EDUCATIONAL CRITERIA USED BY MANUFACTURERS IN THE DEVELOPMENT AND PUBLICATION OF EDUCATIONAL SOFTWARE

By

SUSAN KIES ROTH

Bachelor of Science
The University of Illinois
Urbana, Illinois
1978

Master of Science
Oklahoma State University
Stillwater, Oklahoma
1983

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF EDUCATION
May, 1987

EDUCATIONAL CRITERIA USED BY
MANUFACTURERS IN THE DEVELOPMENT
AND PUBLICATION OF EDUCATIONAL
SOFTWARE

Thesis Approved:

Thomas D. Johnston
Thesis Advisor

[illegible]

L. Herbert Bruneau

N. Jo Campbell

Dean of the Graduate College

Name: Susan Kies Roth　　　　Date of Degree: May, 1987

Institution: Oklahoma State University

Location: Stillwater, Oklahoma

Title of Study: EDUCATIONAL CRITERIA USED BY MANUFACTURERS IN THE DEVELOPMENT AND PUBLICATION OF EDUCATIONAL SOFTWARE

Pages in Study: 75　　　　Candidate for Degree of Doctor of Education

Major Field: Curriculum and Instruction

Scope and Method of Study: The purpose of the study was to assess the educational criteria used by manufacturers in the development and publication of educational software and compare it to an evaluation system used by educators, who are members of the California Software Evaluation Consortium. The 132 subjects were educational software manufacturers, developers, and publishers identified by The 1986 Educational Software Preview Guide. Subjects were contacted and a request was made for the procedures and criteria used by subjects to select educational software for publication.

Findings and Conclusions: Ninety-one or sixty-nine per cent of publishers responded. Overall, the majority of publishers do not have a formal or standard set of criteria to guide in software development or employ in the selection of educational software for publication. However, the small number of publishers employing formal or some criteria or policies regarding the development and selection of educational software for publication do not employ the same criteria that educators deem important.

ADVISER'S APPROVAL Thomas W. Johnston

ACKNOWLEDGEMENTS

I would like to thank the members of my doctoral committee: Dr. Thomas D. Johnsten, Dr. Bruce A. Petty, Dr. N. Jo Campbell, and Dr. L. Herbert Bruneau who helped me with this study. I am especially grateful to Dr. Bruce A. Petty for his guidance, concern, and friendship during my graduate work at Oklahoma State University.

Special thanks are due to my husband, Tim, who gave me the confidence to pursue my graduate studies and the support to complete them. I thank my children, Gregory and Karen, and my mother, Clara Kies, for their inspiration.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The use of microcomputers for instructional purposes is growing rapidly. Watt (1983 p. 83) observed that "schools are in the grip of computer mania". Researchers have tried to determine the number of microcomputers in schools. In a study conducted at Johns Hopkins University, 53% of approximately 1600 elementary and secondary schools in the public, private, and parochial sector, had at least one microcomputer at the end of 1982 (Becker, 1983). Since 1982, the numbers of microcomputers in schools have grown at an astounding rate. Over the past few years the number of computers has roughly doubled each year (Bork, 1984). One estimate holds there will be at least one million microcomputers by the end of 1986 with 96% of the schools having at least one microcomputer (Ingersoll, Smith & Elliot, 1983). Several factors explain the influx of computers into the public schools.

One of the strongest reasons computers are becoming prevalent in public schools is the prevalence of computers in all aspects of society. The "information age" described by Toffler (1981) is upon us. As a reflection of the computer's impact on society, computers are proving to be one of the most important technological breakthroughs in education to date (Grabowski, 1984). Because of this impact on society, parents are demanding that schools prepare their children for the future technological market place (Tetenbaum and Mulkeen, 1986).

Research lends support for educational computer use in the areas of retention of material, development of positive student attitudes toward computers, and the reduction of time for content delivery. Kulik, et.al. (1980) reported that in four out of five studies at the secondary level, which examined retention over a six month interval, the groups receiving computer-based instruction scored higher than groups receiving traditional instruction. Edwards et.al. (1974) found that students experiencing computer assisted instruction possess a more positive attitude toward computers than those students who do not experience computer assisted instruction. Kulik et.al. (1980) found a "substantial and highly significant difference" between the amount of time necessary to instruct by conventional methods and through computer-based learning.

Several researchers have summarized studies on computer assisted instruction (Burns and Bozeman, 1981; Edwards et.al., 1974; Lawton and Gerschner, 1982; Visonhaler and Bass, 1972). The content areas most often examined are language arts and mathematics. Researchers generally concluded the following:

(1) Instruction supplemented by computer assisted

instruction was more effective than normal instruction alone.

(2) Computer-based instruction was not more effective relative to student achievement than traditional methods alone.

