 and #2 | Hartley Software |
| Spelling Test #3 | Hartley Software |
| Vocab Drill #4 | Hartley Software |
| Listen and Spell (wh-th words) (T) | Little Bee Educational Programs |

## Listening Skills--Aural Comprehension

| Relevant Software | Publisher |
|---|---|
| Skilldrill #5 | Hartley Software |
| Speak and Read Module Package Gr. 1-3 | Scott Foresman |

## Listening Skills--Speaking Skills

| Relevant Software | Publisher |
|---|---|
| Dramatizing--Short Vowel Exercise | Edutek |
| Oral Reading--Short Vowel Exercise | Edutek |
| Oral Expression--Short Vowel Exercise | Edutek |

## Word Processing and Authoring Programs

| Relevant Software | Publisher |
|---|---|
| Instructor [authoring system] (A) (Note: Requires BCD interface) | BCD Associates |

MicroTeach [authoring system] (AT) — Compumax

(Compute, May 1983)

Peachy Writer (A) — Cross Educational Software

The Learning Box (A48) — M. D. Fullmer & Associates

The Overdue Writer (A48, T48) — The Library Software Company

(EL--Vol.2, No.7, April 1983)

Bank Street Writer (A48) — Scholastic Inc.

Easywriter

William Wresch

153

16

# APPENDIX C

SCHOOL MICROWARE EVALUATION FORM (COPY AS NEEDED)

Your Name_____________ Organization_____________ Position_____________

Address_____________ Tel:_____________

Product Name_____________ Supplier_____________ Price $_____ No. of Progs. Under This Name_____

Subjects and Grades to Which Applicable_____________ Est. Student Time Required_____________

FUNCTIONAL DESCRIPTION - Describe the program in terms of its goals and what it does to achieve them. Give as much detail as possible (use extra sheets as necessary).

_____________

_____________

_____________

PRELIMINARY CONSIDERATION - Does this program contribute to the teaching of topic(s) which should be taught in today's schools? ___Yes ___No If No, give your reasons for this answer in the Comments Section at the end of the form and omit the balance of the questionnaire.

INSTRUCTIONS - Enter a number in the blank to indicate the extent to which the product fulfills the description in the item, as follows: 2 - Completely, 1 - Partially, 0 - Not at All. If the item is not applicable to the product, enter N/A. If the item is unclear, enter U. Elaborate on answers as necessary in Comments at end or on extra sheets, giving item numbers.

DOCUMENTATION - List materials accompanying the program, e.g., teachers guide, student workbook.

1. Indicate types of information included.
   - _____ a. Suggested course/subject, grade levels.
   - _____ b. Goals.
   - _____ c. Performance objectives.
   - _____ d. Suggested teaching strateg(ies).
   - _____ e. Correlation with standard texts.
   - _____ f. Prerequisites for use of program.
   - _____ g. Student exercises, teacher answers.
   - _____ h. Operating instructions.
   - _____ i. Listing and sample runs of program(s).
   - _____ j. If a simulation, description of the model used.
   - _____ k. Suggested topics for follow-up discussions.
   - _____ l. Suggested references/activities for follow-up.
2. _____ The documentation is written clearly.
3. _____ If a workbook is included, the format and content are appropriate.

INSTRUCTIONS GIVEN TO USER BY PROGRAM

1. The instructions are adequate regarding:
   - _____ a. The instructional task to be performed.
   - _____ b. Details of how to interact with the program.
2. _____ User has the option of skipping instructions if already known.

STUDENT-COMPUTER DIALOG

1. _____ Output is displayed screen by screen (paged) rather than scrolled.
2. If output is paged:
   - _____ a. User has control over continuing to the next page.
   - _____ b. Amount of information in each page is appropriate.
   - _____ c. The perceptual impact (amount of type and lines) is suitable.
3. _____ Output is spaced and formatted so as to be easily readable.
4. _____ Language is well suited to most students' reading ability.
5. _____ Uses correct grammar, spelling, hyphenation and punctuation.
6. _____ Any grid or coordinate system used is consistent with common conventions.
7. _____ Students can respond with common symbols & ways of using them, e.g., right to left entry of sums.
8. _____ Accepts abbreviations for common responses.
9. _____ Provides for individual needs, e.g., opportunity to work with harder or easier material.
10. _____ Dialog is personalized, i.e., makes appropriate use of student names.
11. _____ Uses devices to get & maintain interest, e.g., variation of computer responses, humor, pace change, surprise.
12. Makes good use of any special features computer:
    a. Graphics _____ b. Color _____ c. Sound _____
13. _____ Reinforcing responses (indications of right, wrong, etc.) are appropriate.
14. _____ The number of wrong answers allowed is reasonable.
15. _____ Responds appropriately if allowed number of wrong answers is exceeded.
16. _____ Provides opportunity to get help if difficulty is encountered.
17. _____ Minimizes bad entries via devices such as objective formats (multiple choice, etc).
18. _____ Deals well with inappropriate entries, i.e., response to typing errors, etc., is intelligible and useful.
19. _____ Required entries are within students' capabilities (esp. typing, vocabulary).
20. _____ Reports student performance periodically and at end of session.

