DEVELOPMENT OF A COMPUTER-ASSISTED INSTRUCTIONAL PROGRAM TO TEACH WORD PROCESSING TERMINOLOGY

by

Marjorie Keatley

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

DOCTOR OF EDUCATION

in

Vocational-Technical Education

APPROVED:

B. June Schmidt, Chair

F. Marion Asche

Jerald F. Robinson

Walter L. Shell

Daniel E. Vogler

May, 1987

Blacksburg, Virginia

ACKNOWLEDGMENTS

I wish to thank the members of my committee for their support and encouragement during the completion of this study: Dr. F. Marion Asche, Dr. Jerald F. Robinson, Dr. Walter L. Shell, and Dr. Daniel E. Vogler. A special thanks is due Dr. B. June Schmidt, chair of my committee, for her encouragement, support, and sense of humor. Without her quick edits and helpful comments, the task would have taken much longer.

A debt of gratitude is due my parents, James A. and Alma E. Keatley, for their love, encouragement, and prayers during the years of this study. An equal debt is due Mrs. Ann Robinson, in whose home I have lived, for her love and friendship, and for pushing me when I needed it.

Thanks are due my colleagues in the Office Technology Department at Marshall University who helped during the program testing and who shouldered my load many times during the last year of this study. To my former students, Christine Thorn, Michele Roberts, and Sanna Bexfield, I owe a very special thanks for the many hours they gave to ease my load during the final push to finish.

Two friends share a special place in these acknowledgments. Pamela Ford provided much personal and professional support. The many hours she spent evaluating the program and giving assistance on the mainframe were crucial to the completion of the study.

Without the endless patience of Vicki Thornhill, who provided friendship, encouragement, and constant help, the completion of this study would still be unrealized.

To Dr. Robert Sullins, who prodded me, encouraged me, and believed in me, I owe special thanks. His sense of humor and his wise counsel helped me keep things in perspective.

The acknowledgments would not be complete without a heartfelt thanks to my fellow Pi Pi Beta Mu graduate students at UCOB for sharing the good times and the hard times. Their understanding and encouragement sustained me through the rough spots.

Most of all, thanks to God, through Whom all things are possible.

TABLE OF CONTENTS

Chapter Page

Chapter Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

STUDY OVERVIEW

The growing use of computers in the workplace has emphasized the need for computer literacy and skills. The word processing area has been especially affected as computers have become standard office equipment for the processing of words. Educators preparing students for word processing must maximize the available instructional time by providing students with as much hands-on computer training as possible. Students in word processing programs must have a general knowledge of computer hardware and software as well as special word processing skills.

To facilitate the acquisition of word processing skills, instruction in a common core of word processing terminology is needed. Availability of this instruction could result in more efficient use of class time, with more time available for hands-on training. Computer-assisted instruction affords a learning strategy that can be used to provide a common core of word processing terminology and expand the use of computers in the classroom.

While use of a computer-assisted instructional program provides for more efficient use of class time, it must be shown to be successful at meeting performance objectives to justify its implementation. The evaluation process used during the development of the computer program in this study ensured that performance objectives were met.

This chapter (a) provides a background and need for the study, (b) describes the research problem, (c) defines terms used in the study, and (d) outlines delimitations and research assumptions.

Background

A decade ago, computer education was reserved for students who planned careers in computer-based fields. The proliferation of computer technology and the availability of the microcomputer have forced educators to broaden the base of computer education to include all students. Educators are learning that they must not only teach about, but also with, computers. The place of the computer in the curriculum is being reevaluated, with increasing pressure to incorporate it into as many courses as possible (Goddard, 1984; Lundgren & Lundgren, 1984).

No area of study has been more affected by the growth of computer technology than business education. The proliferation of computer-based hardware and software as standard office equipment has " . . . placed business education at the crossroads of change" (Jorgensen & Campbell, 1983, p. 25). "If business education is to survive in this new [technological] environment, new methods and new courses will need to be incorporated into the curriculum. These new methods and new courses must involve the use of computers" (Stocker, 1981, p. 25).

According to Gary, Callahan, and Lanoue (1985), "the greatest challenge facing business education is to develop an instructional delivery system that reflects the rapidly changing technology" (p. 28). One means of delivery to meet this challenge is computer-assisted instruction (CAI).

The use of computer-assisted instruction is especially relevant to the area of word processing. In the community college setting, students take word processing for a variety of reasons, ranging from curiosity to preparation for employment. The collection of word processing terms that these students bring to the classroom is as varied as their reasons for taking the course. Valuable class time must be spent in teaching new or alternative terms to simplify the delivery of hands-on instruction.

A survey of available textbooks, literature, and software documentation revealed that there is no standard vocabulary of word processing terminology. Use of CAI materials can provide a method for teaching a common core of word processing terminology while meeting the varying needs of the students, facilitating the delivery of hands-on training, and reinforcing data entry/word processing skills at the same time terminology is being learned (Drum, 1981-82).

A major obstacle to using computer-assisted instruction, however, is the lack of quality software (Alty, 1983; Holmes, 1982). More often than not, available software is created by individuals whose background is not primarily education (Podemski, 1981). Moreover, the software has largely been developed without tryouts or evaluation to locate and resolve problem areas (Golas, 1983). The development of the microcomputer has resulted in a flood of "sugar coated" applications software and educational courseware as designers have rushed to take advantage of this new, low-cost technology (Hofstetter, 1985).

One solution to the problem of quality software is to use formative evaluation procedures in the development process. Formative evaluation utilizes a systematic instructional design to evaluate and validate a product empirically. It involves the stages of planning, development, revision, and field testing. A priori performance objectives are set during the planning stage to ensure standards against which the completed product can be evaluated.

The developmental and field testing stages in the formative evaluation process enable the designer to identify and correct weaknesses in the program content, instructional strategies, and delivery. The end result is the ". . . improvement of the quality and deliverability of instruction" (McMeen, 1986, p. 42).

As the demand for more and better courseware increases, "standards for quality in educational software will become more widely accepted and applied" (McMeen, 1986, p. 44). The consequent availability of quality courseware will help ease the shift to a more computer-oriented curriculum.

Assumptions

Certain assumptions were made by the researcher in planning this study.

1. It was assumed that CAI is an appropriate method for teaching a common core of word processing terminology.

2. It was assumed that formative evaluation is an appropriate procedure for validating computer-assisted instructional programs.

Statement of the Problem

The increasing use of the computer as standard office equipment has created a need to integrate the computer into the business education curriculum. Instructional materials are needed that not only provide training on the computer but also utilize the computer for the delivery of instruction. Students in the word processing area, in particular, would benefit from computer instructional materials that promote the acquisition of word processing terminology, which would facilitate the delivery of hands-on instruction. This goal could be accomplished by using a computer-assisted instructional program to teach a common core of word processing terminology. The result would be a more efficient and effective use of class time.

Dependence upon computer-assisted instruction is limited, however, by the lack of quality software. Much of the available software has been developed without benefit of an evaluation process to ensure its validity. Vasek (1983) and Sherron (1984) have shown that validated software can be developed through formative evaluation procedures. Development of software can be expedited through the use of an authoring language. This approach enables an individual with little or no programming skill to program instructional materials tailored to specific needs.

Although the use of a computer-assisted program to teach word processing terminology would facilitate delivery of hands-on instruction, the program must be educationally sound to make its implementation worthwhile. No studies were found in the literature that dealt with the development and validation of software to teach word processing terminology, either with or without the use of an authoring language. The literature did indicate that the use of tutorials is common as a method of presenting basic word processing information to consumers of word processing software (Regan-Goff, 1985). Regan-Goff cited three advantages for using tutorials in word processing classes:

1. Use of the tutorial gets the student working with the system immediately, which is particularly important for those students with no prior exposure to computers.

2. The tutorial can teach word processing in the sense that it introduces the commands and the effect they produce on the screen.
3. The tutorial can be used as an introduction to word processing functions and capabilities to get the student comfortable working with the equipment.

Although commercial tutorials may have utility in the classroom, there is no evidence that these programs are developed using recommended principles of design and learning theory. Therefore, the problem of this study was to develop a validated computer-assisted program to teach a common core of word processing terminology using an authoring language.

Purpose Statements

The purpose of the study was to develop a computer-assisted program to teach word processing terminology that would provide a common core of terms and facilitate classroom instruction. To achieve this main objective several ancillary procedures were undertaken. The researcher

1. synthesized the literature.
2. identified word processing terms from the literature.
3. defined the selected terms from the literature.
4. determined the preferred sequence of instruction for the word processing terms according to a panel of word processing experts.

5. identified performance objectives for the computer program.
6. developed a criterion performance test.
7. developed a computer-assisted instructional program to teach word processing terminology.
8. validated the computer-assisted instructional program through use of formative evaluation procedures.
9. conducted field tests of the computer-assisted instructional program.
10. assessed the effectiveness of the computer-assisted instructional program in meeting performance objectives.
11. assessed the appropriateness of an authoring language for developing a computer-assisted instructional program to teach word processing terminology.

Research Questions

The following research questions were addressed by the study. The first five questions were addressed through a synthesis of the literature.

1. What are the word processing terms and definitions that are revealed in the literature?
2. What are the performance objectives to be met upon completion of the CAI program?

3. What are the components of the criterion performance test?
4. What are the steps in the development of a computer-assisted instructional program to teach word processing terminology?
5. How can formative evaluation be utilized to improve a CAI program?

The following question was addressed through survey analysis:

6. What is the preferred sequence of instruction for the word processing terms identified in the literature according to a panel of word processing experts?

The following question was addressed through an analysis of the developmental and field test data:

7. How effective was the CAI program in meeting the a priori performance objectives?

The final question was addressed through a survey of the literature, discussion with a programming expert, and program development and use:

8. Is an authoring language an appropriate programming method for this computer program?

Definitions of Terms

Authoring language: a programming language that allows for development of software without prior knowledge of a general-purpose computer programming language.

Courseware: material that is designed to be used on a computer and to help teachers use the computer for instruction.

Criterion performance test: a test administered at the end of an instructional program to assess the effectiveness of the program in meeting a priori performance objectives.

Formative evaluation: a systematic process of planning, developing, revising, and field testing instructional materials.

Computer-assisted instruction (CAI): an instructional approach that uses the computer as its method of delivery.

One-to-one evaluation: the initial evaluation of computer-assisted instructional materials, which is conducted with a subject representative of the target population for the materials. The evaluation process is closely observed to note any problems encountered as the subject works through the instructional materials.

Word processing terminology: terms that relate to or define specific word processing procedures.

PILOT (Programmed Inquiry, Learning or Teaching): an authoring language developed for use with the microcomputer.

Delimitations

The terminology was delimited to terms selected by a panel of word processing experts from those terms identified in selected textbooks and software documentation. The identification of terms was restricted to those related to document processing.

The target group for selection of participants for this study was community college students with no prior word processing instruction. This group was selected because of the applicability of the program to office technology and data processing offerings in the community college.

Due to cost and time considerations, the program was delimited to one unit of instruction on terms related to document processing. The program was further delimited to one hour of instruction. The program was written using a version of PILOT authoring language for the IBM-PC or a compatible machine.

Need for the Study

"The acquisition of basic computer competencies is of increasing importance to all students for success in an information society" (McMeen, 1986, p. 43). The competitive nature of today's market places an increasing demand on educators to provide students with state-of-the-art training. The search for effective ways to provide maximum hands-on instruction in word processing has focused attention on the need for instruction in a common core of word processing terminology. A computer-assisted instructional program is needed that can provide such instruction.

Study Organization

This chapter provided an overview of the study. Chapter II discusses the literature. Chapter III details the research and data collection procedures. Chapter IV presents the results of the formative evaluation and data analyses and chapter V presents conclusions and recommendations based upon the research findings.

CHAPTER II

REVIEW OF LITERATURE

The major areas included in the review of the literature are (a) programmed instruction, (b) computer-assisted instruction, and (c) formative evaluation. Programming systems and languages are also reviewed.

Programmed Instruction

The following section details the background and process of programmed instruction. Rules for frame construction are discussed as well as principles of learning that are applicable to the development of programmed instruction. The section concludes with a summary of the major points.

Background

Programmed instruction was introduced as early as 1866 in the form of teaching machines for spelling (Stolurow, 1961). About 1915, Pressey began to mechanize testing and teaching in an attempt to increase attention and learning in the classroom. It was not until Skinner's work with psychological learning theories was publicized during the 1950s, however, that programmed instruction began to be used as a serious educational technique (Bloom, Hastings, & Madaus, 1971).

In the early days of its development, programmed instruction was predominantly in the form of programmed texts, possibly due to a lack of funds for research on teaching machines (Evans, 1965). These texts were written in various formats designed to conceal the answers from the learner until the material had been presented and an effort had been made by the learner to answer questions.

With the invention of computers, programmed instruction found a new avenue of delivery. Its use was hindered, however, by fears that programmed instruction would replace teachers. These fears have been allayed to a great extent as the computer has come to be viewed as a complementary instructional tool rather than as an alternative to the teacher.

Whether programmed instruction is in the form of textbooks or computer programs, it is developed in the same fashion. Cruz (1984) notes that there are no standardized rules for preparing programmed materials. However, the process follows four general steps (p. 30): (a) specification of terminal objectives in behavioral terms; (b) analysis of the subject, including construction of pretests and posttests, and specifications of the target population; (c) design and development of material; and (d) evaluation, revision, field testing, and implementation until students' performance meets the program objectives.

Program Objectives

The establishment of objectives is the starting point for determining the learning experiences and instructional strategies needed for the program. The objectives also provide the basis for performance tests, which must be developed before the program is created to avoid contamination of the program content (Cruz, 1984).

Subject Analysis/Program Development

A subject or task analysis helps the developer concentrate on specific instructional objectives, learning strategies, and instructional sequences. Once these have been selected, the program is designed and developed.

Evaluation

"Even with proven design procedures, empirical tryout, demonstrating how much or how little revision is actually necessary, will always remain a necessary element of educational technology" (Gropper, 1975, p. 7). The evaluation stage of program development must be given "extraordinary importance" (Cruz, 1984, p. 30). The evaluation process examines the instructional product to identify problem areas. In doing so, it also examines the design procedures that have been selected to implement the program.

Frame Development

Klaus (1962) provides 12 programming rules for building frames, which he believes is a critical step in the development process.

1. The frames should <u>require active responses</u> from the learner. This requirement will help avoid lucky guesses and will enable the developer/instructor to evaluate the learner's knowledge of the material (p. 99).
2. <u>Proper cueing</u> should be provided to ensure that the student makes the correct initial response. The amount of cueing should fade as the program develops (p. 100).
3. So that cueing can gradually be reduced, stimuli should be included that will create an <u>appropriate context</u> for the desired response. The intent is to elicit the correct response from the nature of the context rather than from cues (p. 100).
4. The program should proceed in <u>small steps</u> (p. 101), which will aid students' comprehension and success.
5. <u>Careful sequencing</u> of the program is necessary to ensure that the instructional material builds from the simple to the complex (p. 101).
6. Ample provision should be made for <u>frequent repetition</u> of responses. However, it is important to vary the material to reduce the monotony of repetition (p. 101-102).
7. The programmer must have a thorough <u>knowledge of the subject matter</u> to ensure technical accuracy,

i.e., the inclusion and exclusion of material during the design phase. It is important to avoid adding too much material to the program (p. 102).

8. The programmer must be careful to teach, not lecture. The tendency is to present lecture material in programmed form rather than to guide the student in structuring the learning process (p. 102-103).
9. The programmer should carefully evaluate the material that is presented to determine whether it will evoke a relevant response. For instance, if a learner must provide a missing word, the word should be one that will address the concept or fact being taught (p. 104).
10. The material should not provide more cues than necessary. The learner should be able to evaluate the material and make generalizations (p. 104).
11. The programmer should not assume too much knowledge on the part of the learner. However, adequate material must be presented to enable the learner to draw conclusions and provide responses (p. 105).
12. The programmer should not present two new facts within one frame (p. 105). Doing so can inhibit the learning process.

Principles of Learning

Programmed instruction emerged from Skinner's work on psychological principles of learning. These principles must be an integral part of the design process of programmed instruction. Carr (1962) summarized the work of Gilbert and Skinner relating to the five learning principles as follows.

Active Participation

Learning occurs most rapidly if the student is actively involved with the subject matter (Carr, 1962, p. 58). Students learn more from making a response (overt behavior) than from seeing or hearing a response (covert behavior). The student must practice the response to learn and retain it (Klaus, 1962).

Programming experts agree that student responses should be overt, but there is disagreement on the nature of these responses. Pressey advocated the use of multiple-choice or recognition responses, whereas Skinner preferred a construction mode of response (Carr, 1962). An alternative (or compromise) to these two views is to offer a number of response choices and require the learner to spell out a selected answer rather than simply choose a response number (Klaus, 1962).

Practice

Learning is most effective if the skills and knowledge the student gains can be readily generalized to a real-life situation (Carr, 1962, p. 59). Stolurow (1961) stated that the student should work through new material and then be given opportunities to apply the knowledge in different situations requiring the same response.

Feedback

Learning occurs most rapidly if the student is given immediate knowledge of results (Carr, 1962, p. 59). Positive, immediate reinforcement should be provided to the learner's responses. However, Skinner (1954) believed that the use of consistent reinforcement 100% of the time might not provide the most effective condition for retention of responses. Stolurow (1961) suggested it might be more useful to provide a mixed strategy of reinforcement. For example, a 100% reinforcement pattern could be used until the learner reached a predetermined level of correct responses. The program could then follow a partial reinforcement pattern.

Hierarchical Learning

Learning takes place most rapidly if the subject matter is arranged from simple to complex (Carr, 1962, p. 59). The difficulty of the material should gradually be increased. Additional information should be provided to enable the learner to solve problems or answer questions. The material should be provided in small steps to ensure student comprehension and success.

Motivation

Receiving frequent knowledge of results (particularly successes) will motivate the student to continue (Carr, 1962, pp. 59-60). Another way to motivate the student is to vary the techniques used to present the material. Popham and Baker (1971) offer five ways to keep the learner "plugged in" (pp. 199-206). They point out that these are suggested strategies that should be considered; they are not absolutes. The list is not exhaustive, and the programmer should not be limited to these approaches. According to Popham and Baker, the literature is inconclusive on the value of these delivery techniques to the learning process. However, the programmer is encouraged to explore the various approaches.

1. Vary the nature of the instructional stimuli. For example, some material might be presented in anecdotal or story form; other material might utilize graphics, diagrams, etc. Caution is urged, however, that only relevant variations are used. Otherwise, the learning process can be inhibited by too many distractions.
2. Employ humor. It is usually sufficient to use humor on a random, occasional basis. The type of humor used will depend on the learners and the subject matter.

3. Involve the learner in game-type situations. Set up the rules of the game and get the learner involved.
4. Use suspense. A question could be posed early in the program to stimulate the learner's curiosity. The answer to the question would be withheld until later in the program, thus motivating the learner to work through the program to find the answer.
5. Employ format variables effectively. Type fonts and sizes, color, and layout design can be varied to provide visual interest and reduce monotony.
6. Utilize individualized learning. Learning should be designed so that students may proceed at individual rates. Many factors can influence the rate at which students learn: age, personal characteristics, background, and past failures and successes. Individualized instruction allows students to proceed at a pace that suits their learning style.

Summary

Programmed instruction developed as a result of efforts to increase the attention of learners and the retention of instructional material. Over the years, a technology of programmed instruction has evolved that has its roots in psychological learning principles and that follows a systematic instructional design process. The structure of programmed learning, which is particularly suited to the computer technology of today, is experiencing a revival through the efforts of educators to provide for effective computer-assisted instruction.

Computer-Assisted Instruction

"Computer-assisted instruction is the result of the convergence of two technologies: programmed-instruction technology and computer technology" (Bloom, Hastings, & Madaus, 1971, p. 262). The following section provides an overview of computer technology and discusses the evolution of computer-assisted instruction. A description of the different categories of computer-assisted instruction, research comparing computer-assisted instruction with traditional instruction, and techniques for developing and evaluating computer-assisted instruction are also addressed. A summary of the major points concludes the section.