(3) Students who used computers obtained proficiency as measured by their teachers in shorter periods of time than students taught by traditional methods alone.

(4) Computer assisted instruction generated favorable student attitudes toward computers.

(5) Retention rates of students experiencing computer assisted instruction at least equaled those students experiencing traditional instruction.

Another reason for the prevalence of computers in the public schools is the decreasing cost of hardware. Microcomputers are now available from a number of manufacturers at a cost of less than $1000 for each machine. These costs are likely to be less than $100 for similar machines by 1990 (Otte, 1984). Modest costs make microcomputers attrative acquisitions for public schools whose financial resources are limited (Otte, 1984).

Significance of the Study

Although there is much enthusiasm for computer use in schools and research does confirm a positive impact of computers on education, educators still have many concerns regarding the development and publication of educational software. Computer manufacturers, software developers and educational publishers have entered aggressively into the development and marketing of equipment and software to support various educational applications of microcomputers (Otte, 1984). Since software publishing is in its infancy, many of those who engage in the publication of instructional materials lack requisite skills both in instruction and in the management of appropriate evaluation activities designed to have informational value for the user and to provide a basis for revision and modification of the software (Steffin, 1983). Further, software programs are frequently authored either by programmers who have little background in education or by educators who have little background in programming (Gold, 1984). These deficiencies have resulted in much software that is inappropriate or technically unsound (Gold, 1984). Many of the current software packages have left teachers dissatisfied and frustrated. A 1981 survey of computer use revealed that educational software was viewed as little more than electronic flashcards and workbooks (Gold, 1984). There was a general sense among educators that software was dull, unimaginative, and of questionable pedagogical soundness (Ingersoll, Smith and Elliot, 1983). Similarly, a 1983 survey of teachers using computers revealed that the majority were disappointed with the amount and quality of software available (National Education Association, 1983).

The literature is full of reports regarding the need and criteria for good quality software. However, few reports concern what software publishers are doing to meet these educational needs. This study is intended to bring the criteria manufacturers use to select and publish educational software to light.

Purpose of the Study

The purpose of this study is to access the educational criteria used by manufacturers in the development and publication of educational software and compare it to the criteria for the evaluation of software conducted by educators. In this study, the educational criteria for software evaluation are those used by members of the California Software Evaluation Consortium, which is constituted of approximately 30 member groups who routinely evaluate educational software.

Limitations of the Study

This study is limited to the software publishers identified by the California Software Evaluation Consortium in The 1986 Educational Software Preview Guide (Lathrop, 1986). These publishers are geographically located in the continental United States, and have been identified by the California Software Evaluation Consortium as publishing software of a quality high enough to be considered for preview by educators.

Assumptions

It is assumed that publishers of educational software have employed educational criteria when they design, develop, select, and publish software. It is also assumed that these criteria may be collected by contacting the publishers.

Definitions

Traditional Instruction. Instructional materials and methods employed by most public school teachers where textbooks are the main source of information and direction and delivery is presented by the teacher to all students as a group.

Computer Assisted Instruction (CAI). Instruction that is a combination of traditional instruction integrated with supplemental instructional activities wherein students interact with lessons programmed into the computer.

Logo. An interactive high-level procedural language developed for educational purposes at the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology.

Computer-Based Instruction. Classroom instruction where the primary delivery of instruction is provided by the computer to small groups of students and individuals.

Hardware. The physical computer equipment. This includes such items as the monitor, the keyboard, disk drives, and the printer.

<u>Software.</u> The learning package that is loaded into the computer. This contains the programming or code that tells the computer what to do.

<u>Computer-Managed Instruction.</u> Software that gathers, stores, and manages information to guide students through individualized learning experiences.

CHAPTER II

REVIEW OF RELATED LITERATURE

The purpose of this chapter is to present a review of the literature related to the potential of computers in education, how computers are used in today's classrooms, the current state of software, software and learning, characteristics of good software design, and the future of educational computing. Since technology and its impact upon education is changing so rapidly, the literature reviewed in this chapter was chosen to give background information to the reader.