MISCELLANEOUS CONCERNS

1. _____ If a simulation, the program gives a sufficiently accurate representation of the situation simulated.
2. _____ The concepts and vocabulary required to use the program are reasonable.
3. _____ Operates properly and is free of bugs.
4. _____ Is well structured and documented internally to facilitate any necessary debugging/modification.

COMMENTS - Please use this space and additional sheets as necessary to provide any information which you believe would help someone who is thinking about buying of the product being reviewed. In particular, indicate what you like best and least about the program. Also, list any changes which should be made.

_____________

_____________

_____________

APPENDIX D

Evaluation Questions: School CourseWare Journal

**Is the program demonstrably 'user friendly'?**

- Does it always use paging (*not* scrolling)?
- Can the user control the speed of presentation?
- Can the user control going back to a previous frame or leaving the program at any desired time?
- Is the screen output easily readable?
- Is the reading level correct for the target student?
- Does it use correct grammar, spelling, hyphenation and punctuation?
- Does it accept abbreviations for NO (N), YES (Y), etc.?
- Is it individualized, e.g., does it change to meet individuals' differing needs?
- Is the dialogue personalized?
- Can the student get help?
- Can the program be easily modified to suit individual teacher and student needs?
- Can you get to the program 'guts' to examine program coding?
- Does the program never ask for more than one data item at a time?

**Is the program 'user proof'?**

- Is the program 'crash-proof,' e.g., will it continue even if an inappropriate key is struck?
- Does the program reject ALPHA when NUMERICS are expected (and v.v.)?
- Does it reject other characters beside ALPHA and NUMERIC unless they are needed by the program?
- Does it reject too-long inputs from the user?
- Does it reject a NULL input (return key only) unless the program is using the NULL input as a response?
- Does it reject 'garbage' input? (This means input must be validated before it is used.)
- Does it refuse to 'freeze' or cycle in an endless loop if the user doesn't know the answer?

**Does the program do best what the computer does best?**

- Does it keep a running timer?
- Does it maintain a continuing score?
- Can it instantly adjust to the rate of presentation based on the accuracy rate of the student's response?
- Does it give instant feedback on performance?
- Does the student control movement forward or backward in a step-by-step development?
- Does it simulate difficult, costly, time-consuming or dangerous events?
- Are students motivated to improve their 'score'?

**Does it provide the printed documentation required for effective use in the classroom?**

- Is the program's name given?
- Is the author's name and address stated?
- Does a copyright notice appear in print and at the beginning of the program?
- Is the name of the target microcomputer clearly displayed on the cover, on the media, in print and in the program?
- Are instructional objectives stated?
- Are specific chapters of widely used textbooks cited for use with the program?
- Is the abstract of the program adequate?
- Are teaching strategies and techniques illustrated?
- Are examples of use and/or application given?
- Are prerequisite skills needed described?
- Are the student guides readable?
- Are student worksheets and record sheets provided?
- Are class record sheets given?
- Has the program been classroom tested?
- Has a panel of teachers given an independent evaluation?
- Has the degree of student enjoyment been tested?
- Is the logic of the program explained clearly?
- Is top-down, structured, modular programming used?
- Is there an adequate teacher's guide which includes instructions on how to use and modify the program?

**Is additional documentation given so the program can be used to teach programming?**

- Is the coding an exemplary model of readable, structured, modular style?
- Is a description of variables table given?
- Is an index of sub-routines given?
- Is a complete, readable listing of the program given?
- Is a complete pseudocode or flow-chart logic given?
- Is the pseudocode top-down, structured, one-way-in/one-way-out, modular style?
- Is the program well-structured?
- Is the program code internally well-documented with many REMarks?

**Is quality assurance given?**

- Have other programs of the supplier been used successfully in hundreds of schools?
- Have student and teacher satisfaction been assessed objectively?
- Has the program been classroom-tested?
- Are published reviews available?
- Is replacement guaranteed within 30 days if the program fails to perform mechanically?
- If the teacher is not satisfied, does the guarantee provide full credit within a 30-day period?
- If you have a problem, does the supplier respond rapidly by letter or phone?
- Does the supplier have a philosophy of providing only the highest quality, user-friendly, user-proof courseware?