Computer Technology

Although the origins of the computer go back to 1822 to the creation of Charles Babbage's analytical engine, the beginning of the Computer Age is seen as June 14, 1951. This was the date the first Universal Automatic Computer (UNIVAC) was delivered to a client (Capron & Williams, 1984).

The Computer Age is divided into "generations" that are characterized by the predominance of a particular computer technology. The first generation of computers was based on the invention of the vacuum tube and was used for scientific applications. Punched cards and magnetic tape were the principal input and storage devices.

The second generation of computers (the transistor age) began in 1959 and included the expansion of computer applications to business. New high-level languages were created to improve the machine languages of the first generation. During the third generation of computers, which began in 1965 and was based on the integrated circuit, the BASIC language was created, on-line processing became a reality, and microcomputers were introduced.

The fourth generation of computers, which began in 1971, is dominated by microcomputers and microprocessor technology. Filing and storage capabilities for all computer systems have increased tremendously during this generation. New programming languages have been invented that enable nonprogrammers to utilize computer capabilities. With the ready availability of microcomputers, there has been a renewal of interest in the computer as an instructional delivery method.

Evolution of Computer-Assisted Instruction

Computer-assisted instruction began in the mid-1950s as a way to teach binary arithmetic using a drill and practice approach (Michael, 1981). It has its roots in the disciplines of computer science and psychology--a blend of technology with learning theory, instructional strategies, and motivation (Chambers & Sprecher, 1983).

During its formative years, educators prophesied that ". . . CAI would sweep the country and ultimately change the entire structure of education" (Chambers & Sprecher, 1983, p. 6). Instead, computer-assisted instruction has developed slowly and has only recently received renewed attention.

Some of the reasons for the lack of enthusiasm for CAI in the early years included (a) difficulty in developing courseware, (b) lack of training among educators in the use of computers in the classroom, (c) incompatibility of hardware and software, and (d) the expense of developing CAI (Pressman & Rosenbloom, 1984). Lack of quality software has been cited as a major reason CAI has not been readily adopted (Holmes, 1982; Smith & Boyce, 1984; Sprecher & Chambers, 1980; Tyler, 1983).

Early computer-assisted instruction was developed for use with mainframe systems. The expense of using these systems and the inaccessibility of the mainframe to most classrooms further diminished interest in computer-assisted instruction.

The introduction of the microcomputer provided a point of renewal for computer-assisted instruction. The relatively low cost of the microcomputer made it accessible as an instructional tool, and the push to integrate the microcomputer into existing courses resurrected interest in CAI.

While much of the CAI developed for use with the microcomputer has been plagued with the charge that it lacks instructional validity, the body of literature relating to the development of quality CAI is growing. Guidelines for developing CAI using established instructional design strategies and principles of learning abound (Dence, 1980; Smith & Boyce, 1984; Tyler, 1983). The process of formative evaluation has been adapted to the development of CAI materials to produce validated products (Golas, 1983).

The expanding capabilities of the microcomputer, as well as its availability, provide an avenue for growth in the use of CAI materials. The prediction that CAI would change the structure of education may well be realized as the microcomputer gains a central focus in the educational delivery system.

Research on Computer-Assisted Instruction

Research on computer-assisted instruction has addressed several variables such as subject matter, student attitudes, and feedback. Much of the research has focused on comparisons of CAI with traditional methods of study (Sawyer, 1985). Research on comparisons of different types of CAI, retention of material learned via CAI, and cost-effectiveness of CAI is sparse.

The following section discusses CAI research findings dealing with attitudes, achievement, and computer literacy; comparisons of CAI instruction versus traditional instruction; and cost-effectiveness of CAI.

CAI Research on Attitudes, Achievement, and Computer Literacy

Johnson (1974) found that CAI students showed a more positive attitude toward their instructional mode than did students in a programmed learning course. The completion time for the CAI students was also less than that of the programmed instruction group. A study by Hughes (1976) evaluated gains in math skills and attitudes toward the course for students in a CAI course versus those in a lecture-demonstration course. The results showed no difference between the two groups in gains in skill. Attitude scores for the CAI group were not significantly different from those in the traditional course.

A 1981 study by Steele showed no difference between CAI instruction in math and individualized instruction using a math kit. Computer literacy of high- and low-ability students improved significantly after using the CAI materials. D'Onofrio and Slama (1983) also found significant increases in computer literacy for accounting students using a CAI program compared to students in traditional instruction.

Rushinek (1979) conducted an investigation of the relationship between CAI and students' attitudes toward the instruction, the course, and the computer. CAI was found to improve students' attitudes significantly toward the instructor and the computer. Changes in attitudes toward the course were not significant. Lippold (1977) evaluated the relationship of personalization, encouragement, and humor in CAI programs to student attitudes and found that these factors did not significantly affect attitudes.

A study by Rankin (1983) examined the effect of anxiety on performance and its relationship to gender for students using a CAI approach versus a traditional approach. The results revealed that performance for the CAI students was lower than for the traditional students. Females performed better than males in both groups. No significant difference was found regarding the effect of anxiety upon performance. Rankin concluded that the method of delivery interfered with the computer group's performance.

An analysis of student attitudes toward computer-assisted instruction in Nebraska public high schools by Beck (1979) yielded the following conclusions:

1. CAI does not negatively affect student attitudes toward CAI or the course of study.
2. Female students tend to show a more positive attitude toward CAI than do males.
3. Self-oriented students tend to demonstrate a more positive attitude toward CAI than do interaction-oriented or task-oriented students.
4. Student grade point average, time spent at the terminal, and prior experience do not affect student attitude toward CAI.

A study by Hoffman and Waters (1982) on the effects of student personality on success with computer-assisted instruction showed that learning by means of a computer-assisted instructional program seems to benefit individuals who can concentrate quietly, pay attention to details, memorize facts, and remain on task until completion. Magero (1980) conducted a study on the relationship between performance and individual learning style. The results of his study indicated that learning style affects performance and should be considered in selecting instructional design strategies.

Kulik (1983) performed a meta-analysis of 51 studies of CAI and offered the following three conclusions:

1. CAI can improve student learning.
2. Whereas students' attitudes toward the subject matter and quality of instruction were only slightly more favorable with CAI, attitudes toward computers were much more positive after use of CAI.
3. CAI can save a significant amount of student learning time.

Dalgaard and Lewis (1985) surveyed CAI literature to develop a list of generalizations regarding CAI. These generalizations, which parallel the findings by Kulik, include the following:

1. The research generally supports the effectiveness of CAI in improving student achievement.

2. Studies measuring the amount of time spent on instruction showed that a considerable amount of time is saved by using CAI, thus reducing the average training time for students.
3. Although not a strong finding across all studies, students' attitudes toward the subject and quality of instruction tend to be more favorable with CAI.
4. Use of CAI has a strong positive impact on students' attitudes toward computers. Increased motivation, attention span, and class attendance are frequent results.
5. Use of CAI increases computer literacy.

Anelli (1977) studied the relationship between time spent on CAI and reading achievement and attitude. Neither total CAI time spent nor length or frequency of CAI sessions appeared to affect reading achievement. Girls were found to make greater progress than boys in proportion to the time spent. Enthusiasm for CAI decreased after seven to eight hours of accumulated machine time.

CAI Versus Traditional Instruction

A review of CAI studies by Dence (1980) revealed that much of the research comparing CAI with traditional methods of instruction yielded no significant differences between methods. The following section discusses specific research aimed at comparing CAI with other methods of instruction.

Casner (1977) compared the attitudes of math students receiving computer-assisted instruction and those receiving conventional classroom instruction. The study found no significant difference in female attitudes toward math between the two groups. Male CAI students were less scared and less nervous about doing math problems than non-CAI male students.

Lasoff (1981) evaluated the effects of feedback in CAI instruction and programmed instruction on achievement and attitude. He found that regular feedback was significantly superior to enriched feedback or no feedback in promoting achievement. Attitudes toward computer instruction were not significantly affected by the three types of feedback or by the type of instruction provided.

Paden and Barr (1980) investigated achievement and attitudes of students receiving economics instruction in a CAI program and students receiving the instruction in a traditional class. Students receiving CAI instruction showed significant increases in achievement and favorable attitudes toward computer instruction when compared with the control group students.

Daughdrill (1978) compared instruction in college algebra using a CAI approach with a traditional lecture and demonstration approach to determine the effects on achievement. He found no significant difference in achievement between the two groups. Burns (1981) used meta-analysis to evaluate research comparing the effectiveness of CAI versus traditional instruction in mathematics at the elementary and secondary school levels. The findings of the study indicated that drill/practice or tutorial CAI were significantly more effective than traditional instruction in fostering student achievement. Gallitano (1983) examined the effects of a CAI approach versus a traditional approach to teaching trigonometry on student achievement and attitudes. No significant differences in achievement were found between the two groups.

Herbert (1982) studied the use of CAI utilizing tutorial and drill and practice strategies versus lecture instruction in punctuation usage for collegiate business students. The computer-assisted instruction was found to be equally as effective as the traditional instruction. Herbert concluded that CAI offered the benefit of increasing student familiarity with computers; was an enjoyable experience for the students in the study; and, if used outside the class, could free class time for additional instruction.

Cost-Effectiveness of CAI

One area that needs additional research is the cost-effectiveness of CAI compared with traditional methods of instruction. Most of the data related to cost-effectiveness are based on old technology, performance, and costs. In a study by Levin and Meister (1986), the authors concluded that for mathematics and reading courses CAI was relatively cost-effective but not necessarily the most cost-effective instructional strategy. This conclusion was countered in a study by Niemiec, Blackwell, and Walberg (1986), who charged that Levin and Meister underestimated the gains achieved with CAI. In the Niemiec et al. study, CAI was found to be twice as cost-effective as peer tutoring and more cost-effective than adult tutoring, a longer school day, and smaller class sizes.

As equipment costs (hardware and software) decline and labor costs increase, the issue of CAI cost-effectiveness will undergo changes. Ongoing research is needed to resolve the question of cost-effectiveness for computer-assisted instruction.

Summary

Most of the research indicated that CAI is at least as effective as traditional instruction and, in some instances, leads to improved student achievement (Holmes, 1982). Further research must be conducted with CAI to determine for whom and under what conditions it is most effective. Different CAI strategies must be evaluated in terms of locus of control, cognitive style, anxiety level, and personality type to see if relationships exist between these variables and specific CAI strategies (Dence, 1980). Additional human factors research is needed to evaluate interaction with hardware and software. Further study is needed to compare

CAI with other forms of instruction in terms of retention of material and cost-effectiveness.

Categories of Computer-Assisted Instruction

There are seven categories of computer-assisted instruction given in the literature. Several authors cite drill and practice, tutorial, and simulation forms of CAI as the most common ones (Gagne, Wager, & Rojas, 1981; Vargas, 1986). Other authors broaden the list to include gaming, problem solving, inquiry, and dialog (Dence, 1980; Perreault, 1986; Roblyer, 1986; Smith & Boyce, 1984; Splittgerber, 1979).

Drill and practice, tutorial, and simulation are the most common forms of computer-assisted instruction because they are the least expensive and least difficult to program. Gaming, problem solving, inquiry, and dialog require a higher level of interaction with the computer and a concomitant level of programming expertise. These forms of CAI require greater investments of time and cost in programming, debugging, and error trapping than the less sophisticated drill and practice, tutorial, and simulation forms.

Drill and Practice

Drill and practice is the simplest form of computer-assisted instruction because it involves only two instructional events: it elicits a response and provides feedback on results (Gagne, Wager, & Rojas, 1981; Roblyer, 1986). Drill and practice is also the most commonly used form of CAI (Lemaster, 1985).

The purpose of drill and practice is to provide speed and accuracy practice on material that has already been learned (Vargas, 1986). Drill and practice utilizes only one form of learning strategy: rote memorization (Caldwell, 1980). This limitation is both an advantage and a disadvantage. Drill and practice is useful as an alternative to pen and paper homework practice and other situations in which practice on previously learned material is the focus of study. Used in this way, well-designed drill and practice lessons can increase student and teacher productivity and motivation (Roblyer, 1986). The major disadvantage to drill and practice is that its tendency to be repetitive can be boring for students (Perreault, 1986).

Tutorial

The tutorial, another common form of CAI, is designed to teach new material. It may also be used to provide for review or enrichment of material introduced by a teacher (Perreault, 1986). Spitler and Corgan (1979) stated that "true CAI is tutorial" (p. 14) and can stand alone in its presentation of objectives, content, and practice exercises. Gagne, Wager, and Rojas (1981) referred to tutorial instruction as primary instruction rather than supplemental instruction because of its ability to stand alone. Well-designed tutorials utilize a range of instructional strategies. Feedback is provided for correct and incorrect responses. Provision is made for additional study or review if a student's response indicates further study is needed.

Tutorial programs come in two forms: linear and branched. The linear form is the most common (Burke, 1982; Gagne, Wager, & Rojas, 1981). The linear program presents test questions, feedback, and review for incorrect responses. The linear approach is the basis for development of tutorials using an authoring language or system. It is the easiest design to revise and validate (Burke, 1982). Limitations to the linear tutorial design can be overcome through use of branching techniques utilizing a general-purpose programming language.

A more sophisticated tutorial can be developed using branching techniques that permit the student to select alternate learning paths based on responses to the study material. This approach provides a personalized learning experience for the student.

Simulation

A third category of computer-assisted instruction is simulation. The purpose of the simulation is to provide a real-life setting that enables the student to practice skills. The quality of the simulation is determined by the degree to which it imitates real-life situations (Vargas, 1986).

Simulations are based on trial and error learning. The setting is established and the student is expected to initiate the action. One of the primary uses of computer simulations is to allow students to practice skills or conduct experiments that are costly, dangerous, or time-consuming in actual use.

Generally, simulations are used after basic concepts and principles relating to an area of study are taught to develop decision-making skills (Perreault, 1986). Some designers of simulations have added tutorials to provide the requisite background in concepts and principles (Vargas, 1986).

Problem Solving

Computer-assisted instructional programs that address problem-solving skills or exploratory learning are included in the problem-solving category of CAI. Problem-solving programs permit the student to explore fantasy worlds. The goal may be to facilitate discoveries or experience unplanned insights. In either case, the courseware deviates from a systematic approach to teaching (Roblyer, 1986).

Gagne (1977) viewed problem solving as the process of combining previously learned rules into a new higher-order rule that can be generalized to similar problems. This approach provides more structure than emphasizing discovery or insight although elements of insight and discovery will be a part of any problem-solving CAI program.

Gaming

The gaming form of CAI permits a student to play games against the computer or against other students who are involved in the game. Students must demonstrate mastery to win the game (Cohen, 1983).

Aside from the recreational games available for computers, well-written gaming programs can serve as motivational tools in instructional settings. Gaming programs provide additional practice on skills and transfer of learning to other skills (Roblyer, 1986).

Dialog/Inquiry

Dialog. The final two categories of CAI programs are similar in their approach to learning. The dialog (or dialogue) CAI program, which is often called Socratic dialog, utilizes artificial intelligence to "talk" with students. Dialog programs can be very motivational for some students and can create an interest in computers. Dialog does have limitations; it is expensive and time-consuming to produce, and there is no solid evidence of the educational value of this approach (Futrell & Geisert, 1984).

Inquiry. An inquiry CAI program involves a question/answer approach to learning. The program contains the requisite information for solving tasks. The student must query the computer to reach the solution to the problem.

Summary

Most of the CAI programs in commercial use are written in the lower-level forms of drill and practice and tutorial. The time and expense involved in producing and evaluating high-level forms of CAI such as problem solving, dialog, and inquiry limit their availability and also limit research on their instructional value.

Development of Computer-Assisted Instruction

A 1980 study by Sprecher and Chambers of factors affecting courseware development cited lack of quality software as a major hindrance to the use of computer-assisted instruction. Hord (1984) stated that the popularity and availability of microcomputers tempt teachers to prepare lessons for the microcomputer with little consideration of instructional design. Vargas (1986) echoed this finding with the charge that "many CAI programs contain serious instructional flaws" (p. 738). This section outlines procedures for the development of CAI based on learning theory and instructional design strategies. The application of learning theory to the development of computer-assisted instruction is discussed, as well as strategies for evaluation of computer-assisted instruction.

Rowntree (1974) listed four phases of development that are applicable to educational technology: statement of objectives, design of learning, evaluation, and improvement. Each of these phases consists of several intermediate steps.

Statement of Objectives

The steps to be completed during this phase of development include analyzing aims, describing students, specifying objectives, and designing criterion tests.

The initial step in the development process of CAI instructional material is to determine its intended purpose. "Each program should be developed in the context of a curriculum with links to its prerequisites and postrequisites" (Hofstetter, 1985, p. 6). Spitler and Corgan (1979) list four primary goals for using CAI programs: (a) remedial learning, (b) course enrichment, (c) reinforcement, and (d) independent offerings. The goal of the CAI program provides the basis for selection of a design strategy. The purpose of the CAI program also affects the selection of the target audience.

During the objectives phase of development, performance objectives are written. The intended audience for the CAI materials is selected, and entry behaviors for this audience are specified. Exit behaviors are stated in terms of the performance objectives.

The type of criterion performance test to be used and the purpose(s) of the test are determined. The test is developed based on the performance objectives.

Design of Learning

During this phase, the conditions necessary for achieving the objectives are considered. Learning sequences and teaching strategies are determined and the script for the learning materials is developed.

Evaluation

During the evaluation stage the materials are tested by members of the target audience. The results are analyzed and revisions are made if necessary. This process is repeated with a field test of the material.

Improvement

The process of developing instructional materials is dynamic. Analysis and revision of instructional materials should continue during actual use of the materials. Changes in technology may necessitate revision to the materials to provide relevant learning experiences.

Learning Theory and Development of CAI

Each step in the development process is important but the selection of learning strategies and the application of learning theory to the instructional materials are especially critical. Cognitive learning theory, operant conditioning theory, and social learning theory have applicability to particular forms of CAI materials and should be considered in the overall design.

Cognitive Learning Theory

Gagne, Wager, and Rojas (1981) detailed learning events that should be a part of computer-assisted instruction: (a) gain the learner's attention, (b) inform the learner of the lesson's objectives, (c) assess prerequisite knowledge and readiness, (d) present new information, (e) facilitate learning by elaborating on the information, (f) elicit performance, (g) provide feedback, and (h) facilitate retention and transfer. The instructional events build on Gagne's cognitive learning theory and provide a sound basis for development of CAI. Alesandrini (1984) detailed an approach to CAI design based on Gagne's external learning events with added emphasis on the importance of visual as well as verbal presentation of material. Cognitive learning theory is most suitable to the tutorial form of CAI (Chambers & Sprecher, 1983).

Operant Conditioning Theory

Skinner's operant conditioning theory detailing stimulus-response behavior and patterns of reinforcement is another learning theory that should be considered in the development of CAI. Skinner's views are particularly applicable to drill and practice and tutorial forms of CAI (Chambers & Sprecher, 1983).

Social Learning Theory

Although in the past most CAI was developed using drill and practice or tutorial strategies, the lower costs and resultant availability of hardware should increase development of higher levels of CAI such as simulations.

Bandura's social learning theory is most applicable to CAI simulations (Chambers & Sprecher, 1983). Social learning theory combines tenets of cognitive psychology and behavior modification with its own emphasis on the individual in the social setting.

Summary

There is no single design for the development of CAI. Dick (1981) performed an analysis of 40 instructional design models. He noted that the models are similar and that a consensus on the content of instructional designs seems to be emerging. The specific steps in the development process may vary, but the statement of objectives, selection of an instructional design, and selection of procedures for evaluation and improvement are essential.

Vargas (1986) summarized four basic principles of instructional design that are based on learning theory and should be found in high-quality CAI: (a) a high rate of relevant overt responding, (b) appropriate stimulus control, (c) immediate feedback, and (d) successive approximation. Vargas emphasized that these conditions are necessary for learning and that CAI can teach effectively only if these factors are present.