The Potential of Computers in Education

Educators who acquired computers with the expectation that they were the answer to all educational problems have been largely disappointed (Tetenbaum and Mulkeen, 1986). Programming does not "make" problem solvers (Tetenbaum & Mulkeen, 1986); computer assisted instruction does not dramatically raise achievement scores above traditional instruction (Kulik, et.al., 1979); and, the motivational effects of computers are certain to evaporate as the novelty wears off (Tetenbaum and Mulkeen, 1986). In order to effectively implement the use of computers in the learning environment, educators must understand the computer's potential. The appropriate implementation of microcomputers in instruction can provide a broad range of new experiences in learning and thinking atypical of the convergent style of thinking traditionally prevalent in education (Steffin, 1981). Most important is that computer technology provides teachers with a workable means for adapting the creative process to suite the individual ability level of all students (Gallini, 1983). According to Ignatz (1985) the computer has the potential to: (a) provide practice sessions to enable students to sharpen needed skills; (b) drill endlessly and patiently as well as provide immediate feedback, encouragement, and reinforcement; (c) develop problem solving skills; (d) stimulate students to recall, apply, and integrate knowledge; (e) break down concepts into manageable steps; (f) encourage students to focus on one phase of the concept at a time until understanding occurs; (g) go beyond what the teacher does in the classroom; (h) provide additional help to students who need it; (i) promote knowledge processing and application strategies; (j) promote the development of problem solving skills; (k) permit experiments that require expensive or not readily available equipment or chemicals to be performed; and (l) provide opportunities for students to learn science concepts and processes which otherwise might not be possible due to such factors as the shortage of qualified teachers, overcrowded classrooms, and limited teacher preparation time.

As these systems are applied to computer assisted instruction, the goal is the development of a learning system in which the child is the teacher and the machine is the learner. This approach reverses the traditional model, in which the computer is a pre-programmed teaching machine which cannot be modified by the learner (Morris, 1983). "The revolutionary effects of the technology in education are tied to the nature of the computer itself...a computer is viewed not as a source of information, but as a problem-solving device..."(Fiske, 1983).

How Computers are Used Today

In this section, the discussion turns to the current uses of the microcomputer in the public school classroom. Taylor (1980) suggested that all computing in education can be categorized into one of three modes: "tutor/tool/tutee."

The use of microcomputers as a tutor has a long history, going back to B. F. Skinner and programmed instruction of the 1960s and 1970s. The following uses are included under tutor: drill and practice, course review, remediation, testing, homework, and instructional dialogue (Tamir, 1986).

The microcomputer as a tool can serve many functions related to management, administration, and instruction. The following uses are included under tool: calculation and statistical analysis; writing and word processing; drawing;

information display; generation of teaching aids; data accumulation and processing; information retrieval; decision making and problem solving; and, simulations and games (Tamir, 1986).

When the microcomputer is used in the tutee mode, the user "teaches" the computer by programming it (Tamir, 1986). Papert (1980) suggests this mode is very powerful because the student assumes the responsibility for his learning and this makes learning qualitatively different. Taylor (1980) noted several benefits arising from knowledge of programming:

> First, because you can't teach what you don't understand the human tutor will learn what he or she is trying to teach the computers. Second, by trying to realize broad teaching goals through software constructed from the narrow capabilities of computer logic, the human tutor of the computer will learn something both about how computers work and how his or her own thinking works. Third, because no expensive predesigned tutor software is necessary, no time is lost searching for such software and no money is spent acquiring it (p.4).

Fiske (1984) reported that the uses of computers range from single drill and practice to simulations of the theory of relativity. Schools are using computers in a variety of subject areas: in music for composition, foreign languages, reading, in home economics to analyze diets, in art for graphics, banking and applying for a job in vocational courses, in study courses for ACT and SATs, and in special education to accommodate students with learning difficulties. In addition, programming and programming languages are taught including BASIC, FORTRAN, Logo, and

Pascal. The Johns Hopkins University survey revealed that secondary schools are the largest pre-college users of microcomputers. The majority of their usage time is spent teaching students programming (Becker, 1983).

According to a 1983 survey, elementary schools use drill and practice programs more often than any other type of courseware. Approximately 40% of all instructional time on microcomputers in elementary schools is spent using courseware to practice mathematics, langauge facts, spelling, and various other memorization tasks. Thirty-five percent of instructional time is spent having students copy, write, and test computer programs. The remaining 25 percent is occupied playing games under the direction or approval of a teacher (Becker, 1983).

While secondary schools report little usage of microcomputers for drill and practice (19%), they report intensive use of microcomputers for programming in 64% of the high schools. The study conducted at Johns Hopkins University (Becker, 1983) also reported regional differences in computer usage across the United States. In the Northeast, the elementary schools do not use their microcomputers as intensively as other parts of the country and provide access to fewer students each week. The student-to-microcomputer ratio is even less in the South and microcomputers are used for drill and practice more there than in other regions. In Western cities the computers are used predominantly to teach programming skills to above-average students. Drill and practice is the preferred activity in rural elementary schools. It was reported that schools in the West and Midwest tended to have more active programs with microcomputers than other regions.