All learning theory may be incorporated to some extent in the various forms of CAI. However, particular learning theories have specific applicability to certain forms of CAI. Cognitive learning theory and operant conditioning are applicable to the two most common forms of CAI: drill and practice and tutorial. These two forms of CAI utilize learning approaches that are at the lowest level of the cognitive process, such as rote memorization. Higher levels of CAI, such as simulation or dialog, are based on higher levels of the cognitive process. Social learning theory, which combines cognitive psychology, behavior modification, and social interaction processes, is an applicable learning theory for the higher levels of CAI.

Evaluation of CAI

Careful attention to the development of CAI must be supplemented by equal attention to evaluation of the materials. Evaluation may apply to commercial software or to CAI materials that are developed in-house.

Elements of Evaluation

To evaluate CAI effectively, an understanding of the purpose of the software is essential. Evaluation must include an assessment of design as well as content. Perreault (1986) gives three main areas to consider in evaluating commercial CAI software: content, instructional quality, and technical quality. The factors to be considered in the evaluation process of commercial software can also serve as a checklist for the evaluation of in-house CAI programs. However, the definition of evaluation becomes broader when applied to in-house instructional materials and often includes the use of formative evaluation procedures.

Formative Evaluation

Assessment of instructional materials most often is accomplished through use of formative evaluation procedures. The product is tested at the draft state on a one-to-one basis to locate and eliminate technical and instructional problems before the material is programmed. Further tests with small groups provide additional feedback on problem areas. The program is then field tested. Data from such sources as a criterion performance test, patterns of student responses to test items, and student responses to interview questions are used to locate problem areas (Gropper, 1975).

Summary

Evaluation of CAI materials should include formative evaluation techniques as well as checklists to locate and resolve problem areas. Whatever the type of evaluation, "continuous evaluation is the hallmark of an effective development effort" (Tyler, 1983, p. 40).

Formative Evaluation

The following section provides a review of the literature relating to formative evaluation. Included in this section is a discussion of different terms and definitions for formative evaluation and for intended audiences. A discussion of the evaluation process and a summary of this section are also included.

Passmore and Asche (1978) state that "educational products . . . should be rigorously examined and, if indicated, improved before they are routinely used with learners" (p. 56). The process of examination and improvement has been called by various names over the years: "developmental testing" by Markle (1967) and Golas (1983); "experimental analysis of instruction" by Hively (cited in Sherron, 1984; Vasek, 1983); "internal evaluation" by Hartley (1972); "instructional validation" by Branstad, Cherniavsky, and Adrion (1980); "tryout and revision" by Abedor (1972); and "empirical tryout" by Gropper, 1975. Scriven (1967) first used the name "formative evaluation" in making a distinction between evaluation conducted during the development of a product (formative) and evaluation conducted after a product is completed (summative).

Evaluation Definitions

The scope of the definition of formative evaluation varies widely. To some it is an evaluation (testing) stage within the total development process (Burke, 1982; Hartley, 1972; Rizza, 1981). To others it is the entire systematic process of planning, developing, and evaluating a product (Lawson, 1973; Passmore & Asche, 1978; Sarapin, 1981; Signer, 1983-84; Tennyson, 1976).

Bloom, Hastings, and Madaus (1971) extend the definition even further. They define formative evaluation as "the use of systematic evaluation in the process of curriculum construction, teaching, and learning for the purpose of improving any of these three processes" (p. 117). Cross (1987) defines formative evaluation as the collection of data "with the intention of improving the educational process while it is in the formative stages" (p. 498). Whether the broader definition or the more restricted one is used, all of the definitions embody the notion of judging or evaluating a product or process to identify and correct areas of weakness.

Evaluation Audiences

Some authors make a definitional distinction between formative and summative evaluation that goes beyond the point where evaluation takes place. Formative evaluation is directed to the developers of products, whereas summative evaluation is of use to the consumers of products (Baker & Alkin, 1973; Cunningham, 1972; Sanders & Cunningham, 1973; Sarapin, 1981).

Evaluation Process

Rizza (1981) specifies six phases in the development process: analysis, design, development, formative evaluation, implementation, and summative evaluation. These phases may be combined into three major stages of product development: planning (analysis and design), development, and evaluation (formative through summative). The following section provides details on each of these stages.

Program Planning

During this stage of development, instructional goals are developed. An instructional task analysis is conducted that becomes the basis for formulation of performance objectives for the program.

The next step is to develop a criterion test to measure student performance against the objectives of the program. The criterion test must be developed during this stage, before the courseware is developed. This step will ensure that the test "align[s] with performance objectives rather than with irrelevant portions of the instructional programs which do not relate to original program objectives" (Vasek & Vogler, 1984, p. 42).

The criterion level(s) for the program must also be established during the planning phase. There is a lack of agreement in the literature on what these levels should be. Burke (1982) suggests that the intended audience, the material, and the conditions under which the program is to be used are factors that must be considered in selecting criterion levels. Vasek (1983) utilized three levels to indicate different degrees of program effectiveness. The most common level from the literature seems to be 90/90: a minimum of 90% of the subjects will score at least 90% on the criterion test. The selection of criterion levels is discussed in greater detail in chapter III.

Program Development

The first task in the development stage is to draft detailed design specifications for the instructional content and sequence. These specifications must then be critically reviewed to identify areas of weakness. According to Grobman (1971), "the most immediate evaluation need of the developmental curriculum process (is) the feedback of information on preliminary materials during the process of material preparation" (p. 439). Sarapin (1981) refers to this process of gathering feedback on the draft materials as prototype evaluation.

Rizza (1981) and Golas (1983) suggest using a student from the target population for a one-to-one evaluation of the handwritten program at this stage. Rizza adds that at least 50% of all errors are identified during the one-to-one evaluation. This step helps eliminate problems before they are built into the program. After revisions have been made, the actual courseware is developed and the evaluation stage begins.

Program Evaluation

Although there is an overlap of formative evaluation through all the stages of development, the bulk of the formative evaluation occurs during the program evaluation stage. The instructional program is subjected to small group tryouts by target group subjects and by content experts during this stage. Golas (1983) suggests that no more than three subjects at a time should be used for the tryout tests.

After each group has completed and evaluated the program, problem areas that have been identified are addressed through a revision of the program content or instructional strategies. This process is repeated for all the tryout groups. Klaus (1962) believes that the tryout cycles should continue until trial subjects can work through the program without error and the criterion objectives are met.

The final step in the evaluation process is to field test the revised instructional program with the target population. The data from the field test are analyzed to identify additional areas for revision and to determine whether the performance objectives for the program have been met.

Summary

Formative evaluation is a systematic process of planning, developing, revising, and field testing to improve instructional materials. Detailed design strategies are prepared to guide the process from beginning to end. Content experts and target group subjects are used to test the instructional materials during the development and evaluation stages. Testing is conducted on the materials until the a priori criterion levels are met. The goal of formative evaluation is to produce an instructional product that can be used without further modification.

Programming Languages and Systems

The instructions that enable a computer to operate and communicate are called programming languages. Programming languages range from low-level (closest to the language the computer uses) to high-level (closest to human language) (Futrell & Geisert, 1984). Languages toward the machine end of the range require less time and energy for the computer to process at the expense of ease of operation. Those at the human end provide greater convenience of operation but expend more time and energy being translated into machine-related symbols.

There are several levels of computer languages including (a) machine languages, (b) assembly languages, (c) high-level languages, (d) application languages, and (e) application systems. In their purest form, machine languages utilize binary digits in strings of 0s and 1s to represent information. Machine languages are called low-level languages. Next in the hierarchy of languages are assembly languages, which use alphabetic abbreviations or mnemonic codes to replace the binary digits of machine language.

High-level languages are English-based, which makes them easier to use than assembly languages. Three of the most common high-level languages are FORTRAN, COBOL, and Pascal. FORmula TRANslation language (FORTRAN) was introduced in 1954 as the first of the high-level languages.

It was written with scientific applications in mind and is still the most widely used language in the scientific community (Capron & Williams, 1984). COBOL (COmmon Business Oriented Language) was introduced in 1959 and is designed for business applications. The Pascal language was introduced in 1971 and combines greater simplicity with more structure. Its most popular use is in computer science programs.

Application languages or systems are more commonly known as authoring languages and systems. The major advantage to authoring languages and systems is that CAI materials tailored to specific teaching/learning needs can be developed in a relatively short period of time and with little knowledge of computer programming (Holmes, 1982). Authoring languages and systems are discussed more fully in a later section.

Programming Approaches

There are three programming approaches that can be used in the development of computer-assisted instructional software: programming languages, authoring languages, or authoring systems.

Programming Languages

In 1965, a high-level general-purpose language called Beginners All-purpose Symbolic Instruction Code (BASIC) was created at Dartmouth College for use in education. It is the most commonly used programming language for the microcomputer although Pascal is gaining in popularity (Capron & Williams, 1984; Gleason, 1981).

BASIC has a fairly simple branching structure, which limits it to programs that are not too involved (Schuyler, 1979). In addition, the BASIC language has not been standardized. For each brand of microcomputer, there is a slightly different version of BASIC (Vasek, 1983).

Although BASIC is easier to learn than the other high-level languages, it still requires considerable programming skill to operate efficiently. Acquisition of this skill may require more time than an individual can devote to the development of instructional materials. However, an alternative is available through the use of authoring tools.

Authoring Languages

To facilitate the development of microcomputer-assisted instructional software and to alleviate the problem of programming difficulties, special-purpose educational application languages called authoring languages have been created. These languages are designed to contain instructions that are helpful to the microcomputer-assisted instruction author or programmer (Schuyler, 1979).

An authoring language is preprogrammed to concentrate on specific instructional tasks. The result is a high-level language devoted to certain instructional sequences. An authoring language is not limited to a particular lesson structure, however (Klass, 1984). The author decides, presumably based on the type of content to be presented, which instructional strategy would be best. The authoring language is then chosen that will best suit this strategy.

Advantages. One of the major advantages of an authoring language over a general-purpose programming language is the inclusion of subroutines that facilitate instructional interaction (Merrill, 1985). No single authoring language has every capability. The language to be used must be determined by the content and instructional strategies to be programmed. Merrill gives the following summary of the subroutines that should be considered in selecting an authoring language.

1. Match. An adequate match routine will do more than just accept exact words or numbers. This routine should (a) provide for detection of spelling errors in responses without judging the response as incorrect, (b) ignore or acknowledge capitalization as instructed, (c) exclude extraneous spaces in responses, (d) enable matching on key words or word strings, (e) enable an AND match on two or more responses in any order as directed, (f) enable an

OR match on one answer or another, and (g) enable AND/OR matches, i.e., match the response meaning if (A AND B) OR (C AND D) are present.

A more sophisticated match routine would allow the computer to judge equivalence of a response to the choices given, such as an algebraic equation that is equivalent to an answer but has not been reduced to its lowest form. The most sophisticated match routine allows the system to judge the adequacy of a student's natural language responses.

2. Accept. This routine is designed to show the user's input on the screen, allow use of the backspace key for making corrections, and wait for the user to strike the Return or Enter key before evaluating the input.

3. Display commands. These features include positioning commands, which provide user control over text movement; color commands, which enable the user to change the background or color of the text; text size and font commands, which enable the user to change sizes and fonts of text; and window commands, which allow the user to display text windows on the screen without disturbing the remaining text.

4. Wait. With this routine, a pause of specified length can be inserted between procedures.

5. Graphics command. This routine provides for the creation of screen graphics by the author.

6. Sound commands. These commands allow the author to create simple sound effects.

7. Link to subroutines in general-purpose language. One of the most important features for an authoring language is the ability to branch out of the authoring language and use a general-purpose language to write subroutines. This feature can be used when an instructional procedure is desired that the authoring language is not equipped to handle.

Disadvantages. The major disadvantage with any authoring language is a decrease in flexibility and speed of execution concomitant with the increase in ease of use. Another disadvantage is the amount of disk space required for storage. The storage problem can be alleviated to some extent through use of a compiler to convert author language codes to machine codes and thus save disk space.

Authoring Systems

An alternative to using an authoring language is to use an authoring system. Burke (1982) describes an authoring system as " . . . basically a software package which guides the . . . author through the programming process and virtually eliminates the need to know how to program the computer or to know a programming language of any kind" (p. 9).

<u>Advantages</u>. A good authoring system can significantly reduce the time necessary to organize content material, enter it into the computer, and test the results for errors (Pogue, 1980). These advantages must be weighed against the lack of flexibility and decrease in execution time that characterize authoring systems. However, authoring systems do provide a quick means of producing courseware for those individuals who do not have the time or expertise to employ a programming language approach. Following are additional advantages in using an authoring system.

1. One of the major advantages of an authoring system is that it minimizes or eliminates programming through the use of a menu-driven format (Merrill, 1985). The author is guided through the instructional process by means of screen prompts and helps. The system uses well-defined formats for organizing the materials to be taught and tested (Young, 1984).
2. An authoring system saves design time through the use of instructional templates. The author simply "fills in the blanks."
3. Use of an authoring system minimizes writing and debugging lines of code (Beebe, 1983). The system is preprogrammed to translate author language to machine language.

4. An authoring system provides for use of a student management program, which enables the student and the teacher to monitor progress.

Disadvantages. There are several disadvantages to using an authoring system. These must be balanced against the convenience and ease of use such a system provides.

1. The author is limited to a predefined instructional strategy (Merrill, 1985). The author is obliged to "pick and choose" among available systems.
2. The ease of operation created through use of natural language commands results in a further decrease in execution time. The system must convert the human language to a machine language before processing can occur.
3. The use of templates severely limits the number and type of instructional strategies that can be used in authoring systems. For instance, the branching technique would not be available for a system using a linear drill and practice approach.
4. The disk storage requirements for an authoring system are even greater than for an authoring language due to the natural language menus that are provided.

Summary

The development of computer languages has been a recent but rapid phenomenon. As the use of the computer (and particularly the microcomputer) has become more widespread, the demand for access to computer capabilities has resulted in the invention of "user-friendly" languages. The latest developments in the computer language field include authoring languages and authoring systems that require a minimum of programming skill to learn and operate. Each approach has several advantages and disadvantages. The prospective user must weigh the drawbacks against the benefits in selecting a programming approach.

Chapter Summary

This chapter reviewed the relevant literature in the areas of programmed instruction, computer-assisted instruction, and formative evaluation. Programming systems and languages were also reviewed.

Programmed Instruction

Programmed instruction evolved as a result of efforts to improve the learning process. It is based on principles of learning that should govern the development of instructional materials. These principles include (a) providing for active participation, (b) allowing practice, (c) providing feedback, (d) ordering the text from simple to complex, and (e) using motivational strategies. Programmed instruction follows a systematic design process that is especially suited to computer-assisted instruction.

Computer-Assisted Instruction

The review of computer-assisted instruction literature revealed a scarcity of validated software for instruction. Available software has often been developed with inadequate attention to instructional design and learning theory. Much of the research has been conducted to compare CAI with traditional instructional methods. Important variables such as comparisons of different CAI strategies, retention of material learned via CAI, and cost-effectiveness of CAI have received little attention. Human factors research has also been a neglected issue.

Lack of quality software has been a major hindrance to the use of CAI. The literature is conclusive in its presentation of guidelines for the development and evaluation of CAI. The development process should provide for (a) statement of objectives, (b) selection of a learning design, and (c) designation of evaluation and improvement strategies. Quality CAI must be based on established learning theories that are suited to the specific form of CAI being used. Evaluation of CAI should assess the design as well as the content in terms of instructional and technical quality. Formative evaluation procedures can be used to improve the instructional materials. The program evaluated through formative evaluation procedures should provide instructional software that needs no additional changes.

Formative Evaluation

Formative evaluation is a method used to improve an educational product or process during its development. It involves program planning, development, evaluation, and revision. Detailed design specifications are used to guide the evaluation process with the intent of producing a product that needs no further modification.

Programming Languages and Systems

Three approaches to programming CAI materials were given in the literature: programming languages, authoring languages, and authoring systems. High-level general-purpose languages, such as BASIC, require a considerable investment in time and cost to learn and use.

To ease the difficulty of programming with a general-purpose language, authoring languages and systems were developed that simplify the codes and processes used to program materials. The result is a computer language that facilitates development of CAI materials and provides a feasible programming alternative for educators interested in the development of instructional materials.

CHAPTER III

METHODOLOGY

The purpose of this study was to develop a validated computer-assisted program to teach word processing terminology using an authoring language. The procedures used to conduct the study are detailed in this chapter. The topics discussed include (a) the research design, (b) the subjects, (c) the data collection, (d) the data analyses, and (e) limitations of the design.

Research Design

This was a descriptive study using formative evaluation procedures to produce a validated computer-assisted program. The program, written with the PILOT authoring language, was based on instructional strategies identified from the literature relating to computer-assisted instruction.

Methodology

Qualitative and quantitative data were used in the study. The quantitative data were collected from a content experts' card sort, an alternative terms selection instrument, and a criterion performance test. Ranked data from the card sort and the alternative terms selection instrument were collected prior to program development. The data from the criterion performance test were collected during the developmental and field testing cycles of program development. The amount of time required for each subject to complete the CAI program was obtained by observation during the testing process. Demographic data were also obtained from each subject.

The qualitative data were gathered from the researcher's observations during the developmental and field testing stages and included (a) difficulties encountered and (b) questions/comments from each subject. Qualitative data were also collected from the interview questionnaire completed by each subject at the end of the tests.

Procedures

Certain decisions were made before the study was undertaken. The researcher (a) identified the content experts for participation in the terminology card sort, (b) limited the number of formative evaluation cycles, and (c) established a priori criterion levels for the criterion performance test.

Content Experts

Six content experts were selected from faculty who teach or have taught word processing at the postsecondary level. The researcher contacted each expert by phone, explained the purpose of the study, and asked the expert to participate. All agreed to participate and were mailed a card sort package and a cover letter explaining the purpose of the study (see Appendices A and B).

Formative Evaluation Cycles

The literature is inconclusive on the number of formative evaluation cycles needed. Dick (1977) noted that the most common approach is to use three cycles. Lange (1967) suggested 2-9 cycles. The researcher conducted five revision cycles: three developmental testing cycles and two field tests.

Criterion Levels

According to Burke (1982), the process of validation must be repeated until "the program meets an acceptable criterion of performance" (p. 102). However, the literature is not clear as to what that criterion level should be. The most common level suggested in the literature is 90/90 (Burke, 1982, p. 102; Futrell & Geisert, 1984, p.105; Hartley, 1972, p. 142); i.e., a minimum of 90% of the students must accomplish at least 90% of the objectives. Burke (1982) stated that the criterion level "must be matched to the students, the material and the conditions under which the lesson will be used" (p. 102).

Vasek (1983) utilized three criterion levels to measure the effectiveness of the instructional unit during the field testing stage of program development (pp. 89-90). Levels of 90/90, 80/80, and 70/70 indicated an acceptable program, a need for minor revisions, and a need for major revisions, respectively. The criterion levels for this study were (a) 90/90 to indicate an acceptable program and (b) 80/80 to indicate that revisions needed to be made.

Program Design

The development of the program was guided by a systematic instructional design, which included the use of formative evaluation procedures to identify and correct problem areas.

Systematic Instructional Design

Figure 1 details the steps dictated by the systematic instructional design used during the development of the program. Cognitive and affective performance objectives were written for the unit and a criterion performance test was developed. The CAI program was constructed from the validated word processing terminology identified by the content experts' card sort using instructional design strategies identified in the literature.

Formative Evaluation Design

Development of the CAI program was governed by the four stages of the formative evaluation model: (a) program planning, (b) program development, (c) program revision, and (d) field testing. The specific problems addressed during each stage are detailed in the following section.

Program planning. The planning stage of the program design included identification of the word processing terms from the literature and development of the survey instruments, cognitive and affective performance objectives, and the criterion performance test. Terms were identified

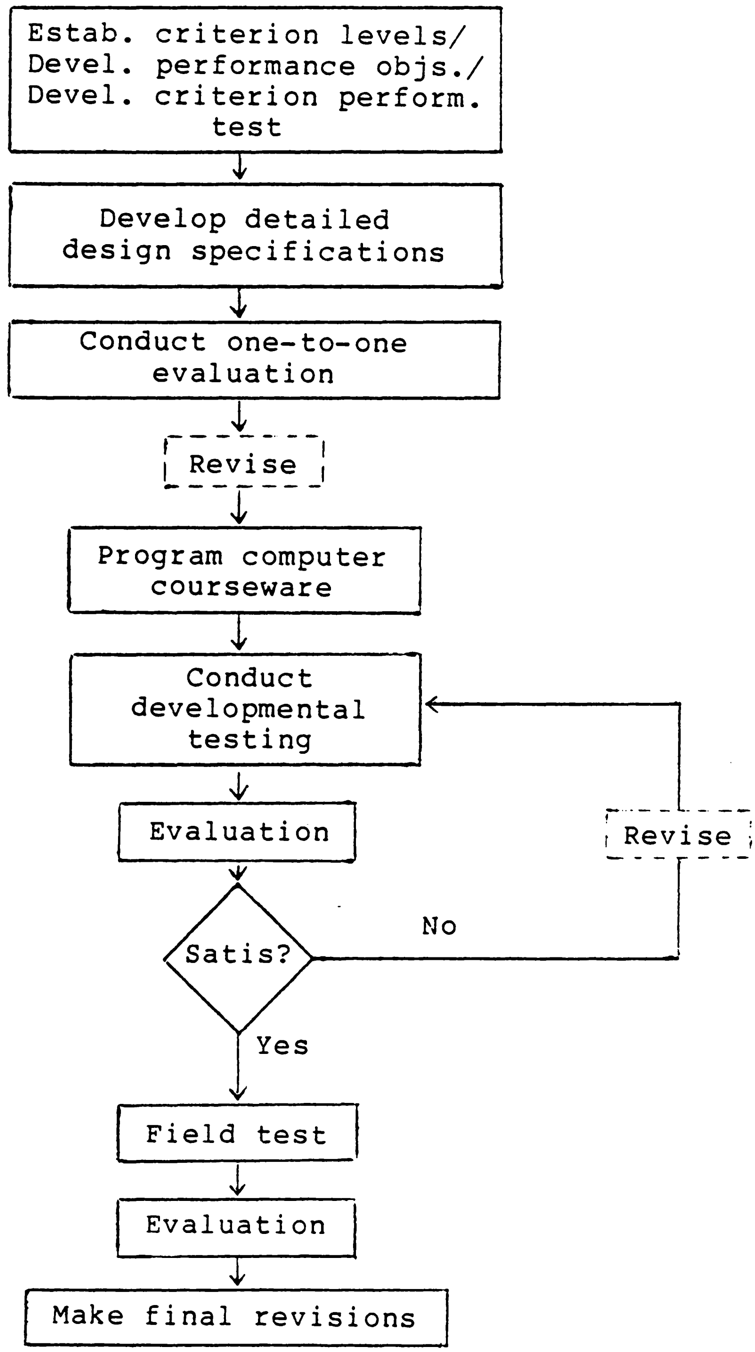


Figure 1. Flow Chart for Instructional Development Process

from textbooks and software documentation and incorporated into a content experts' card sort to validate content. The card sort was distributed, collected, and analyzed during this stage. Cognitive and affective performance objectives were developed for the CAI program. A criterion performance test was developed based on the performance objectives. The choice of names for terms with alternate names was made by the content experts during the planning process. The terms were then sequenced for presentation in the computer program.

Program development. During the program development stage, demographic and structured interview questionnaires were constructed. These instruments were designed to collect subject demographic data and subjects' reactions to the program. The program was designed and flow charted, instructional material was written, and one-to-one and faculty evaluations were conducted.

A flow chart was developed detailing the sequence of presentation for each term, including provision for review. The instructional and criterion frames were written and programmed. The criterion performance test was also programmed. The program was designed to take no more than one hour to complete and included a pretest, a tutorial, and a posttest. The program was tested on a one-to-one basis with a subject representative of the target population. Program changes were made to the instructional content and presentation strategies based on subject input.

The program was tested by two college instructors of word processing and a college employee with programming and word processing experience. Each evaluator recorded problems and comments and was interviewed by the researcher. The data from each evaluator were analyzed to identify problem areas. Revisions to the program were made based on the problems encountered and the evaluators' suggestions for improvement.

Program revision. A formative evaluation design was used for program revision that included (a) developmental testing and (b) field testing. In the developmental phase of testing, the program was evaluated by three groups of subjects (designated D-1, D-2, and D-3). Prior to testing, the purpose of the study and the subjects' role as evaluators were explained to the participants. A demographic questionnaire designed to assess how representative the subjects were of the target population was completed by each subject prior to program testing. The subjects were asked to be specific and thorough in identifying problems during completion of the interview questionnaire at the end of the testing.

As the three subjects in D-1 worked through the instructional program, the researcher observed each subject and recorded any problems encountered, any questions asked, and the amount of time required to complete the program. Each subject was interviewed using a structured interview questionnaire and a tape recorder to record the conversation. Each subject was administered the criterion performance test upon completion of the tutorial portion of the program. The data collected from D-1 (observations, interviews, and test scores) were analyzed to identify problems. The pattern of response choices for the three subjects was also analyzed to identify frames containing questions that seemed to cause problems. Revisions were made to the program.

The testing procedures used with D-1 were repeated with the four subjects in D-2, using the revised program. After analysis of the data from D-2 and additional revisions of the program, the four subjects in D-3 worked through the program. The data from this group were analyzed and revisions were made.

Program field testing. Upon completion of the three developmental cycles the program was field tested with 15 subjects during February 1987. The purpose of the program was explained to the field test group and the demographic questionnaire was administered. Computer keys used to operate the program were identified. The researcher monitored the group but did not offer assistance. The researcher recorded any problems that were noted. The criterion performance test was administered to each field test subject upon completion of the tutorial portion of the program.

As each subject finished, the time required to complete the program was recorded and an interview questionnaire was filled out. The data from observations, interviews, and criterion performance tests were analyzed and revisions were made to the program. The criterion performance test scores were compared to the a priori performance objectives to determine whether the standards were met. A second field test involving 12 subjects was conducted during February 1987 following the same procedures used for the first field test.

Subjects

The target population for this study was community college students with no prior word processing instruction. Volunteers were solicited for both the developmental and field test stages of program development. The subjects used in the developmental testing cycles and the field tests were students at Marshall University Community College, Huntington, West Virginia.

According to Green (1967), between 5 and 10 subjects are needed for developmental testing. Dick (1977) gave 5 to 15 students as an appropriate number. Golas (1983) suggested that no more than three students be used at the same time during the developmental stage. All subjects used for the developmental testing in this study were volunteers with no prior word processing instruction. Volunteers from an introductory word processing class were used for the first developmental testing cycle. Three subjects were randomly selected from the pool of volunteers. The test was conducted on the first day of class before formal instruction began. The remaining volunteers were excluded from later participation in the study since formal word processing instruction had begun.

Subjects for the second and third developmental testing cycles were volunteers from keyboarding and beginning typing classes. Four subjects were randomly selected from the pool of volunteers for the second developmental cycle of testing. Two of the four subjects for the third cycle of testing were randomly selected from the remaining volunteers. The other two subjects volunteered to evaluate the program.

Hartley (1972) stated that a minimum of 15 students should be used for the field test. Dick (1977) increased this number to 20 or more. Twenty-six subjects were used for the field tests in this study. Subjects participating in the first field test were students enrolled in an introductory computer technology course at Marshall University Community College. All students in the course with no prior word processing instruction volunteered to participate. Of the 19 students available, 15 reported to the field test site.

Students in an introductory communications class at Marshall University Community College were used for the second field test. All 22 students in the class began the test, but due to mechanical problems during testing only 9 completed the test. Three of the nine subjects were eliminated. Additional volunteers were sought from a records management class at Marshall University Community College. Six students volunteered to test the program, resulting in 12 students for the second field test.

Data Collection

The following instruments and procedures were used to collect the data for the study.

Content Experts

Six word processing experts were selected to conduct the content experts' card sort. Recommendations from faculty members were considered as a source for selection of the six experts as well as the researcher's knowledge of national leaders in the field of word processing. Each expert was contacted by phone and asked to participate. A cover letter explaining the purpose of the study and a card sort were mailed to each expert. (See Appendices A and B for copies of the cover letter and card sort.)

Content Experts' Card Sort

A list of 60 word processing terms was obtained from word processing textbooks and software documentation. These terms were incorporated into a card sort and mailed to the six content experts. The experts were instructed to indicate the sequence of instruction for the terms by ranking them in four groups of 15 terms, with Group 1 the terms to be taught first and Group 4 the terms to be taught last.

Alternate Names Selection Instrument

An instrument was constructed to select from among a list of alternate names the name of each term to be used in the program. The instrument was mailed to the content experts with instructions to select the most appropriate name for each term. The content experts' choices for names were tallied for each term with alternate names. The terms with the most votes were selected for use in the program. (See Table 2 in chapter IV.)

Demographic Questionnaire

A demographic questionnaire was used to gather data for comparing subject characteristics to the target population (see Appendix C). Data were collected from each subject on age, race, and sex. The subjects also indicated whether they had received previous word processing instruction.

Researcher's Observations

During the developmental and field test cycles of the program development, the researcher observed the subjects and recorded all problems, requests for assistance, and time required to complete the program. Problems were grouped and tallied to aid in the revision process. The time required to complete the program was entered on each subject's interview questionnaire.

Structured Interview Questionnaire

A structured interview questionnaire was used to record any problems and suggestions for improvement from the developmental and field test subjects. Upon completion of the program, each subject completed the interview questionnaire, which consisted of 11 questions. A copy of the questionnaire is included in Appendix D.

Taped Interviews

Conversations during the completion of the interview questionnaire were recorded. These conversations enabled the researcher to verify the data on the questionnaire form and to check for omissions in the data.

Criterion Performance Test

The criterion performance test was administered by the computer to all subjects upon completion of the program. Administration of the criterion performance test to the developmental groups served as a pilot test of the instrument. Any needed revisions were made before the instrument was field tested. A copy of the test is included in Appendix E.

Data Analyses

Data were analyzed to (a) determine the order of the terms in the program, (b) determine which name to use for terms with alternate names, (c) determine subject characteristics, (d) identify problem areas for revision, and (e) determine whether the performance objectives were met.

Content Experts' Card Sort

The responses to the content experts' card sort were categorized according to the preferred sequence of instruction as ranked by the content experts. The 60 terms were ranked in four groups of 15 terms, with Group 1 those terms to be taught first and Group 4 the terms to be taught last. The number of terms assigned to each group was recorded. The terms were ordered from high to low by the researcher, with "high" being the term to be taught first. The 30 highest ranked terms were selected as a starting point for the instructional unit.

Demographic Questionnaire

Data were collected for the categories of age, sex, and race. The category of race on the interview questionnaire was divided into white, black, and other (with "other" to be specified). Sex and age were divided into categories, with age data reported in five categories: under 18, 18-20, 21-30, 31-45, and over 45. The number and percentage of subjects in each category of age, race, and sex were computed.

Researcher's Observations

Observations made by the researcher during program testing cycles were categorized by type and frequency and compared with subject interview data to identify additional areas for revision.

Structured Interview Questionnaires/Taped Interviews

The structured interview questionnaires were analyzed to determine the type and frequency of problems listed. The taped interviews were analyzed to identify additional problems that were omitted from the interview questionnaire form. The responses to the interviews/questionnaires were compared with the researcher's observations to identify areas of the program that needed revision.

Time to Complete

Mean time and range for the amount of time required for completion of the program were recorded for each developmental and field test group. These data were used to determine an appropriate length for the computer program. (See Tables 7, 8, and 9 for lists of the time required to complete the program. Appendix F lists the mean time to complete the program for each group.)

Criterion Performance Test

Scores from the criterion performance test for each developmental and field test group were compared with the a priori performance objectives to see if the objectives had been met.

Limitations of the Research Design

Vasek (1983) listed three limitations that are applicable to the research design of this study (p. 102). First, having the researcher serve as the observer during the testing stages of program development may result in a biased view of subject criticisms. This limitation was minimized by using a structured interview questionnaire.

Vasek suggested the second limitation--that the study subjects may be reluctant to offer criticism of the instructional program--be minimized by explaining the purpose of the study to the learners. The researcher included the explanation of the study purpose as part of the orientation session for subjects and emphasized the role of the subjects as evaluators in the developmental process.

The third limitation identified by Vasek is the vague nature of some of the qualitative data. This problem was addressed in this study by using a structured interview questionnaire to record comments. The subjects were asked to be as specific and as thorough as possible in identifying problem areas.

In addition, Branstad, Cherniavsky, and Adrion (1980) stated that it is difficult for researchers to locate their own errors in a program. It was suggested that a colleague evaluate the program. This study controlled for this problem by utilizing two college teachers of word processing and a college employee with programming and word processing experience to evaluate the program.

CHAPTER IV

RESULTS

The purpose of this study was to develop a validated computer program to teach word processing terminology using an authoring language. Eight research questions were addressed in the study:

1. What word processing terms and definitions are revealed in the literature?
2. What is the preferred sequence of instruction for the terms identified in the literature according to a panel of word processing experts?
3. What are the performance objectives to be met upon completion of the computer program?
4. What are the components of the criterion performance evaluation instrument?
5. What are the steps in the development of a computer-assisted instructional program to teach word processing terminology?
6. How can formative evaluation be utilized to improve a computer-assisted instructional program?
7. How effective was the program in meeting the a priori performance objectives?
8. Is an authoring language an appropriate programming method for this computer program?

The results of the study are detailed in this chapter. The discussion is organized according to the four stages of the formative evaluation model: program planning, development, revision, and field testing. The research questions are discussed in the order in which they were addressed in the study. Preliminary decisions affecting the study are also discussed.

Preliminary Decisions

Prior to the planning stage, several decisions were made that affected the study. Content experts were selected to conduct a card sort of the word processing terms revealed in the literature. The number of formative evaluation cycles proposed was four. (Because an additional field test was needed, the final number of cycles was five.) Criterion levels for the program were established at 80/80 to indicate revisions were needed and 90/90 to indicate an acceptable program. The selection of content experts, number of evaluation cycles, and criterion levels is discussed in detail in chapter III.

Program Planning

The program planning stage included identification of word processing terms from the literature, development of survey instruments, statement of cognitive and affective performance objectives, and development of the criterion performance test.

Word Processing Terms and Definitions

Four research questions were posed during the planning stage. The first research question, _What word processing terms and definitions are revealed in the literature?_, was addressed through a review of textbooks and software documentation. (See Appendix G for a list of the sources used.) Software packages were limited to the two most often used in Virginia community colleges (Hall, 1986).

A list of 168 terms from chapters dealing with document processing was compiled from the sources. The list was edited to eliminate terms that were hardware/software specific, that were listed more than once (using alternate names), and that addressed computer functions in general. A final list of 60 terms remained for use in the study (see Appendix H).

The 60 terms were defined from selected textbooks and software documentation. (See Appendix G for a list of definition sources.) Information from all the sources was included in a composite definition for each term. The definitions were later used in the development of instructional material for the program.

Survey Instruments

A content experts' card sort and an alternative names selection instrument were developed during the planning stage. The card sort was used to determine the preferred sequence of instruction for the word processing terminology.

The names to be used in the program were chosen for those terms with alternate names by the content experts.

Content Experts' Card Sort

A card sort was developed using the 60 terms identified from the literature. (See Appendix B for a copy of the card sort.) The card sort was mailed to six content experts who were selected from among postsecondary teachers of word processing. (See Appendix K for a list of the content experts.) All six experts contacted as first-choice selections for the card sort agreed to participate. However, one did not return the card sort and did not respond to a follow-up letter so another expert was selected. One expert was unable to complete the card sort but selected a secretary experienced in the use of word processing hardware and software to fulfill his commitment. Although the research design called for teachers of word processing to perform the card sort, the researcher felt the addition of the secretary added breadth to the feedback obtained. Therefore, no additional experts were sought. The remaining content experts, all of whom were currently, or had been, teachers of word processing, included college teachers of word processing, authors of word processing and computer textbooks, and heads of departments.

The results of the card sort provided the answer to the second research question, What is the sequence of instruction for the terms identified in the literature

<u>according to a panel of word processing experts</u>? The purpose of the card sort was to rank the terms in the sequence in which they should be taught. The card sort was constructed to rank the 60 terms in four groups of 15 terms each, with Group 1 being the set of terms to be taught first and Group 4 the terms to be taught last. The terms were not ranked within each group by the content experts. Table 1 shows the results of the card sort by group assignment of the terms and by frequency within group.

Some of the terms were placed in the same group by all experts. If fewer than four experts placed a term in a group, the researcher's judgment was used in assigning the placement of that term in the computer program.

Using the results of the card sort as a guide, the 60 terms were sequenced for presentation in the computer program. The process of sequencing the terms was conducted eight times before a final list was obtained. During this process, the terms were kept in the group sequence dictated by the card sort where possible. The researcher ranked the terms within each group according to the logical sequence of instruction. Terms for which no consensus of placement had been reached were sequenced according to the researcher's judgment. The relationship of each term to the other terms was also a factor in designing the instructional sequence of the computer program. The researcher arbitrarily selected the first 30 of the ranked terms as a cutoff for the initial

Table 1

Results of Card Sort by Group Assignment and Frequency Within Group

Term	Group Assignment (1, 2, 3, 4)	Frequency Within Group 1	2	3	4
advance	1,1,1,1,1,1	6	-	-	-
auto carrier return	1,1,1,1,1,1	6	-	-	-
auto centering	3,1,1,2,1,2	3	2	1	-
auto hyphen	3,3,2,3,3,2	-	2	4	-
auto indent	3,3,2,2,2,3	-	3	3	-
auto justification	3,3,2,4,3,3	-	1	4	1
auto underline	3,1,1,2,2,2	2	3	1	-
automatic tab grid	4,1,2,1,1,3	3	1	1	1
block	3,2,3,3,2,2	-	3	3	-
boilerplate	4,4,3,4,4,4	-	-	1	5
constant	4,4,3,4,4,4	-	-	1	5
copy	3,2,3,3,3,2	-	2	4	-
cursor	1,1,1,1,1,1	6	-	-	-
cursor control keys	1,1,1,1,1,1	6	-	-	-
decimal tab	3,3,2,3,4,3	-	1	4	1
delete	1,2,2,2,1,1	3	3	-	-
edit	2,2,2,2,2,2	-	6	-	-
format line	1,2,1,1,1,2	4	2	-	-
format	1,3,2,1,2,3	2	2	2	-
footer	4,4,3,4,4,4	-	-	-	6

Table 1 (cont'd)

Results of Card Sort by Group Assignment and Frequency Within Group

Term	Group Assignment (1, 2, 3, 4)	Frequency Within Group 1	2	3	4
get	4,2,1,4,4,3	1	1	1	3
global search/replace	3,2,4,4,2,3	-	2	2	2
go to	3,2,2,1,3,2	1	3	2	-
hard copy	2,4,1,3,2,2	1	3	1	1
hard return	1,3,1,2,1,2	3	2	1	-
header	4,4,4,4,4,4	-	-	-	6
page end code	2,3,4,2,3,4	-	2	2	2
input	1,1,1,1,1,1	6	-	-	
insert	1,2,2,3,2,1	2	3	1	-
line end zone	4,4,3,2,2,1	1	2	1	2
merge	4,4,3,4,4,4	-	-	1	5
move	3,2,3,3,2,2	-	3	3	-
no adjust	4,4,3,2,4,3	-	1	2	3
original document	1,4,2,1,1,2	3	2	-	1
page	2,3,1,1,3,3	2	1	3	-
page advance	2,1,4,1,3,3	2	1	2	1
page break	2,3,4,2,3,4	-	2	2	2
paginate	2,3,4,3,3,3	-	1	4	1
reformat	2,3,3,3,3,3	-	1	5	-
repaginate	2,3,4,3,3,4	-	1	3	2

Table 1 (cont'd)

Results of Card Sort by Group Assignment and Frequency Within Group

Term	Group Assignment (1, 2, 3, 4)	Frequency Within Group 1	2	3	4
repetitive document	4,4,3,4,4,4	-	-	1	5
required space	2,3,3,2,2,2	-	4	2	-
revised document	2,2,2,2,2,2	-	6	-	-
scroll	1,1,2,1,1,1	5	1	-	-
search	3,2,4,3,4,3	-	1	3	2
search/replace	3,2,4,4,3,3	-	1	3	2
selective search/ replace	3,2,4,4,3,3	-	1	3	2
shell	4,4,3,4,4,4	-	-	1	5
strikeover	2,1,2,2,2,1	2	4	-	-
soft copy	2,3,1,3,2,1	2	2	2	-
spell check	4,4,4,4,3,2	-	1	1	4
status line	1,1,1,1,1,1	6	-	-	-
stop codes	2,1,3,4,4,4	1	1	1	3
super/subscript	3,4,4,3,4,4	-	-	2	4
system page	2,3,4,2,3,4	-	2	2	2
text string	1,1,3,3,2,1	3	1	2	-
undelete	4,2,2,2,1,1	2	3	-	1
variable	4,4,3,4,4,4	-	-	1	5
widow/orphan control	4,4,4,3,4,4	-	-	1	5
word wrap	1,1,1,1,1,1	6	-	-	-

programming. It was decided that the remaining terms could be added if needed.