The State of Software Today

Thousands of computer software programs are available to support educational applications (Otte, 1984). Although it is widely said that current commercial educational materials for computers could be greatly enhanced, seldom is there much careful discussion of quality issues (Bork, 1984). While many computer assisted instruction programs deliver the promised instruction, the current majority cannot pass even a modest examination of their instructional value (Tyler, 1983). Many programs are nicely packaged and made very attractive and entertaining with the use of graphics but are not efficacious or cost-effective learning tools (Ignatz, 1985). Addressing the problem of quality control regarding software, Bork (1984, p.94) describes several factors that characterize poor software:

-failure to use adequately the interactive capabilities of the computer
-failure to use the individualizing capabilities of the computer
-use of extremely weak forms of interaction such as multiple choice
-heavily text-dependent presentations
-heavily picture-dependent presentations, where the pictures play no important role in the learning process
-screens treated like the page of a book
-material that is entertaining or attractive, but with no, or vague, discernible educational objective
-games which are nothing but games

-long sets of "instructions" at the beginning of programs, difficult to follow even by the teachers, and even more difficult to recall
-dependence on auxiliary print material
-small pieces of material, lacking context
-material which does not hold the student's attention.

Bialo and Erickson (1985) conducted a study of 163 software programs to detect strengths and weaknesses in instructional and technical designs. Trends emerged when data were analyzed by characteristic for all courseware. They found:

-There was an overwhelming lack of field-testing evidence in the course of program development for all courseware. Approximately 80.5% of the programs had no such development evidence.
-One-third of the programs had well defined, educationally appropriate objectives. More than one -half had either no objectives stated or had objectives that were unclear or developmentally inappropriate. Eighty percent of the courseware in logic/problem solving was weak in this area.
-Findings in the area of Goal/Content Math were quite mixed, with one notable result. More than 50% of all mathematics courseware had goals and objectives that were well supported by the content. Multiple disk mathematics programs made an even better showing with 66.7% having well supported goals and objectives.
-Findings with respect to the appropriateness of materials for intended users were mixed.
-Most programs (81.6%) were accurate in that there were few, if any, errors of fact, spelling, or language usage. Little controversial content and few instances of sex, race, or ethnic stereotyping were noted.
-Clarity of presentation as demonstrated in directions, frame formatting, and content expression was mixed across courseware. In the area of reading, results indicated that most of the reading courseware (55%) did not meet even minimal expectations.
-A large portion (68.8%) of the courseware examined included no support materials of any kind. When support materials were provided, they were not generally useful or appropriate.
-Most of the programs (87.1%) evaluated were easy to load and use and were free of programming errors.
-Fifteen percent of the programs had no warranty. When warranties were included they were typically 30

to 90 day replacement.

-Findings regarding operating instructions were mixed with one exception. Technical documentation for reading programs was clearer and more complete than in any other area.

-Instructional suggestions or information to aid in integrating a program into the curriculum were absent or inadequate in 62% of the programs.

-The inclusion of menus and options to exit, review instructions, call for help, alter the rate of presentation, etc., were present as often as not. Reading courseware, however, provided less user control than any other area.

-In most cases, feedback was immediate and included some form of reinforcement (68.8%). In only 16.1% of the programs was there any remediation.

-The use of graphics varied. In almost one-half of the early childhood, logic/problem solving and mathematics programs, graphics were more likely to be embedded in content as well as feedback, and to be clear, innovative and appropriate.

-There was no audio component in more than one -quarter of the programs evaluated. When it was used, 55.5% of the time it could not be turned off and was likely to be distracting, as was the case for 92.3% of the early childhood programs.

-In general, courseware tended to make little use of random generation. It was used in content and activities only 39.9% of the time.

-More than half of the courseware examined failed to use an approach that lent itself to effective delivery, or to appealing presentation which clarified or enhanced content. This was especially true for reading (70%) and language arts (63.2%) courseware.

-Seventy-eight percent of the courseware examined included no tests of any kind. That is, only 36 out of the 163 programs examined included tests.

-Almost three-fourths of the courseware did not make use of branching for presentation of either content or feedback. Mathematics programs, however, made better use of branching than any other area. When branching was incorporated, it was likely to be used for both content and feedback.

-Data on record keeping were varied across areas with the exception of early childhood programs, where 75% did not include any form of record keeping.

-Overwhelmingly, the programs examined had no management systems (78.9%). Out of the total sample, only 34 programs included a management system. Programs developed for early childhood (93.8%) and logic/problem solving (93.3%) were least likely to include this feature.

-There was little evaluation of student learning provided. Approximately 61.1% of the programs

> examined had no form of evaluation. Only 5 programs all in mathematics--adequately measured mastery of objectives. Multiple-disk programs were more likely to include some type of evaluation than single-disk programs (p.231-233).

In addition, Truett (1984) found that very little field testing occurred before educational software was marketed. Her advice regarding purchasing of software is:

> ...do not expect programs whose use in the field has been proven to result in a significant difference in student achievement, nor are data regarding such results (or any field test results, for that matter) likely to be included in a program's documentation even if this type of evaluation was performed on the product...it is standard practice for many producers to improve their computer software after marketing it in response to comments and feedback from users...do not assume that the educational software you purchase from commercial publishers represented many long years of formative evaluation and extensive national field studies. Not only is the business still too new and the cost of such studies high, but many producers are not even convinced that such efforts enhance marketability of their software...(p.12).

In summary, most software is aimed at teaching skills or draws almost exclusively on recall rather than developing higher order thinking. Further, programs are frequently "boring and pedagogically flawed" (Hechinger, 1982, p. C4), rarely drawing upon established educational psychological principles such as advanced organizers, intermittent reinforcers, or feedback as to why the answer is incorrect rather than merely presenting the correct answer (Tetenbaum and Mulkeen, 1986). Kenneth Komoski, director of Educational Products Information Exchange, concluded that only about one in four of the products reviewed met minimum technical and instructional standards and only three or four out of 100 were considered excellent (cited in Hassett, 1984). Leaders in the Association for Supervision and Curriculum Development estimate that about 75% of math software and 95% of language arts software are worthless (Otte, 1984). Hassett (1984) has expressed surprise at the overwhelming enthusiasm among educators for the computer given that there have been no clear breakthroughs in educational software and that the products now available are so poor.

Software and Learning

To understand what is missing from many of today's software programs, publishers must be aware of fundamental learning concepts associated with the development of instructional materials. The following discussion outlines basic cognitive concepts that must be considered when attempting the development of computer assisted instructional materials.

The development of Programmed Instruction was influenced by B. F. Skinner in the late 1950s. Shoemaker and Holt (1965) state that the early Programmed Instruction in industry was used to promote the acquisition of general text information, specific task-related information, and job skills and procedures. During the 1960s this behaviorist view was applied to computer assisted instruction as a means to facilitate learning. The predominant style of these programs, termed frame-oriented computer assisted instruction, consisted of course material in lessons that were optimized for individual students (Pliske and Psotka, 1986). Initial hardware and software development was costly; the types of programs that existed had limited applications; learner simulation and motivation were sometimes lacking; and, technically competent personnel were not readily available (Kamouri, 1984). With a new cognitive approach to instructional thinking being more widely adapted by the early 1970s, the prior emphasis on the retention of specific information was rapidly replaced by the interest in using the computers as interactive and responsive to students (Kamouri, 1984). To improve the quality of educational software, researchers focused on curriculum design. Wade (1980) took the work of Gagne and Briggs (1974) and applied it to computer instructional programs. He used their framework for classifying characteristics of instructional programs, termed "instructional events." They include (a) gaining attention; (b) informing the learner of the objectives; (c) stimulating recall of prerequisite learning; (d) presenting the stimulus materials; (e) providing learning guidance; (f) eliciting the performance; (g) providing feedback about performance correctness; (h) assessing the performance; and, (i) enhancing retention and transfer. Wade acknowledged that not all of these characteristics are necessarily found in every instructional program and that educational software might be used as part of a larger unit (i.e. to motivate, test, or drill). He also identified five characteristics of a good learning situation: (a) the learner must be approached positively; (b) the learner must be ready; (c) learning needs to be managed or facilitated; (d) the simulation must be practicable; and, (e) learning must be efficient. Again, Wade cautioned that most computer instructional programs are not intended as completely self-contained units to provide all elements of a good learning situation.

Additionally, Jay (1983) reported five human information processing abilities which cognitive psychologists would anticipate must be accounted for in order to develop good courseware. These abilities include the following: (a) memory and attention; (b) language and text characteristics; (c) graphics and visual processing; (d) cognitive characteristics of the user; and, (e) feedback to users.

Further, Jensen (1985) stated that a number of components in computer assisted instructional software design are typically overlooked: (a) attention to learning principles, including the transfer of stimulus control from one set of variables to another, and providing appropriate consequences for responses (Skinner, 1968 cited in Jensen, 1985); (b) a data base designed to individualize curriculum development; (c) a motivational system which includes reinforcers administered by the teacher as well as reinforcers generated by the computer (Xerox Palo Alto Research Center, 1980 cited in Jensen, 1985); and, (d) a delivery system which gives the teacher flexibility and control of the learning environment without requiring them to be computer experts.

In summary, if computer assisted instruction is to meet the needs of education, publishers must design software incorporating Gagne's and Briggs' (1974) "Instructional events", Wade's characteristics of a good learning situation, Jay's human information processing abilities, and Jensen's components of computer assisted instruction software design.

Characteristics of Good Software Design

In addition to understanding principles of learning, publishers must incorporate elements that make computer assisted software meet student learning needs. The following are elements that should be considered.

Roblyer (1983) described three areas that producers of educational software must consider when designing their products: (a) matching courseware to identified needs; (b) providing effective material for students; and, (c) integrating courseware into the teaching process.