Alternate Names Selection Instrument

Several of the word processing terms to be used in the study had alternate names. To eliminate researcher bias toward a particular name, a list of terms with alternate names was drawn from the definitions. The list was mailed to the content experts with a cover letter asking them to select from among the choices the term they felt was the most appropriate. The majority choice for each term was used in the computer program. (See Table 2 for a copy of the instrument and results.)

Cognitive Performance Objectives

In response to the third research question, What are the performance objectives to be met upon completion of the computer program?, three cognitive performance objectives were written for the program before development of instructional material began. These objectives were used as guidelines for development of the criterion performance test and the program instructional material.

1. Given a list of word processing terms related to document processing and a list of possible definitions, the student will select the correct definition for each term. Performance will be satisfactory if the student correctly selects 90% of the terms that are defined.

Table 2

Alternate Word Processing Terminology Names Selected by Content Experts

Please circle the term you feel is the most appropriate one to use. If there is an additional term you feel is better than those given, list it in the blank and circle it as your choice.

	Term	Frequency
1.	format line	5
	format scale	-
	ruler	1
	other ______________	-
2.	automatic tab grid	4
	tab grid	1
	standard format line	1
	other ______________	-
3.	word wrap	4
	line end adjust	-
	line adjust	-
	word wraparound	1
	soft return	-
	adjust	-
	auto return	1
	automatic carrier return	-
	other ______________	-
4.	hot zone	1
	line adjust zone	1
	line-end zone	3
	adjust zone	1
	other ______________	-
5.	hard return	5
	coded return	1
	other ______________	-
6.	automatic underline	1
	auto underline	3
	automatic underscore	1
	auto underscore	1
	other ______________	-

Table 2 (cont'd)

Alternate Word Processing Terminology Names Selected by Content Experts

Please circle the term you feel is the most appropriate one to use. If there is an additional term you feel is better than those given, list it in the blank and circle it as your choice.

	Term	Frequency
7.	typeover	2
	replace	1
	other strikeover	3
8.	required space	5
	coded space	1
	other ______	-
9.	text string	4
	character string	1
	other string	1
10.	soft page	1
	temporary page	1
	system page	2
	temporary page ending	1
	other page ending	1
11.	hard page	-
	required page break	1
	hard page break	-
	new page command	1
	required page end code	-
	new page marker	-
	page break	2
	force page	-
	advance page	-
	page advance	-
	other forced page	1
	coded page	1
12.	widow/orphan protect	1
	widow/orphan adjust	1
	widow/orphan control	4
	other ______*	-

Table 2 (cont'd)

Alternate Word Processing Terminology Names Selected by Content Experts

Please circle the term you feel is the most appropriate one to use. If there is an additional term you feel is better than those given, list it in the blank and circle it as your choice.

	Term	Frequency
13.	hard copy	3
	printout	2
	other __output__	1
14.	reformat**	6
	rewrite	-
	other ______	-
15.	automatic indent	1
	block indent	1
	auto indent	3
	other __1 or 3__	1
16.	search	4
	find	2
	other ______	-
17.	discretionary search and replace	1
	selective search and replace	5
	other ______	-
18.	spell check	5
	spelling verification	1
	other	-

* No selection made

2. Given a list of definitions of word processing terms related to document processing, the student will select from a group of terms the term being defined. Performance will be considered satisfactory if the student correctly selects 90% of the terms that are defined.

3. Given a list of word processing definitions related to document processing, the student will type the name of the term that is defined. Performance will be considered satisfactory if the student types 90% of the names of the terms that are defined.

Affective Performance Objective

In addition to the three cognitive performance objectives, an affective objective was written for the computer program. This objective was as follows:

> At the conclusion of the word processing program, at least half of the learners will respond positively to a single questionnaire item that asks, "Would you like to learn or review word processing terminology by this method?"

The affective objective question was included in the structured interview questionnaire. The results of this question are discussed later.

Criterion Performance Test

The fourth research question addressed during the planning stage was What are the components of the criterion performance test? The criterion performance test was developed from the performance objectives and the list of terms to be used in the program. The criterion performance test was used to measure attainment of the performance objectives. Test questions were developed to address each of the performance objectives. The original version of the test contained 30 questions; the final version contained 10 questions. Administration of the criterion performance test during the developmental testing cycles served as a pilot test of the instrument. Minor changes to the instrument were necessary during the evaluation process. One question that was frequently missed was rewritten. Other questions were deleted as terms were removed from the program.

The criterion performance test was used as a pretest to select study subjects. Only those subjects scoring below 60% on the pretest completed the program. Subjects scoring above 60% were exited from the program by the computer. The pretest and posttest were programmed to compute the number and percentage of questions answered correctly. This information was provided to the students by the computer upon completion of each test.

The criterion performance test was composed of multiple-choice questions written as terms for which the subject selected definitions, definitions for which the subject selected terms, and fill-in questions for which the subject typed an answer. The results of the criterion performance test are discussed later. A copy of the final version of the criterion performance test is included in Appendix E.

Program Development

During the program development stage, the demographic and structured interview questionnaires were developed, the program was designed and flow charted, instructional material was written, and one-to-one and faculty evaluations were conducted.

Demographic Questionnaire

A demographic questionnaire was used to compare the study subjects to the population of community college students at Marshall University. The questionnaire was administered to each developmental and field test subject during the orientation session prior to testing. Data were collected on age, race, and sex. The demographic profile of the study subjects is shown in Table 3. Thirty-eight subjects were used: 11 in the developmental tryouts, 15 in Field Test A, and 12 in Field Test B. Of the 38 subjects, 66% were females and 34% were males. The subjects ranged in age from 18 to over 45, with 50% between 18 and 20, 29% between 21 and 30, and 18% between 31 and 45. Eighty-nine percent of the subjects were white and 5% were black.

Table 3

Subject Demographic Profile

Category	Developmental n	Developmental %	Field Test A n	Field Test A %	Field Test B n	Field Test B %	Total n	Total %
Sex								
Male	3	27	6	40	4	33	13	34
Female	8	73	9	60	8	67	25	66
Race								
White	11	100	11	73	12	100	34	89
Black	--	--	2	13	--	--	2	5
Other	--	--	2	13	--	--	2	5
Age								
< 18	--	--	--	--	--	--	--	--
18-20	5	45	7	47	7	64	19	51
21-30	2	18	5	33	3	27	10	27
31-45	4	36	2	13	1	9	7	19
> 45	--	--	1	7	--	--	1	3

The community college student population at Marshall University is composed of 66% females and 34% males. Thirty percent of the students are between 18 and 20, 40% are between 21 and 30, and 24% are between 31 and 45. Ninety-four percent are white and 5% are black (Office of Institutional Research, 1987). The sample closely parallels Marshall University Community College students when compared on sex and race. The two groups differ somewhat on age, with fewer of the sample between 21 and 30 and more between 18 and 20 than in the population.

Structured Interview Questionnaire

The interview questionnaire was structured to elicit comments on the strengths of the computer program, student opinions of the program, and problem areas. (See Appendix J for a list of program strengths.) Responses to the interview questionnaire revealed areas of the program where revisions were needed. Tables 4, 5, and 6 summarize the problems identified during the revision process. Specific problems and corrective actions are discussed later in this chapter.

The interview questionnaire also addressed the affective objective:

> At the conclusion of the word processing program, at least half of the learners will respond positively to a single questionnaire item that asks, "Would you like to learn or review word processing terminology by this method?"

Table 4

Summary of Problems Identified From Developmental Groups D-1, D-2, and D-3

Item	Frequency
Need review loops on missed questions	4
Got boring after a while	3
Misunderstood instructions	3
Too long	2
Couldn't go back if mistakes were made	2
Had problems with fill-ins	2
Need more reinforcement	2
Didn't understand document	1
Couldn't remember correct words (terms)	1
Need more than multiple-choice questions on test	1
Need to be able to review terms for reinforcement	1
Need fewer terms with more instructional material	1
Found typographical error	1
Need to be able to delete something	1

Table 5

Summary of Problems Identified From Field Test A

Item	Frequency
Need more time to learn, too much to learn in short time (too long)	2
Program had an incorrect answer on one loop	1

Table 6

Summary of Problems Identified From Field Test B

Item	Frequency
Too long	4
Didn't like having to push RETURN to enter answer and then again to continue with program	1
A little confusing with all the terms	1
Could get boring after a while	1
Need to be able to go over it 2-3 times	1

Of the 36 subjects who completed the program, 35 responded _yes_ to this question. Seven of the subjects who answered _yes_ qualified their responses as follows: (a) yes, if I could take my time and really study but I would rather have a teacher present; (b) very much so; (c) yes, it's much better than the keyboarding program; (d) probably; (e) not for a whole semester as it could get boring; (f) yes, you could go at your own pace; and (g) sure.

Program Design and Flow Chart

The fifth research question, _What are the steps in the development of a computer-assisted instructional program to teach word processing terminology?_, was addressed during the program development stage. The instructional program was designed with a pretest, a tutorial, and a posttest. The program was designed with an autoboot feature to load automatically when the computer was turned on. The first three frames, which were self-loading, gave the title and identification of the program and requested the student's name. When RETURN was pressed after the name frame, a record file was activated on the computer program to store all student responses, the number and percent of correct answers on the pretest and the posttest, and the amount of time required to complete the program. If more than one person used the same program disk, a separate record file was activated for each person.

The sequenced word processing terms obtained from the card sort were used as a guide to the flow charting process for the instructional material. A detailed flow chart was developed for the 30 terms selected as the initial program and was changed after each testing cycle to reflect needed revisions. (See Appendix K for the final program flow chart.)

Program Instructional Material

The program instructional material was developed in a tutorial format. Instructional, response, and criterion frames were developed for each term. Practice tests were used to provide reinforcement and review. Color was used to emphasize the presentation of each term. The instructional material was programmed in a linear format with provision for review.

Tutorial design. The instructional content of the tutorial was written using the flow chart and the word processing definitions. A storyboard was constructed for the frame script and visuals. Illustrations were included for each simulation/graphics frame.

The tutorial, which was self-paced, consisted of instructional frames that introduced and defined the terms; response frames, which followed the initial presentation of each term and required a typed response from the subject; simulation and graphics frames to allow the subject to see the function represented by the term and/or to practice the action represented by the term; and criterion frames, which provided a test question following each term. The tutorial was divided into blocks of five terms, with a practice test following each block. The final version of the program included a review frame before each practice test.

Instructional and response frames. An instructional frame introduced each term and was followed by a fill-in response frame that required the subject to type the name of the term before continuing. If the response was incorrect or no response was typed, the correct response was then provided and the subject was instructed to type it. The response frame was programmed with an infinite loop, requiring the subject to type the correct response before the program would continue. Additional instructional, graphics, and/or simulation frames followed the response frame for each term.

Criterion frames. The last frame for each term was a criterion frame. The criterion frames were written from the instructional objectives prior to the development of the instructional material. There were four response choices for each multiple-choice criterion frame question. Feedback was provided for both correct and incorrect responses. The subject could answer incorrectly up to three times and would be looped to the beginning of the instructional material for that term. If an incorrect response was chosen on the fourth attempt, the correct response was given and the subject was instructed to continue to the next term.

Practice Tests. After each block of five terms, a practice test was provided that required a fill-in response for each question. If a question was answered incorrectly, the correct answer was presented and the subject was looped to a brief review of the term (usually no more than one frame of information) and then given the question again. This section was also programmed with an infinite loop, forcing the subject to type the correct answer before the program would continue.

The final version of the program had a review frame before each practice test. A list of the five terms covered in that block, along with a brief one- to two-line definition for each term, was given. Following the last practice test, the tutorial concluded with instructions for the posttest.

Color. Throughout the program, color was used to make the program visually appealing as well as to enhance the presentation of content. Blue was used as a background screen color, with regular text in yellow, terms highlighted in white, and commands/reinforcement messages in yellow. Simulation and graphics screens used white and yellow on blue, with red used to introduce or highlight specific aspects of a term's definition.

Instructional strategies. The computer program in this study was developed in a linear format with loops to review material. The instructional material presented in the program requires low-order cognitive reasoning and lends itself to presentation and review. Simulations and graphics were needed to define some of the terminology adequately. Therefore, a combination of tutorial, drill and practice, and simulation CAI strategies was used to present the instructional material.

Programming. A programmer proficient in the use of the PILOT authoring language programmed the instructional material and edited the program during the revision process. The researcher edited the program from the third developmental testing cycle to the end of testing.

One-to-One Evaluation

During the preliminary stages of development, the program was tested by a volunteer representative of the target population. Before developmental testing cycles began, the program was also evaluated by two college instructors of word processing and a college employee with programming and word processing experience. Additional revisions were made to the program based on the comments and suggestions for change from these evaluators.

The one-to-one evaluation of the program was conducted after the initial programming of the instructional material. Rizza (1981) and Golas (1983) recommended that a one-to-one evaluation be conducted with the handwritten materials prior to programming. Due to the number of simulations and graphics in the program in this study, it was decided that a better assessment of the program could be made if the evaluator could use the programmed materials. The one-to-one evaluation proved to be the most extensive and most valuable in terms of feedback. Several major problems were identified and corrected during this evaluation.

The literature suggests using an individual representative of the target population for the one-to-one evaluation. The researcher deviated somewhat from this recommendation. The individual who agreed to evaluate the program was typical of the population in age, sex, race, and absence of formal word processing instruction. The evaluator, however, had skill in computer programming and had learned word processing through self-instruction but did not know the correct terms for word processing functions. The evaluator's skill in programming enabled her to identify programming difficulties and inconsistencies early in the revision process. The following section discusses the problems that were identified during the one-to-one evaluation and the corrective action taken.

1. Problem: Once a question response choice (a, b, c, or d) was typed, the program accepted the response and automatically moved to the next frame. There was no opportunity to correct an error or select another response choice.

Action taken: The in-key command was removed and the program was rewritten to require the subject to press RETURN to proceed to the next frame.

2. Problem: The response choices for the criterion frames were all names of terms. Once the subject noticed this pattern, the correct answer was selected without reading the question.

Action taken: Some criterion frames were rewritten so that the response choices were definitions of terms, requiring the subject to read to find the correct answer. In addition, the questions used for criterion frames were rearranged so that questions with a similar response pattern were not in sequence.

3. Problem: Inconsistencies existed in text arrangement (horizontal and vertical spacing, indentations) that were not esthetically pleasing.

Action taken: Changes were made to correct inconsistencies throughout the program.

4. Problem: There were too many "a" answers (often several in a row) on the criterion frames and pretest/posttest questions.

Action taken: Response choices were changed to eliminate the high number of "a" answers. The question/answer frames were rearranged to avoid a pattern of the same response letter in sequence.

5. Problem: The program did not flow well in places and was too brief in other places.

Action taken: Additional instructional material was added to some of the frames; other material was written to make the information flow more smoothly.

6. Problem: The only possible response choices were names or definitions of terms. The subject was forced to select an answer from among these choices even if the correct answer was not known.

Action taken: "Don't know" was added as a response choice for all criterion frame questions and multiple choice pretest/posttest questions.

7. Problem: The symbol for RETURN was not identified as RETURN anywhere in the program.

Action taken: The RETURN symbol was included as part of the instructions given at the beginning of the program.

8. Problem: Use of the word return was inconsistent. In some places it was RETURN; in others, Return.

Action taken: All occurrences of the word return that were prompts to the subject were programmed in all caps.

9. Problem: The first two frames, which introduced the tutorial and gave a brief explanation of the contents of the program, were self-loading and did not remain on the screen long enough for the subject to read them.

Action taken: The wait-time instruction was increased to allow more time between these frames.

10. Problem: The pretest and posttest scores extended several decimal places.

Action taken: The program was rewritten to round each score to a whole number.

11. Problem: The evaluator did not like the researcher closely observing the evaluation process.

Action taken: Care was taken during the developmental testing cycles and field tests to appear to be working on something else in an effort to put the subjects at ease.

12. Problem: Typographical errors were found in the instructional material.

Action taken: The program was carefully proofread both on screen and on hard copy to correct any typographical, grammatical, and format errors.

13. Problem: Three frames contained instructions that were not clear.

Action taken: The instructions were rewritten.

14. Problem: There were too many terms for the subject to remember. Action taken: Practice tests were built into the program after every five terms.

15. Problem: A suggestion was made that a frame with a list of the terms should be given for review before the subject was tested on the terms.

Action taken: The researcher elected to proceed with the developmental testing cycles before making a decision on this suggestion. This frame was finally added before the second field test.

Faculty Evaluation

Two college teachers of word processing and a college employee with programming and word processing experience also evaluated the program. Each evaluator worked through the program independently and recorded any problems. Each was then interviewed by the researcher to discuss the problems encountered and the suggestions for improvement. Based on the evaluators' input, the following changes were made to the program:

1. Problem: There was not enough practice material; more simulations/graphics were needed.

Action taken: Simulations and/or graphics were added to the program. (Nine of the 10 terms remaining in the final version of the program had simulations or graphics.)

2. Problem: A transition frame was needed after the tutorial and before the posttest.

Action taken: A frame was added that congratulated the subject on having finished the tutorial and instructed the subject to continue to the posttest.

Program Revision

The sixth research question, How can formative evaluation be utilized to improve a computer-assisted instructional program?, was addressed during the program revision and field testing stages. The first developmental group (D-1) that tested the program was composed of three volunteers. The two remaining groups (D-2 and D-3) had four subjects in each. Data from each developmental group were analyzed to identify problem areas. Revisions were made to the program prior to the next testing cycle.

Developmental Testing Cycles

After revisions had been made following the one-to-one and faculty evaluations, the first developmental group (D-1) worked through the program, followed by the second and third developmental groups (D-2 and D-3). The three subjects in D-1 were tested in the microcomputer classroom at Marshall University. Prior to testing, the researcher discussed the purpose of the testing procedure with the group and identified the computer keys necessary to operate the program. Each subject completed the demographic questionnaire. Two of the subjects were tested together; the other was tested at a later time.

The researcher observed the three subjects as they worked through the program and recorded any problems noted. Each subject completed an interview questionnaire at the end of the testing and also completed a taped interview with the researcher. The comments from the questionnaire and the taped interviews were used to identify problem areas in the computer program. The computer record file for each subject was analyzed to note any problem areas. The subjects' responses were compared to determine any patterns of incorrect responses. A discussion of the problems encountered with D-1 and the corrective action taken follows.

Developmental Group 1 (D-1). The revisions made during the developmental testing were most extensive after the first testing cycle. The program was long (30 terms) and there was not enough reinforcement. None of the three subjects in D-1 reached the minimum criterion level. Analysis of the computer record files for this group indicated that all subjects were missing most of the fill-in questions on the practice tests and posttest. Comments from the subjects also indicated problems with the fill-in questions. The specific problems identified during this testing cycle and the corrective action taken are discussed in the following section.

1. Problem: The program gave only the number of correct answers on the pretest; it did not give this number as a percentage of the total number of possible answers.

Action taken: The program was rewritten to give the subject the number of questions answered correctly and the percentage of correct answers.

2. Problem: On the pretest, subjects typed out "don't know" as a response choice for questions rather than selecting "d," ("don't know") or pressing RETURN for fill-in questions. One subject asked what to do for "don't know."