According to Roblyer (1983, p.28-29), when matching courseware to identified needs, producers of software must perform a needs assessment to determine if the topics or objectives they have in mind for development meet particular criteria.

> ...For example, they are necessary in terms of stated objectives in school curriculum or there is a

> demonstrated link between acquiring the skills and learning the ones specified in the curriculum...If developers consider these issues before beginning a project, there should be a shift away from developing yet another math drill or game and an emphasis on making courseware match known needs in education.

To Roblyer (1983, p.29), effectiveness with students is an essential criterion. It is important "...that students can use the resulting materials easily, can accomplish the objectives specified for the package, and enjoy learning with the materials." To achieve this, producers must field test.

> ...Preliminary field testing can help identify not only screen errors and bugs in the program, but also directions which are confusing and items which frustrate students. Further tryouts with data collection can tell developers if students are achieving desired objectives and, if not, which parts need to be modified.

Roblyer (1983, p.29) suggested that producers must be involved in

> ...integrating courseware into the teaching process. Perhaps the most pervasive problem...<is> that <software> is limited in what it addresses...unfortunately, this combining is usually left to others to accomplish, and it is a task which teachers are often not aware that they are expected to do. The result is products which appear fragmented and which are difficult to fit into a teacher's existing curriculum.

A project undertaken at the Microcomputer Resource Center at Teachers College, Columbia University, developed attributes that should be used in development and evaluation of instructional software. Table 1 lists these attributes (Roblyer, 1981).

Along the same lines as Roblyer, Ignatz (1985, p.27) reported that "software must be scrutinized in light of its educational value, not as an innovation..." Software must: (a) have discernible behavioral objectives; (b) capitalize on the interactive capabilities of the computer; (c) provide for experimental error; (d) be free of factual errors, outdated information, improper use of statistics or inaccurate graphs or displays; (e) individualize capabilities of the computer; (f) use multiple tracking to accommodate different rates of learning of students; (g) have simple and easy to understand documentation; (h) be easily adaptable in a classroom situation; (i) be used by students with a minimum of teacher supervision; (j) have clear and concise instructions; (k) be able to maintain student interest throughout the presentation; (l) allow students to control its rate of presentation; (m) be free of system errors; (n) be well organized; (o) be free of racial, ethnic or sexual bias; (p) use blank spaces on a text screen; and, (q) emphasize terms and phrases with the use of delays, inverse, flashing mode, or sound.

Additionally, Czechowicz (1981) suggested that a well-designed educational program (a) assumes the user is naive; (b) includes user/teacher documentation; (c) provides branching routines; (d) is user-proofed; (e) recognizes the need to escape or pause mid-exercise; (f) uses good language; (g) gives control of presentation rate to user; (h) contains descriptive menus; (i) provides immediate non-judgmental responses; (j) reinforces correct responses; (k) is not merely page turning; (l) focuses on defined objectives; (m) makes appropriate use of graphics; and, (n) contains screen displays designed for ease of viewing.

Regarding computer-managed instruction, Goforth and Nassif (1984) suggested that the following functions be included to facilitate good software design. The software must: (a) establish an identification data base; (b) process performance data; (c) aid in writing reports; (d) perform utility functions; and, (g) have the capacity for cost effective data entry.

In summary, improving the development of software could make a much greater impact on resolving couseware problems than identifying them after the fact (Roblyer, 1983). Although considerable investment must be made to change the way courseware is currently being created, sound instructional design methods seem essential if microcomputers are to become a significant addition to our educational system. Not using all that we have learned to date about effective development, teaching, and learning methods would be missing an opportunity that may not come again (Roblyer, 1983).

The Future of Educational Computing

Computers have been a part of education since the 1960s when university educators harnessed mainframe computers for research and demonstration projects. For example, the PLATO system, which utilizes a mainframe computer linked to widely dispersed terminals via telephone lines had its beginning during the 1960s. Projects such as the PLATO system focused attention on the potential of the computer as an educational device, but cost and inaccessibility prevented widespread adoption (Berg and Bramble, 1983).

In the late 1970s the powerful but inexpensive microcomputer was introduced. Public schools began adopting computers as a means of involving students in the technology. Initially, computers performed low level drill and practice, and the major benefits derived by students were often more social than academic (Becker, 1983).

Now, computers and the courseware that runs on them is becoming more sophisticated. Berg and Bramble (1983) predict that significant hardware and software innovations will occur in the mid-1980s and continue until the turn of the century. These changes, they suggest, will include the following:

- Educational computing systems will decrease in price as several companies become dominant in the microcomputer hardware market. The microcomputer of the late 1980s will be less expensive and far more powerful than models which preceded it by a decade.
- Digitized voice output will become an important part of computer assisted instruction as microcomputer memory capacity increases and costs decline. Unlike synthesized voice, which is machine generated and artificial, digitized sound reproduces the human voice. Instructional software which utilizes digitized voice will be especially effective for young students.
- Instructional materials will become available which will utilize computers as one of the several media in the instructional program. Educators and instructional material developers will become much more sophisticated in the art of applying an appropriate technology to instructional problems. Especially in the area of early elementary skills

development, look for integrated instructional programs which allow the average student to progress several grade-skill levels in reading or math in less than a year.