Action taken: The instructions were rewritten for both the pretest and posttest to indicate clearly the desired response behavior.

3. Problem: One subject asked for instructions on how to continue after the pretest.

<u>Action taken</u>: The instruction "Press RETURN to continue" was added to the frame following the pretest. (Previously, only the RETURN symbol was on the screen.)

4. <u>Problem</u>: The program gave feedback on the practice test questions that indicated only whether the response was correct or incorrect, and then continued to the next question. The subject could not go back if the question was missed.

<u>Action taken</u>: The practice tests were rewritten to give the subject immediate review on an incorrect response. For incorrect responses, the subject was instructed to press RETURN to review the term; the practice question was then repeated after the review frame(s). The program was changed to an infinite loop on the practice tests so that the subject could not continue until a correct response was given.

5. <u>Problem</u>: Subjects indicated they had difficulty remembering the terms, especially for fill-in questions. (A high percentage of fill-in questions were missed.)

<u>Action taken</u>: A fill-in response frame was added immediately following the introduction of each term. The subject was required to type the term before continuing.

6. <u>Problem</u>: Subjects indicated the program was too long in terms of the time required to complete it and the number of terms to remember.

Action taken: Ten terms were removed to reduce the length of the program to 20 terms.

7. Problem: The term document was not defined anywhere in the program and was not understood by one subject.

Action taken: Document was defined at the beginning of the program when the term was initially introduced.

8. Problem: Instructional material for one of the terms was unclear.

Action taken: This term was among the 10 terms that were removed from the program.

9. Problem: Some of the terms did not have enough instructional material.

Action taken: Additional material was added to these terms.

10. Problem: The program did not flow well in places.

Action taken: Instructional material was modified or rewritten to ensure that each new term related to the previous term. This change improved the cohesion of the program and provided additional reinforcement of terms.

11. Problem: Analysis of response patterns for the D-2 subjects showed that the questions for automatic tab grid were frequently missed.

Action taken: The researcher elected to proceed with the D-3 testing cycle before making a change in this question.

12. In addition to the above revisions, graphics for two of the terms were changed and graphics were added to three other terms to provide further reinforcement.

Developmental Group 2 (D-2). The four subjects in the D-2 tryout were tested in the Community College's microcomputer lab at Marshall University. The same procedures were followed as those used with D-1. Positive remarks about the practice tests showed that the review material and the chance to correct mistakes provided reinforcement. Problems encountered with D-2 are detailed as follows.

1. Problem: Two of the subjects seemed to get bored or restless and indicated that the program was too long. (Although 10 terms had been removed, the addition of the response frames, practice tests, and additional instructional material had lengthened the program considerably.)

Action taken: A relax frame was added after 10 terms to provide a break.

2. Problem: A typographical error was noted by one of the subjects.

Action taken: The error was corrected.

Developmental Group 3 (D-3). After the revisions suggested by the D-2 subjects were made, the program was evaluated by the D-3 group. Two of the four D-3 subjects were tested in the microcomputer lab at Marshall University

Community College. The other two subjects were tested in a closed lab with only the subjects and the researcher present. The following problems were encountered.

1. Problem: Subjects were consistently missing the questions on automatic tab grid.

Action taken: The questions were rewritten to use tab grid rather than automatic tab grid.

2. Problem: Two subjects had trouble locating the SHIFT and DELETE keys.

Action taken: The location of these keys was pointed out during the orientation session at the beginning of the field test.

3. Problem: Subjects with little or no typing experience had problems manipulating the shift keys.

Action taken: Text was rewritten for one frame that required subjects to type several capitalized words.

4. Problem: Although none of the D-3 subjects indicated that the program was too long, the tutorial portion of the program was taking an average of 33 minutes to complete after the D-3 testing cycle. The entire program was taking an average of 46 minutes to complete. The researcher felt the length of the program would cause problems for some subjects during the 50-minute class hour available for the field test. In addition, all four D-3 subjects met the 80/80 criterion level but only two mastered 90% of the terms.

Action taken: Five terms were removed from the program, leaving 15 terms. None of the subjects used the relax frame so it was removed also.

Volunteer Testing

After the problems identified by the D-3 group were corrected, four volunteers were asked to evaluate the program before it was field tested to see how long the tutorial portion would take to complete. All four of the volunteers agreed to work through the entire program. One volunteer exited the program after the pretest. The average time for completion of the tutorial was 25 minutes with a range from 20 minutes to 32 minutes. The average completion time for the entire program for the remaining three volunteers was 36 minutes. Although the tutorial portion of the test was still long, the decision was made to administer the program under field test conditions before making any further changes to the program.

Field Testing

Two field tests were conducted during the study. The success criterion level was not reached for the first field test and the subjects indicated that the program was too long. After changes were made to the program, a second field test was conducted.

Field Test A

The subjects for Field Test A were students in a beginning computer class at Marshall University Community College. They were tested in the Marshall University Community College's microcomputer classroom. Fifteen students participated in the test. The orientation procedures used were identical to those used for the developmental groups with the exception of additional instruction on the location of keys necessary to operate the program.

One subject scored above 60% on the pretest and exited the program. The remaining 14 subjects worked through the program. Four students took 50 minutes or more and were concerned about missing the next class. They were encouraged to finish the program and all agreed to do so. The problems encountered during the first field test are discussed in the following section.

1. Problem: One subject pointed out a response error on the loop of one criterion frame question.

Action taken: The error was corrected.

2. Problem: Several subjects were not able to complete the program within one class period and were concerned about being late for the next class period. Several indicated the program was too long.

Action taken: Five terms were removed from the program, leaving 10 terms. A review frame was added before each practice test, listing the five terms just studied and giving a brief explanation of each. A review frame of all the terms was added before the posttest.

Field Test B

Because the criterion level of 90/90 was not reached with Field Test A and length of the program was felt to be a factor, a second field test with the shortened version of the program was conducted. Students in an introductory communications class at Marshall University Community College were used for this field test. To simplify the testing process, all students in the class were tested. Those students with prior word processing instruction were to be eliminated after the testing was completed. Mechanical problems at the beginning of the testing procedure forced the elimination of several students. Of the 9 students who finished the program, only 6 were suitable as study subjects.

Additional volunteers were sought from a records management class at Marshall University Community College. Six students agreed to test the program. Twelve subjects were used in the second field test. One subject scored above 60% on the pretest and exited the program. Of the 11 subjects who completed the program, 10 met the 90/90 criterion level. The following problems were noted.

1. Problem: Several subjects indicated the program was too long, was boring, and was confusing with all the terms.

Action taken: None. These comments came from subjects who were tested under adverse conditions. They were enthusiastic initially about testing the program, but the mechanical problems experienced required the subjects to "wait their turn" to take the test, which seemed to diminish their eagerness to participate.

2. Problem: One subject indicated that the user needed to be able to go over the material two or three times.

Action taken: None. The subject apparently had not followed instructions as the material could be reviewed repeatedly.

3. Problem: One subject did not like having to push RETURN to enter an answer and then again to continue with the program.

Action taken: None. No other comments were made regarding this action. It was felt that one comment did not warrant making a change to the program design.

Results of the Criterion Performance Test

The seventh research question, How effective was the program in meeting the a priori performance objectives?, was addressed by an analysis of the data obtained during the revision and field testing stages. The scores for the three D-1 subjects were below the 80/80 criterion level. Three of the D-2 subjects met the 80/80 criterion level, and two of these met the 90/90 criterion level. All four D-3 subjects met the 80/80 criterion level, with two of these meeting the 90/90 level. Table 7 shows the scores on the criterion performance test for the three developmental groups.

Ten of the 14 subjects who completed Field Test A met the 80/80 criterion level, with 7 of the 14 subjects meeting the 90/90 level. For Field Test B, the 11 subjects who completed the test met the 80/80 level, with 10 of the 11 meeting the 90/90 level. Subjects' scores on the criterion performance test for the field test groups are shown in Tables 8 and 9. Table 10 shows the number of subjects in each test group who met the criterion levels.

The performance objectives for the program stated that a minimum of 90% of the subjects must correctly answer at least 90% of the questions on the criterion performance test. The results of the criterion performance test for Field Test B revealed that the program met the a priori criterion level of 90%.

Authoring Language

The ninth research question, Is an authoring language an appropriate programming method for the computer program in this study?, was addressed through a review of the literature, discussion with a programming expert, and program development and use.

The PILOT authoring language (IBM-PC version) was chosen for this study because of its capabilities in color, graphics, and the ease of writing and editing. PILOT proved

Table 7

Program Completion Times and Ranked Scores on Criterion Performance Test--Developmental Testing Groups

Subject	Time (in minutes)	Number Correct	Percent Correct
D-1 (30 terms)			
2*	49	23	77 a
3	55	18	60 a
1	59	9	30 a
D-2 (20 terms)			
4	40	20	100
6	41	19	95
5	47	17	85 b
7	50	15	75 a
D-3 (20 terms)			
8	41	20	100
11	43	18	90
10	45	17	85 b
9	54	16	80 b

* Assigned subject number

a Did not meet criterion level

b Met 80/80 revision criterion level but not 90/90 success criterion level

Table 8

Program Completion Times and Ranked Scores on Criterion Performance Test--Field Test A

Subject	Time (in minutes)	Number Correct	Percent Correct
12*	--	--	-- a
25	29	15	100
24	27	15	100
14	31	15	100
13	51	15	100
21	32	14	93
18	50	14	93
16	34	14	93
22	49	13	86 c
19	24	12	80 c
15	51	12	80 c
26	37	11	73 b
23	48	11	73 b
17	45	10	66 b
20	65	7	46 b

* Assigned subject number

a Exited program after pretest

b Did not meet criterion level

c Met 80/80 revision criterion level but not 90/90 success criterion level

Table 9

Program Completion Times and Ranked Scores on Criterion Performance Test--Field Test B

Subject	Time (in minutes)	Number Correct	Percent Correct
27*	--	--	-- a
38	32	10	100
35	22	10	100
34	26	10	100
32	27	10	100
28	31	10	100
37	45	9	90
33	29	9	90
31	31	9	90
30	31	9	90
29	35	9	90
36	28	8	80 b

* Assigned subject number

a Exited program after pretest

b Met 80/80 revision criterion level but not 90/90 success criterion level

Table 10

Number of Subjects Meeting Criterion Levels by Group

Group	Criterion Level 80/80	Criterion Level 90/90
	Number Meeting Level / Total Number	
D-1	0/0	0/0
D-2	3/4	2/4
D-3	4/4	2/4
FT-A	10/14	7/14
FT-B	11/11	10/11

to be a versatile programming method. One of the advantages of PILOT is that technical programming difficulties can be solved by branching into a general-purpose language such as BASIC to write a particular segment and returning to PILOT to continue. A few terms in the computer program posed difficulties in developing realistic simulations. For example, it was not possible to develop a simulation to demonstrate cursor movement that would operate smoothly. Only two of these terms remained in the final version of the program used for Field Test B. The simulations for these terms were workable; thus, the researcher elected to stay with PILOT as the programming language.

CHAPTER V

SUMMARY, FINDINGS, CONCLUSIONS, AND RECOMMENDATIONS

Summary

The purpose of this study was to develop a validated computer-assisted instructional program to teach word processing terminology using an authoring language. For the study, a list of 60 word processing terms dealing with document processing was compiled from textbooks and software documentation. The list was mailed to six word processing content experts who sorted the terms according to the preferred sequence of instruction. The content experts also selected the names to be used in the program for those terms with alternate names.

Criterion levels were established for the program at 80/80 to indicate that revisions were needed and 90/90 to indicate that the program was successful. The 80/80 criterion level required a minimum of 80% of the subjects to score at least 80% on the performance test; likewise, the 90/90 level required a minimum 90% percent of the subjects to score at least 90% on the performance test.

Cognitive and affective performance objectives were written for the program, and a criterion performance test was developed based on the objectives. The computer program was developed following a detailed flow chart based on instructional design strategies for computer-assisted instruction identified in the literature.

During the development stage of program construction, the computer program was evaluated on a one-to-one basis with a subject representative of the target population. Two college instructors of word processing and a college employee with programming experience also evaluated the program. Revisions were made to the program based on the suggestions of the evaluators and the problems encountered during this evaluation stage.

During the revision stage of the development process, the program was evaluated by three groups of volunteer subjects. The researcher observed the subjects as they worked through the program and recorded any problems noted. Each subject was interviewed using a structured interview questionnaire. Subjects' comments and response data were analyzed to detect problem areas. Revisions were made to the program after each testing cycle.

The program was field tested with a group of 15 introductory computer technology students at Marshall University Community College after the third developmental testing cycle. One subject scored above 60% and exited the program after the pretest. Of the 14 subjects who completed the program and criterion performance test, 10 scored above 80 and 7 scored above 90 on the test.

Subjects' comments after the field test indicated the major problem with the program was not being able to complete the program within the allotted testing time. The program was shortened, and a second field test was conducted with 6 students from an introductory communications class and 6 students from a records management class at Marshall University Community College. One subject scored above 60% on the pretest and exited the program. Of the 11 subjects who completed the program, all scored above 80 on the criterion performance test and 10 scored above 90. The success criterion level of 90/90 was reached with this second field test. Analysis of the data from this field test indicated that no problems were listed that warranted additional revisions and field testing.

Findings/Conclusions/Recommendations

The following findings are related to the evaluation process and testing procedures used in this study. Based on the findings, conclusions and recommendations are offered that apply to the development of computer-assisted instructional materials but may be applicable to other types of programmed materials as well.

1. Finding: The formative evaluation process used in this study was successful in identifying problem areas in the instructional program.

Conclusion: Formative evaluation is an appropriate method to use in the development of computer-assisted instructional materials. This conclusion is supported in studies by Vasek (1983) and Sherron (1984), who identified formative evaluation as a valid method for locating and correcting problem areas in instructional programs.

Recommendation: Formative evaluation should be considered as a method for evaluating and improving computer-assisted instructional materials.

2. Finding: Some subjects in the study who indicated no prior word processing instruction possessed enough knowledge of the subject to exit the program after the pretest.

Conclusion: Absence of formal instruction does not preclude knowledge of the subject matter.

Recommendation: Subjects participating in computer-assisted instructional program evaluation should be pretested to determine their knowledge of the program content. An a priori cutoff level should be established for the test to eliminate subjects with prior knowledge of the program content. This recommendation is supported by Sherron (1984), who recommended that subjects be tested prior to the evaluation process to determine their level of knowledge.

3. Finding: The greatest number of problems was identified during the initial (one-to-one) evaluation of the program with a subject representative of the target population.

Conclusion: The one-to-one evaluation is a critical step in the evaluation process of computer-assisted instructional materials. This conclusion is supported by Sherron (1984), who found that the initial developmental testing was important to the formative evaluation process.

Recommendation: A one-to-one evaluation of computer-assisted instructional materials should be conducted prior to or in the early stages of programming.

Recommendation: A subject representative of the target population should be used for the one-to-one evaluation.

4. Finding: An authoring language (PILOT) proved flexible for the program in this study.

Conclusion: The PILOT authoring language used in this study is a workable programming language for drill and practice and tutorial forms of computer-assisted instruction, and can be used for simple graphics and simulations.

Recommendation: An authoring language should be considered as a feasible method for programming drill and practice and tutorial forms of computer-assisted instructional materials.

5. Finding: Environmental conditions created problems in observing some study subjects during trial use of the program and created interference with the evaluation process.

Conclusion: The testing conditions for computer-assisted instruction affect the evaluation process and results.

Recommendation: Evaluation of computer-assisted instructional materials should be conducted in a controlled environment to facilitate observation of the subjects and to minimize contamination of the data.

6. Finding: Lack of typing skill and lack of familiarity with the computer inhibited the learning process for some students.

Conclusion: Lack of requisite skills to operate computer-assisted instructional software inhibits the learning process.

Recommendation: Computer-assisted instructional materials should be written for an intended audience. Study subjects should have the requisite skills to complete the program, and the information necessary to guide each subject through the material should be clearly provided.

7. Finding: Comments from subjects revealed that feedback indicating correct/incorrect responses alone proved to be insufficient reinforcement.

Conclusion: Use of one form of feedback proved to be insufficient reinforcement for the computer program in this study.

Recommendation: More than one form of reinforcement should be provided for computer-assisted instructional programs dealing with acquisition of terminology. Fill-in responses, affirmation of correct/incorrect responses, criterion frames, review screens, and practice tests are suggested as different methods for achieving reinforcement.

8. Finding: Limitation of subject use of the program to approximately one hour during the evaluation process made it difficult for some students to finish and caused anxiety for these students.

Conclusion: Limitation of the amount of time available for evaluating computer-assisted instructional programs can inhibit the learning and testing processes.

Recommendation: Computer-assisted instructional materials should be evaluated in the setting in which they will be used. This recommendation concurs with a similar recommendation made by Sherron (1984). Limitation of time available for testing during the evaluation process should be carefully considered if time will not be a factor in regular use of the program.

Recommendation: The pool of subjects to be used for testing the program should be composed of individuals who have sufficient time available to complete the evaluation procedure.

9. Finding: One of the best ways to identify problems with the program was to observe the subjects as they worked through the material. Some subjects indicated that having the researcher observe them made them feel uncomfortable. (This finding applied to the developmental testing cycles; it was not applicable to the field tests.)

Conclusion: Observation of subjects as they evaluate computer-assisted instructional materials can inhibit subject participation in the evaluation process.

Recommendation: Steps should be taken to ensure that the observation process used during the evaluation of computer-assisted instructional materials does not inhibit subject participation.

Summary of Findings

The results of this study add support to the findings from related studies by Vasek (1983) and Sherron (1984) and provide direction for instructional use and further research. The use of formative evaluation as an effective means of identifying and correcting problem areas in computer-assisted instruction was supported by the three studies. The importance of pretesting subjects to ascertain prerequisite skills and knowledge of content prior to evaluation of CAI materials was supported by Vasek and Sherron as well as by this study.

Use of a one-to-one evaluation in the early stages of formative evaluation was found to be a crucial step in the program development process in this study. Sherron's study also emphasized the importance of the initial evaluation.

Although not listed as a finding of his study, Vasek reported interference with the testing procedure as a result of using an open lab to evaluate CAI software. Interference with the testing process was found to be a problem with this study and a recommendation was made that further study with computer-assisted materials should control for this factor.

This study yielded additional findings beyond the Vasek and Sherron studies. The program in this study was written with an authoring language that proved to be a workable programming language. The successful use of an authoring language in this study can serve as the basis for further software development using authoring languages.

The use of several forms of reinforcement was found to be important to student achievement in this study. Simple positive or negative feedback to correct/incorrect responses was insufficient reinforcement. A variety of reinforcement strategies were suggested for use in CAI programs.

Two findings in this study dealt with problems encountered during the evaluation process: limitation of the use of the program to approximately one hour and observation of the study subjects as they worked through the program. Limitation of the time available for testing the program caused anxiety for subjects who needed more time. It was recommended that future studies carefully evaluate the amount of time available for the evaluation process. Observation of the subjects caused some of them to feel uncomfortable in working with the program. It was recommended that the observation of study subjects be conducted in an unobtrusive manner to avoid inhibiting the subjects' full participation.

Instructional and Research Recommendations

The study findings and recommendations yield implications for instructional use and further research. The computer-assisted instructional program developed with an authoring language for this study provides a method for teaching a common core of word processing terminology. The program can serve as a model for development of terminology software. Further, the program can serve as a model for development of instructional materials through use of an authoring language. The program has potential use as enrichment, reinforcement, or review in word processing classes. It may also be used for independent study.

The results of the criterion performance test indicated that the program transmitted the knowledge it was designed to teach. However, no comparison was made between the method for learning terminology provided by this program and other methods of instruction. Following is a discussion of recommendations for instructional use and for further research.

Recommendations for Instructional Use

1. The program in this study was developed using an authoring language (PILOT) that proved to be flexible in meeting programming needs. The relative ease of writing and editing through the use of PILOT and the capability for graphics provide a workable programming language, which should offer encouragement and incentive to educators to develop course materials tailored to their needs.