-During the late 1980s, new developments in memory storage will make available inexpensive hand-held computers which can be downloaded and uploaded from a larger computer system. Students will take assignments home in the computer's memory and download their work to the classroom computer the next morning. The full potential of the inexpensive handheld computer remains to be explored.

-Classroom management software will allow for close individual tracking of student skill levels. Teachers will be able to monitor and adjust learning activities. Computers will also expose students to more learning activities in a school day than in the past. Computerization may automate previously inefficient aspects of the traditional classroom, allowing more education to take place in a given time frame (p.105).

Tetenbaum and Mulkeen (1986) explained how children must be prepared to meet the twenty-first century:

In the future, children will need to know how to learn, how to cope with change, how to build and evaluate a body of knowledge that will evolve throughout their life, and how to adapt to a changing work environment. They will need to acquire critical thinking, decision-making, and communication skills with an emphasis on the cognitive processes of inquisitiveness, sequential thinking, and problem solving. To function in business and industry, they will need to learn the traits valued in the new marketplace; namely flexibility, experimentation, autonomy, risk-taking, and innovation. These goals cannot be accomplished merely though the use of computers...<However computers,>...provide an occasion to reconceptualize schools, and to create a basis for change...Now thoughtful, well-developed proactive strategies are necessary so education can productively and meaningfully enter the twenty-first century. (p.102).

Summary

Technology and its impact upon education in changing very rapidly. In order to maximize the positive effects computers can have on the educational process, educators and producers of educational software must combine their efforts to develop high quality software that will meet the needs and challenges of the twenty-first century.

CHAPTER III

DESIGN OF THE STUDY

This study assesses the educational criteria used by manufacturers in the development and publication of educational software and compares it to an evaluation system used by educators. In this study, the educational criteria for software evaluation are those used by members of the California Software Evaluation Consortium, which is constituted of approximately 30 member groups who routinely evaluate software.

Subjects

The subjects are educational software manufacturers, developers, and publishers identified by The 1986 Educational Software Preview Guide (Lathrop, 1986) developed at the California TECC Software Evaluation Forum, December 2-6, 1985. These subjects are located in the continental United States and Canada. There were a total of 132 subjects who had at least one software product positively reviewed by three consortium members. These products were recommended for preview by educators wishing to purchase software for classroom use.

Procedure

The subjects were contacted by letter requesting the procedures and criteria employed in their selection of educational software for publication. (See Appendix A.)

If no response was received within a period of four weeks, a postcard was sent. Again, this card solicited the procedures and criteria used by the subjects to select educational software for publication. (See Appendix B.)

If no response was made after the second contact, a follow-up telephone call was made requesting the procedures and criteria used by the subjects to select educational software for publication.

Treatment of the Data

The data received from respondents was categorized and percentages were calculated providing the percent of subjects who employ each criterion in developing and/or selecting educational software for publication. The data was then plotted on a bar graph identifying the educational criteria consortium members ranked as most important in evaluating software. The following are the most important educational criteria for evaluating software, as identified by the consortium members in the study presented by Bitter (1986):

<u>Correctness of Content Presentation.</u> Is the program free

from content, informational, computational, grammatical, and syntactical errors?

Content Presentation. Is the pedagogical content presented in a clear, concise, logical, and manageable fashion and in sufficient depth of instruction and/or practice so that learning will take place?

Use of Technology. Is this an appropriate use of computer technology such that the program takes full advantage of the computer's capabilities and provides students with a learning experience that cannot be presented better in another media?

Integration into Classroom Use. Can the program be effectively and easily integrated into classroom use? Does the software lend itself to use within a classroom time frame? Are effective and appropriate teacher support materials available? Can the program be easily used by a teacher?

Ease of Use. Is the program user friendly?

Curriculum Congruence. Does the content directly support the curriculum?

Interaction. Is interaction effectively achieved for the target audience? Is there a sufficient amount and a sufficiently high quality of interaction to promote learning?

Content Sequence/Levels. Are there multiple levels of difficulty with appropriate incremental steps between the levels so that the development sequence and the difficulty of the levels is appropriate to the target audience?

Reliability. Is the program free from programming and technical errors?

User Control of Program. Can the user (student or teacher where appropriate) control the rate, amount, and sequence of presentation?