2. Authoring languages were developed primarily for use with tutorial CAI such as the program developed in this study. The suitability of authoring languages for higher levels of CAI has not been established. However, the PILOT language used in this study was capable of simple graphics and simulations and has potential use beyond tutorial forms of computer-assisted instruction. Educators are encouraged

to experiment with authoring languages in developing various forms of CAI materials.

3. The computer-assisted instructional program developed in this study was designed to facilitate the acquisition of hands-on skill in word processing by providing a common core of word processing terminology. More effective use of class time and broader training for word processing students could result from the use of this program. Some of the potential uses follow.

a. The program could be used to provide initial terminology instruction to word processing classes. This approach would create a standard vocabulary of terminology, which would simplify the discussion of various word processing concepts and machine functions. The result would be more class time for hands-on instruction. (This approach would be useful in situations where access to computers outside the classroom is limited.)

b. The program could be used as a supplement to classroom instruction. Students could be assigned "homework" on the terminology software. This would free class time for hands-on instruction as students would already be familiar with the necessary terminology. (This approach would be suitable in situations where students have access to computers outside of the classroom.)

c. The program could be used as selective enrichment or reinforcement according to the individual needs of the students. Students could work through the program independently to reinforce classroom instruction or as a preliminary to classroom instruction. This use of the program would be particularly effective in classes where students have a wide range of word processing knowledge, which is often the case in community college courses.

d. The program could be added to the CAI materials available in the school learning lab or learning resources center. This would make the program available to any student who wished to use the program for review or who wished to learn about word processing.

e. The program could be used in other classes to introduce students to word processing. Incorporating the program into related classes such as introduction to business or business writing would serve to illustrate the processing of words and data as well as increase familiarity with the computer.

Recommendations for Further Research

1. Further study under controlled testing conditions and with larger sample sizes is necessary to evaluate the

effect of learning word processing terminology through use of the program in this study on the acquisition of word processing knowledge and skills as compared with other methods of learning terminology.

2. Additional factors such as student interest and motivation, learning styles, time required to learn the material, and retention of the material should be compared among various methods of word processing terminology instruction.

3. The computer-assisted instructional program in this study was primarily a tutorial design. Comparisons with various forms of CAI are needed to determine the most effective CAI approach for presenting basic terminology.

4. The success criterion level for the computer program in this study was arbitrarily set at 90/90 prior to development of the program. Once the program material was written, it was apparent that this level may have been too high. Vasek (1983) stressed the arbitrary nature of a priori criterion levels in reporting the results of his study.

In Vasek's study, although the revised program fell between the 80/80 and 90/90 established criterion levels, student performance indicated that the program transmitted the instructional material it was designed to teach. Thus, in further research, achievement of the criterion level should not be viewed as the absolute indicator of program success. In addition, criterion levels should be selected with careful consideration of the nature of the material, the amount of material to be learned, the time available for learning, the audience for the material, and the setting in which the material will be used.

5. Further study is needed to compare student achievement in word processing classes using a computer program as a supplement to classroom instruction with word processing classes in which a computer program is not used to present terminology. Studies comparing word processing courses in which a computer program is the sole method for presenting terminology instruction with courses in which another method is used are also recommended.

REFERENCES

Abedor, A. J. (1972). Development and field test of a model for formative evaluation of self-instructional multi-media learning systems. Viewpoints, 48(4), 9-43.

Alesandrini, K. L. (1984). Instructional design for computer applications. Paper presented at the Annual Meeting of the American Educational Research Association, New Orleans, Louisiana.

Alty, J. L. (1983). The impact of microtechnology--A case for reassessing the roles of computers in learning. Computers & Education, 6(1), 1-5.

Anelli, C. M. (1978). Computer-assisted instruction and reading achievement of urban third and fourth graders. (Doctoral dissertation, Rutgers University, 1977). Dissertation Abstracts International, 38(11), 6662-A.

Baker, E. L., & Alkin, M. C. (1973). Formative evaluation of instructional development (ERIC/AVCR Annual Review Paper). Audiovisual Communication Review, 21, 389-418.

Beck, J. J., Jr. (1979). An analysis of student attitudes toward computer-assisted instruction in Nebraska public high schools. (Doctoral dissertation, The University of Nebraska, 1979). Dissertation Abstracts International, 40(6), 3006-A.

Beebe, T. H. (1983). How to write your own instruction using a computer authoring system. Instructional Innovator, 28(6), 34-35, 48.

Bloom, B. S., Hastings, J. T., & Madaus, G. F. (1971). Handbook on formative and summative evaluation of student learning. New York: McGraw-Hill.

Branstad, M. A., Cherniavsky, J. C., & Adrion, W. R. (1980). Validation, verification, and testing for the individual programmer. Washington, DC: U. S. Government Printing Office.

Burke, R. L. (1982). CAI sourcebook. Englewood Cliffs, NJ: Prentice-Hall.

Burns, P. K. (1982). A quantitative synthesis of research findings relative to the pedagogical effectiveness of computer-assisted mathematics instruction in elementary and secondary schools. (Doctoral dissertation, The University of Iowa, 1981). Dissertation Abstracts International, 42(7), 2946-A.

Caldwell, R. M. (1980). Improving learning strategies with computer-based education. Theory Into Practice, 19(2), 141-143.

Capron, H. L., & Williams, B. K. (1984). Computers and data processing (2nd ed.). Menlo Park, CA: Benjamin/Cummings.

Carr, W. J. (1962). A review of the literature on certain aspects of automated instruction. In W. I. Smith and J. W. Moore (Eds.), Programmed learning: Theory and research (pp. 57-80). Princeton, NJ: D. Van Nostrand.

Casner, J. L. (1978). A study of attitudes toward mathematics of eighth grade students receiving computer assisted instruction and students receiving conventional classroom instruction. (Doctoral dissertation, University of Kansas, 1977). Dissertation Abstracts International, 38(12), 7106-A.

Chambers, J. A., & Sprecher, J. W. (1983). Computer-assisted instruction: Its use in the classroom. Englewood Cliffs, NJ: Prenctice-Hall.

Cohen, V. B. (1983). Criteria for the evaluation of microcomputer courseware. Educational Technology, 23(1), 9-14.

Cross, K. P. (1987). The adventures of education in Wonderland: Implementing education reform. Phi Delta Kappan, 68, 496-502.

Cruz, J. (1984). A critical review of programmed instruction. Educational Technology, 24(9), 29-31.

Cunningham, D. J. (1972). Comments on the case studies of formative evaluation--The sources of information. Viewpoints, 48(4), 111-117.

Dalgaard, B. R., & Lewis, D. H. (1985). Current status of computer-assisted instruction with implications for business educators. Journal of Education for Business, 61(1), 20-23.

Daughdrill, R. W. (1978). A comparative study of the effectiveness of computer-assisted instruction in college algebra. (Doctoral dissertation, The University of Mississippi, 1978). Dissertation Abstracts International, 39(6), 3431-A.

Dence, M. (1980). Toward defining the role of CAI: A Review. Educational Technology, 20(11), 50-54.

Dick, W. (1977). Formative evaluation. In L. J. Briggs (Ed.), Instructional design: Principles and applications. Englewood Cliffs, NJ: Educational Technology Publications.

Dick, W. (1981). Instructional design models: Future trends and issues. Educational Technology, 21(7), 29-32.

D'Onofrio, M. J., & Slama, M. E. (1983). An investigation of the computer literacy resulting from the use of microcomputers in high school accounting classes. The Delta Pi Epsilon Journal, 25(4), 143-153.

Drum, W. O. (1981-82). Microcomputers in business education. The Balance Sheet, 63, 143-145.

Evans, J. L. (1965). Programing in mathematics and logic. In R. Glaser (Ed.), Teaching machines and programed learning, II, (pp. 371-440). Washington, DC: National Education Association.

Futrell, M. K., & Geisert, P. (1984). The well-trained computer: Designing systematic instructional materials for the classroom microcomputer. Englewood Cliffs, NJ: Educational Technology Publications.

Gagne, R. M. (1977). The conditions of learning (3rd ed.). New York, New York: Holt, Rinehart, and Winston.

Gagne, R. M., Wager, W., & Rojas, A. (1981). Planning and authoring computer-assisted instruction lessons. Educational Technology, 21(9), 17-26.

Gallitano, G. M. (1983). The effects of a computer-based approach to teaching trigonometry on student achievement and attitudes. Unpublished doctoral dissertation, Columbia University Teachers College, New York.

Gary, B., Callahan, G., & Lanoue, P. (1985). Implications of information processing for vocational schools. Business Education Forum, (yearbook issue), 28-33.

Gleason, G. T. (1981). Microcomputers in education: The state of the art. Educational Technology, 21(3), 7-18.

Goddard, B. (1984). West Virginia begins a statewide computer network. Business Education Forum, 39(1), 6-7.

Golas, K. C. (1983). The formative evaluation of computer-assisted instruction. Educational Technology, 23(1), 26-28.

Green, E. J. (1967). The process of instructional programming. In P. Lange (Ed.), Programed instruction, the sixty-sixth yearbook of the National Society for the Study of Education (pp. 61-80). Chicago: The University of Chicago Press.

Grobman, H. (1971). Curriculum development and evaluation. The Journal of Educational Research, 64, 436-442.

Gropper, G. L. (1975). Diagnosis and revision in the development of instructional materials. Englewood Cliffs, NJ: Educational Technology Publications.

Hartley, J. (1972). Evaluation. In J. Hartley (Ed.), Strategies for programmed instruction: An educational technology. London: Butterworth & Co.

Hall, S. L. (1986). A study of information processing instruction in Virginia community colleges. Unpublished master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA.

Herbert, M. (1982). Microcomputer-assisted instruction: Equal may be better. Business Education Forum, 36(5), 31-33.

Hoffman, J. L., & Waters, K. (1982). Some effects of student personality on success with computer-assisted instruction. Educational Technology, 22(3), 20-21.

Hofstetter, F. T. (1985). Perspectives on a decade of computer-based instruction, 1974-84. Journal of Computer-Based Instruction, 12(1), 1-7.

Holmes, G. (1982). Computer-assisted instruction: A discussion of some of the issues for would-be implementers. Educational Technology, 22(9), 7-13.

Hord, E. V. (1984). Guidelines for designing computer-assisted instruction. Instructional Innovator, 29(1), 19-23.

Hughes, R. J. (1977). An experimental study in teaching mathematical concepts utilizing computer-assisted instruction in business machines. (Doctoral dissertation, North Texas State University, 1976). Dissertation Abstracts International, 37(11), 6911-A.

Johnson, W. G. (1974). The effects of computer-assisted instruction and programmed instruction on the achievement and attitude of ninth grade general mathematics students. (Doctoral dissertation, Temple University, 1974). Dissertation Abstracts International, 35(3), 1426-A.

Jorgensen, C. E., & Campbell, M. E. (1983). Business classroom and laboratory equipment. Business Education Forum, 37(4), 24-42.

Klaus, D. J. (1962). The art of auto-instructional programming. In W. J. Smith and J. W. Moore (Eds.), Programmed learning: Theory and research (pp. 89-107). Princeton, NJ: D. Van Nostrand.

Klass, R. (1984). The TenCORE language and authoring system for the IBM personal computer. Journal of Computer-Based Instruction, 11(3), 70-71.

Kulik, J. A. (1983). Synthesis of research on computer-based instruction. Educational Leadership, 41(1), 19-21.

Lange, P. C. (1967). Administrative and curricular considerations. In P. Lange (Ed.), Programed instruction, the sixty-sixth yearbook of the National Society for the Study of Education (pp. 147-177). Chicago: The University of Chicago Press.

Lasoff, E. M. (1981). The effects of feedback in both computer-assisted instruction and programmed instruction on achievement and attitude. (Doctoral dissertation, University of Miami, 1981). Dissertation Abstracts International, 42(4), 1553-A.

Lawson, T. E. (1973). Formative instructional product evaluation. Educational Technology, 13(5), 42-44.

LeMaster, A. J. (1985). Developing CAI programs for business writing. Business Education Forum, 39(6), 20-24.

Levin, H. M., & Meister, G. (1986). Is CAI cost-effective? Phi Delta Kappan, 67, 745-749.

Lippold, G. A. (1978). The relationship of personalization, encouragement, and humor to student attitudes and post-test performance on a computer-assisted instructional program. (Doctoral dissertation, University of Northern Colorado, 1977). Dissertation Abstracts International, 38(8), 4690-A.

Lundgren, T. D., & Lundgren, C. A. (1984). Occupational opportunities in the new business education. Business Education Forum, 39(3), 9-10.

Magero, J. (1980). A study comparing the effectiveness of two computer-assisted instructional design formats and subsequently correlating individual learning style to performance. (Doctoral dissertation, Northern Illinois University, 1979). Dissertation Abstracts International, 40(10), 5291-A.

Markle, S. M. (1967). Empirical testing of programs. In P. C. Lange (Ed.), Programed instruction, the sixty-sixth yearbook of the National Society for the Study of Education (pp. 104-138). Chicago: The University of Chicago Press.

McMeen, G. R. (1986). The impact of technological change on education. Educational Technology, 26(2), 42-45.

Merrill, M. D. (1985). Where is the authoring in authoring systems? Journal of Computer-Based Instruction, 12(4), 90-96.

Michael, F. W., Jr. (1981) Educational computing at the crossroads. Interface Age, 610, 68-70, 158.

Niemiec, R. P., Blackwell, M. C., & Walberg, H. J. (1986). CAI can be doubly effective. Phi Delta Kappan, 67, 750-751.

Office of Institutional Research. (1987). Demographic profile of community college students, spring 1987. Huntington, WV: Marshall University.

Paden, D. W., & Barr, M. D. (1980). Computer-assisted instruction in an elementary college economics course. Computers & Education, 4, 259-267.

Passmore, D. L., & Asche, F. M. (1978). Information needs for formative evaluation of career education products. Journal of Industrial Teacher Education, 15(2), 55-60.

Perreault, H. (1986). Software evaluation--Everyone's responsibility. Business Education Forum, 40(6), 5-9.

Podemski, R. S. (1981). Computer technology and teacher education. Journal of Teacher Education, 32(1), 29-33.

Pogue, R. E. (1980). The authoring system: Interface between author and computer. Journal of Research and Development in Education, 14, 57-67.

Popham, W. J., & Baker, R. L. (1971). Rules for the development of instructional products. In R. L. Baker & R. Schutz (Eds.), Instructional product development (130-168). New York: Van Nostrand Reinhold.

Pressman, I., & Rosenbloom, B. (1983-84). CAI, Its cost and its role. Journal of Educational Technology Systems, 12, 183-208.

Rankin, A. L. (1984). The effect of anxiety on performance in a computer-assisted learning task and its relationship to age and gender. (Doctoral dissertation, University of Southern California, 1983). Dissertation Abstracts International, 44(9), 2657-A.

Regan-Goff, S. (1985). Training WP professionals--How useful are tutorials? Business Education Forum, 40(3), 23, 24, 25, 28.

Rizza, P. J. (1981). How can we best produce courseware? (The process of courseware development). In Technology and Education (Proceedings of the National Conference on Technology and Education--January 26-28, 1981) (pp. 275-289). Washington, DC: Institute for Educational Leadership.

Roblyer, M. D. (1986). Courseware: A practical revolution. Educational Technology, 26(2), 34-35, 57.

Rowntree, D. (1974) Educational technology in curriculum development. New York: Harper & Row.

Rushinek, A. (1979). The use of computer-assisted instruction in the teaching of data processing and its effect on students' attitudes. Unpublished doctoral dissertation, University of Texas at Austin.

Sanders, J. R., & Cunningham, D. J. (1973). A structure for formative evaluation in product development. Review of Educational Research, 43, 217-236.

Sarapin, M. I. (1981). Formative evaluation: A model for instructional material development and revision. Journal of Industrial Teacher Education, 18(3), 3-13.

Sawyer, T. A. (1985). Human factors considerations in computer-assisted instruction. Journal of Computer-Based Instruction, 12(1), 17-20.

Schuyler, J. A. (1979). Programming languages for microprocessor courseware. Educational Technology, 19(10), 29-35.

Scriven, M. (1967). The methodology of evaluation. In B. O. Smith (Ed.), Perspectives of curricular evaluation (pp. 39-83). Chicago: Rand McNally.

Sherron, J. A. E. (1984). An empirically validated model program for teaching alphabetic keyboarding skills via microcomputer. (Doctoral dissertation, Virginia Polytechnic Institute & State University, 1984). Dissertation Abstracts International, 45(7), 1967-A.

Signer, B. (1983-84). The need for sequential formative and summative evaluations evident from practice. Journal of Educational Technology Systems, 12(1), 67-74.

Skinner, B. F. (1954). The science of learning and the art of teaching. In W. J. Smith and J. W. Moore (Eds.), Programmed Learning: Theory and Research (pp. 18-33). Princeton, NJ: D. Van Nostrand.

Smith, P. L., & Boyce, B. A. (1984). Instructional design considerations in the development of computer-assisted instruction. Educational Technology, 24(7), 5-11.

Spitler, C. D., & Corgan, V. E. (1979). Rules for authoring computer-assisted instruction programs. Educational Technology, 19(11), 13-20.

Splittgerber, F. L. (1979). Computer-based instruction: A revolution in the making? Educational Technology, 19(1), 20-26.

Sprecher, J. W., & Chambers, J. A. (1980). Computer assisted instruction: Factors affecting courseware development. Journal of Computer-Based Instruction, 7(2), 47-57.

Steele, K. J. S. (1982). The effect of computer-assisted mathematical instruction upon the computer literacy of fifth-grade students using a microcomputer. (Doctoral dissertation, Purdue University, 1981). Dissertation Abstracts International, 42(8), 3433-A.

Stocker, H. R. (1981). Integrating computer-based technology in the total business education program. Business Education Forum, 35(5), 25-27.

Stolurow, L. M. (1961). Teaching by machine. Washington, DC: U. S. Government Printing Office.

Tennyson, R. D. (1976). The role of evaluation in instructional development. Educational Technology, 16(9), 17-24.

Tyler, J. G. (1983). Your prescription for CAI success. Instructional Innovator, 28(2), 25, 26, 40.

Vargas, J. S. (1986). Instructional design flaws in computer-assisted instruction. Phi Delta Kappan, 67, 738-744.

Vasek, J. R. (1983). Using formative evaluation to develop a microcomputer assisted instruction program to teach simple microcomputer language acquisition. (Doctoral dissertation, Virginia Polytechnic Institute & State University, 1983). Dissertation Abstracts International, 44(11), 3370-A.

Vasek, J. R., & Vogler, D. E. (1984). Applying formative evaluation to microcomputer assisted instruction: A case study. Journal of Vocational Education Research, 9(3), 38-50.

Young, J. L. (1984). The case for using authoring systems to develop courseware. Educational Technology, 24(10), 26-28.

APPENDICES

APPENDIX A

COVER LETTER TO CONTENT EXPERTS

VIRGINIA TECH

Division of Administrative
and Educational Services

University City Office Building
Blacksburg, VA 24061

July 7, 1986

Dear :

Thank you for agreeing to perform a card sort as part of my dissertation research. The purpose of my research is to develop a validated computer-assisted program to teach a common core of information/word processing terminology. Time constraints dictate that the model program be limited to one module of instruction. Terms related to document processing have been selected for inclusion in this module.

Six word processing specialists have been chosen to select the order in which the list of terms should be addressed in the instructional module. A ranking procedure based on Q-sort methodology will be used to rank the terms in clusters according to the preferred sequence of instruction.

A packet of cards; instructions; and a stamped, addressed return envelope are enclosed. The card sort should take approximately 10-15 minutes. Please return the card sort by July 18. (I know your summer schedule is a very busy one. If work or vacation makes it impossible to meet this deadline, please return the card sort as soon as possible after July 18.)

Your willingness to participate in my research is very much appreciated. I look forward to hearing from you.

Sincerely yours,

Marjorie Keatley
VTE Doctoral Student

Enclosures: Instruction sheet
Additional terms sheet
Cards
Return envelope

APPENDIX B

CONTENT EXPERTS' CARD SORT

Instructions for Card Sort Procedure

Sixty information/word processing terms related to document processing are listed on these cards. You are to create four clusters of cards (with 15 terms in each) indicating the sequence of instruction for these terms. You do not need to rank the terms within each cluster. Four cards are included that are numbered 1-4, with 1 being the set of terms to be taught first and 4 being the set of terms to be taught last.

Place the cards numbered 1-4 on a flat surface. Place each of the 60 cards below one of the four numbered cards. (Keep in mind that terms for Group 1 cards should be taught first, terms for Group 2 cards should be taught second, and so on.) You may rearrange cards as often as you like or repeat the card sort.

When you have completed the card sort, place the numbered card for Group 1 on top of all the cards in that cluster and bind with a rubber band. (The cards do not need to be in any order.) Repeat for the other three clusters.

If there are other terms that you feel should be included, please list them on the attached sheet of paper, indicating the numbered group in which they should be included. If there are terms included that you did not recognize or found confusing, please list them on the attached sheet.

Enclose the cards, instruction sheet, and attached paper in the return envelope and mail to me.

Thank you for your help.

Marjorie Keatley
UCOB 244
Virginia Tech
Blacksburg, VA 24061
703-961-6136 wk
703-961-3195 hm

Additional Terms

Please list any additional terms under the appropriate headings, which correspond to the card sort groups.

1	2	3	4
____________	____________	____________	____________
____________	____________	____________	____________
____________	____________	____________	____________
____________	____________	____________	____________
____________	____________	____________	____________
____________	____________	____________	____________

Please list any terms you did not understand or found to be confusing.

____________	____________	____________	____________
____________	____________	____________	____________
____________	____________	____________	____________
____________	____________	____________	____________

APPENDIX C

DEMOGRAPHIC QUESTIONNAIRE

DEMOGRAPHIC QUESTIONNAIRE

Name_________________________________ Group________________

Have you had prior word processing instruction? Yes No

Sex: M F

Age:

under 18 ________

18 - 20 ________

21 - 30 ________

31 - 45 ________

over 45 ________

Race: White ________

Black ________

Other ________ (specify)

APPENDIX D

STRUCTURED INTERVIEW QUESTIONNAIRE

STRUCTURED INTERVIEW QUESTIONNAIRE

Subject Name ______________________________

Time: Start ____________ Stop ____________

1. What did you like about the program?

2. Was there anything that you disliked?

3. Did you find the program interesting or boring? All of it or portions?

4. Was the program too easy or too difficult? Too long?

5. Were there any parts that were unclear or caused you trouble? Be specific.

6. Do you have any suggestions for changes or revisions?

7. Would you like to learn or review word processing terminology by this method?

8. When you were asked to do something, were there adequate instructions?

9. Was there adequate instruction of the terms? (Enough information?)

10. What would you say is the major strength of the program?

11. The major weakness?

APPENDIX E

CRITERION PERFORMANCE TEST

CRITERION PERFORMANCE TEST

1. Giving a computer instructions on margins, tabs, and spacing that affect the appearance of text on a page is called:

    a. formatting
    b. editing
    c. paginating
    d. don't know

2. A scale that shows margins, tabs, and center point is called a:

    a. format line
    b. status line
    c. default
    d. don't know

3. Data in the form of numbers and/or alphabetic characters that are entered into a word processing system are called:

    a. defaults
    b. input
    c. variables
    d. don't know

4. A computer feature that sets tabs at regular intervals is called a(n) __________ __________.

5. The blinking block, dot, slash, etc., that shows where you are on the screen is a:

    a. constant
    b. text string
    c. cursor
    d. don't know

CRITERION PERFORMANCE TEST (CONT'D)

6. Lines displayed on the screen that give information such as document name, page, and function being performed are called:

   a. format lines
   b. status lines
   c. tab grids
   d. don't know

7. The spaces at the end of a line that signal the computer to return the carriage are known as the __________ __________ __________.

8. Word wrap

   a. inserts a return at the end of short lines
   b. is used to separate pages of text
   c. causes words to move to the next line
   d. don't know

9. Correcting, inserting, deleting, or replacing words or format instructions to improve a document is called:

   a. editing
   b. reformatting
   c. adjusting
   d. don't know

10. A hard return

   a. is inserted by the computer
   b. is used to separate pages of text
   c. is inserted at the end of short lines
   d. don't know

APPENDIX F

MEAN TIME REQUIRED TO COMPLETE PROGRAM

Mean Time Required to Complete Program

Group	Mean Time (in minutes)	Range (in minutes)
D-1	54	49 to 59
D-2	45	40 to 50
D-3	46	41 to 54
FT-A	41	24 to 65
FT-B	30	22 to 45

APPENDIX G

SOURCES OF WORD PROCESSING TERMINOLOGY AND DEFINITIONS

SOURCES OF WORD PROCESSING TERMINOLOGY AND DEFINITIONS

1. Word Perfect Software Manual*

2. Displaywrite 3 Software Manual*

3. Word/Information Processing: Concepts and Applications by Lina Grace Baber*

4. Word Processing for Colleges by Jean R. Warner*

5. Word Processing Operations by Jane T. Varner*

6. Word/Information Processing Concepts by Mona Casady

7. Office Automation and Word Processing Fundamentals by Shirley A. Waterhouse*

8. Word Processing and Information Systems by Marilyn K. Popyk

9. Word/Information Processing: Concepts of Office Automation by Marly Bergerud and Jean Gonzalez*

*Sources of word processing terminology definitions

APPENDIX H

LIST OF WORD PROCESSING TERMINOLOGY

LIST OF WORD PROCESSING TERMINOLOGY

advance
auto carrier return
auto centering
auto hyphen
auto indent
auto justification
auto underline
automatic tab grid
block
boilerplate
constant
copy
cursor
cursor control keys
decimal tab
delete
edit
format line
format
footer
get
global search/replace
go to
hard copy
hard return
header
page end code
input
insert
line end zone
merge
move
no adjust
original document
page
page advance
page break
paginate
reformat
repaginate

LIST OF WORD PROCESSING TERMINOLOGY (CONT'D)

repetitive document
required space
revised document
scroll
search
search/replace
selective search/replace
shell
strikeover
soft copy
spell check
status line
stop codes
super/subscript
system page
text string
undelete
variable
widow/orphan control
word wrap

APPENDIX I

LIST OF CONTENT EXPERTS

LIST OF CONTENT EXPERTS

1. Ms. Marly Bergerud
   Saddleback College
   28000 Marguerite Parkway
   Mission Viejo, CA 92692

2. Dr. Kitty Manley
   Ferris State College
   200 Education Building
   Big Rapids, MI 49307

3. Ms. Barbara Jude Ritzer*
   SHAW/WALKER
   Research, Design & Development Group
   57 East Willow Street
   Millburn, NJ 07041

4. Dr. Mona Casady
   Department of Office Administration
   and Business Education
   Southwest Missouri State University
   Springfield, MO 65804

5. Mrs. Audella Given
   420 Corbly Hall
   Marshall University
   Huntington, WV 25701

6. Dr. Jane Varner
   Montgomery College
   51 Mannakee Street
   Rockville, MD 20850

*replaced Dr. Leonard Kruk

APPENDIX J

LIST OF PROGRAM STRENGTHS

Strengths of Program--All Groups

Item	Frequency
Would like to learn this way	35
Adequate instructions	35
Interesting	34
Not too easy, not too difficult	22
Easy to understand/learn/follow	10
Liked the loops (chance to review)	7
Liked the repetition	5
Very informative	5
Instructions were good/clear	4
Learning the words (terminology)	4
Liked multiple colors	3
Liked graphics (graphics helpful)	3
Enables you to learn the terms quickly	3
Gave the answers	2
Very clear	2
The loops (questions) let you correct errors	2
Well done in that the user could see mistakes	2
Let me go at my own pace	2
Taught basic terminology	2
Using the computer and learning how at the same time	1
Could take your time	1
Taught me a lot	1

Strengths of Program--All Groups (cont'd)

Item	Frequency
Tutorial was good	1
Much better than tutorial for computer class; this was a lot easier	1
Makes you think	1
It was a learning experience	1
User friendly	1
The clarity of explanation of terms	1
An asset to the user	1
Taught me how to do it	1
The information given	1
The review questions after each group of words	1
Covered all parts equally	1
Told me if I was wrong	1
Taught me how to use certain keys	1
It will help me in class	1
Helps people like me who don't know very much about computers to learn and keep up with modern technology	1
Much of it I didn't know, so it was interesting	1
It was different	1
Lets you work with the computer	1
More interesting than learning by a book	1
It was thorough	1

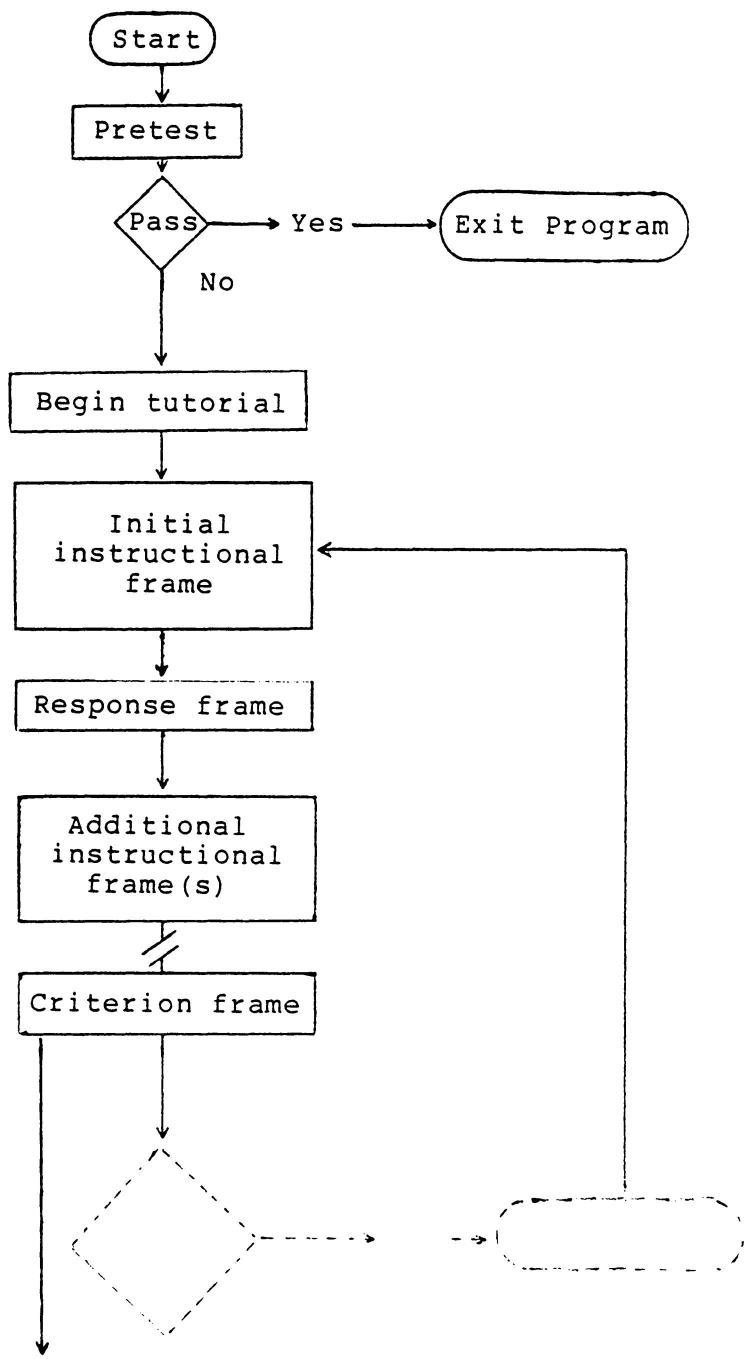
Strengths of Program--All Groups (cont'd)

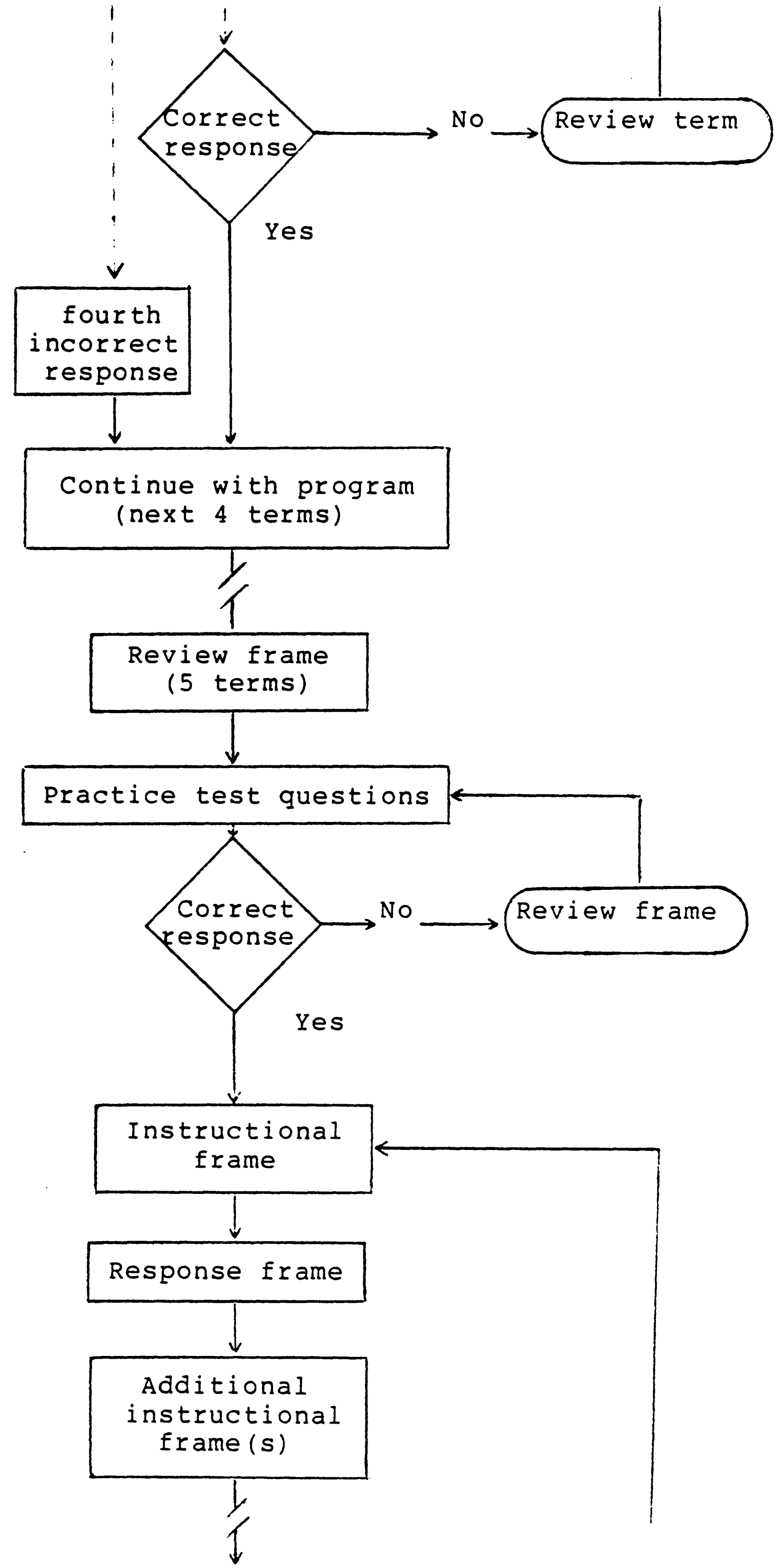
Item	Frequency
Not intimidating	1
It was my first time on the computer	1
The procedure used in presenting the information	1
The way it taught you	1
Very explanatory	1
Easy for someone with no word processing experience to understand	1
Very simple	1
Learned the basics	1
Gets across the terms	1
Well-defined	1
Better than reading; gets you involved	1
Having to type things helped	1

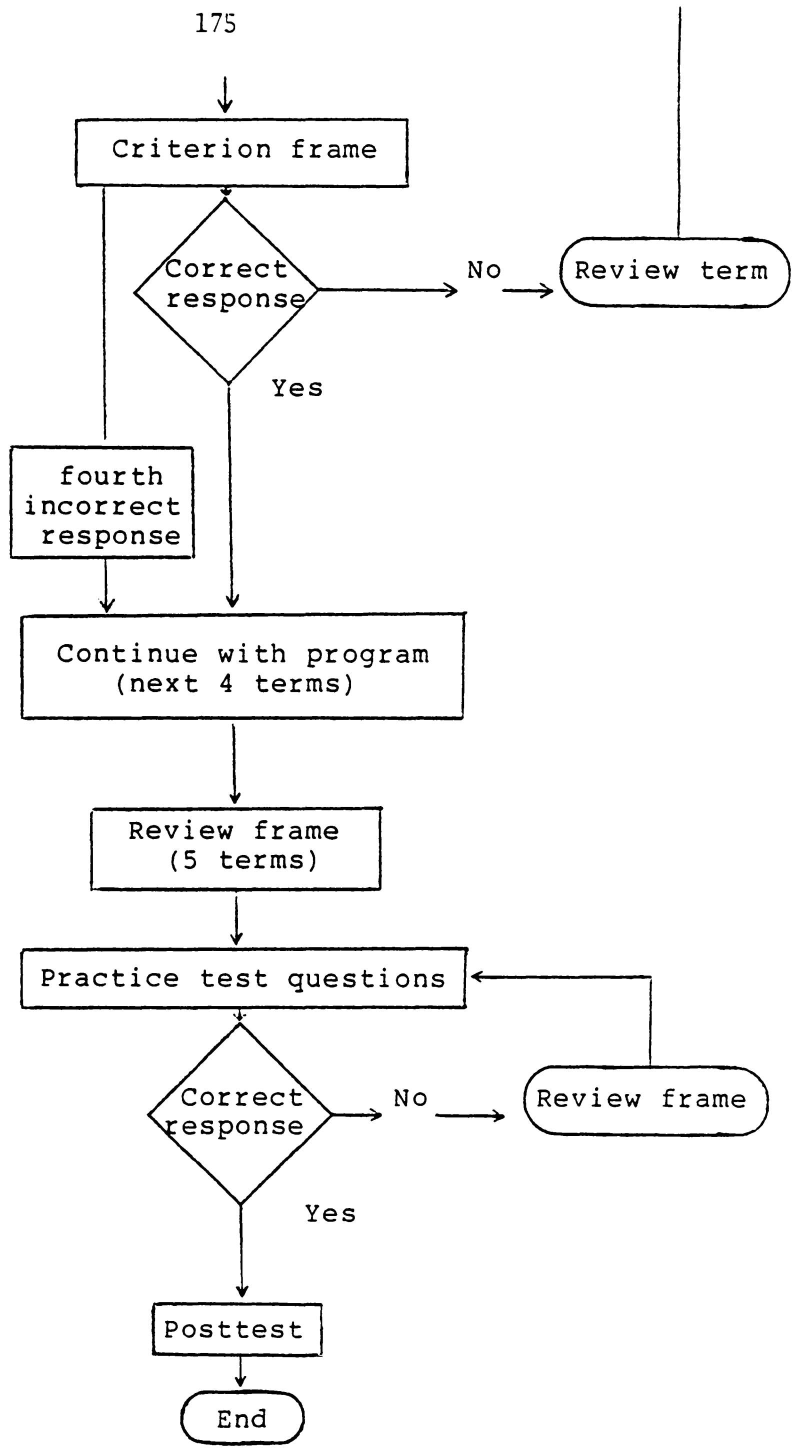
APPENDIX K

INSTRUCTIONAL PROGRAM FLOW CHART

INSTRUCTIONAL PROGRAM FLOW CHART




Correct response
No
Review term
Yes
fourth incorrect response
Continue with program (next 4 terms)
Review frame (5 terms)
Practice test questions
Correct response
No
Review frame
Yes
Instructional frame
Response frame
Additional instructional frame(s)


Criterion frame
Correct response
No
Review term
Yes
fourth incorrect response
Continue with program (next 4 terms)
Review frame (5 terms)
Practice test questions
Correct response
No
Review frame
Yes
Posttest
End

The vita has been removed from the scanned document