Feedback (General). Does the program correctly assess student input and provide appropriate and effective feedback messages?

Objectives. Are objectives clearly stated and are they met?

Motivation. Is the program motivational?

Branching. Are there branches to provide facility for individualized instruction according to the student's needs?

Negative Feedback/Help. Are corrective feedback messages or help screens provided as needed?

Content Modification. Can the content be modified by the teacher?

Content Bias. Is the content free from bias (race, sex, cultural, ethnic, stereotyping, and violence)?

Teacher Documentation. Is the documentation comprehensive, easy to understand and well organized?

User Support Materials. Are user support materials present? Where present, are they appropriate and effective?

Color, Sound, Graphics, Animation. If these features are present, are they used effectively to enhance the program?

Screen Displays. Are screen displays effectively and appropriately formatted?

Management System. Is there a management system which provides an effective means for recordkeeping and/or assignment control.

Summary

The 132 subjects were contacted and a request for the criteria they use in the selection and development of software was made. The responses were then categorized and percentages were calculated providing the percent of subjects who employ each criteria.

CHAPTER IV

ANALYSIS OF DATA

Participants in the Study

Of the 132 publishers of educational software identified by The 1986 Educational Software Preview Guide, 91 or 69 per cent of the publishers responded. Forty-one or 31 per cent did not respond, chose not to respond, or went out of business between the publication of The 1986 Educational Software Preview Guide and the execution of this study. Of the 91 respondents, 49 responded by letter. Thirty-two responded by telephone contact. The participants in this study represented 20 different states and Canada:

--twenty-five located in the State of California
--sixteen located in the State of New York
--eleven located in the State of Massachusetts
--seven located in the State of Illinois
--four located in the State of Texas
--three each located in the States of Connecticut, Michigan, Minnesota, and Oregon.
--two each located in the States of Kansas, Missouri, Washington, and in Canada
--one each located in the States of Arizona, Georgia, Iowa, Maryland, Ohio, Oklahoma, Utah, and Wisconsin.

The scope of responses differed greatly. Of the responses, eight (or 8.8% of) respondents sent detailed, typeset information explicitly outlining the procedural and developmental process they employ when developing and selecting educational software for publication. Twelve (or 13.2% of) respondents briefly outlined, in a letter format, those items their companies deem most important in the development and selection of software. Eleven (or 12.1% of) respondents listed three or less criteria. Sixty (or 67% of) respondents stated they had no formal guidelines. Forty (or 44% of) respondents stated that although they had no formal policy or educational criteria employed in software selection, they request that software be submitted so that it may be evaluated individually. As one publisher stated, "If we like the software and it fits into our line, we'll publish it."

Publisher responses were categorized by the researcher by criteria. Some responses applied to more than one criteria and were placed in both categories. Depending on the nature of the response, some statements were edited to fit grammatically within each category:

Correctness of Content Presentation. Is the program free from content, informational, computational, grammatical, and syntactical errors?

Nine (or 9.9% of) respondents stated the following.

Respondent 1:

--Do check spelling and punctuation thoroughly.
--Do use subroutines as much as possible to make programming clean.
--Don't embed control cues in literal strings.
--Do collaborate with someone to increase creativity, catch more errors, and bring different perspectives and skills into the project.
--Do use subject matter resources, such as textbooks, references, and context experts.

Respondent 2:

--The software should be functional and work on the computer.

Respondent 3:

--Textual content must be correct.

Respondent 4:

--Instructional integrity in content, grammar, and presentation must be embedded into the software.

Respondent 5:

--The content area must be addressed.

Respondent 6:

--The software must run without bugs.

Respondent 7:

--The content must be clearly presented with no bugs.

Respondent 8:

--Does the program correctly teach necessary concepts?

Respondent 9:

--Define all terms, use correct terminology.

Content Presentation. Is the pedagogical content presented in a clear, concise, logical, and manageable fashion and in sufficient depth of instruction and/or practice so that learning will take place?

Nine (or 9.9% of) respondents stated the following:

Respondent 1:

--Does the program correctly teach the necessary concepts?

Respondent 2:

--Does the software teach what it claims to?
--Is the lesson well-organized and presented in a logical manner?

Respondent 3:

--Is the design well thought out?

Respondent 4:

--The software must be educationally sound.

Respondent 5:

--The software must have instructional integrity.

Respondent 6:

--The program must be far more than drill and practice
--It must challenge the user's knowledge and teach critical thinking skills.

Respondent 7:

--The software must present the topic.
--Every screen must be prompted.
--We want step-by-step tutorials.

Respondent 8:

--The software must display clear and logical progression of the material.

Respondent 9:

--Do demonstrate the process of change. Show relationships and interactions.

Use of Technology. Is this an appropriate use of computer technology such that the program takes full advantage of the computer's capabilities and provides students with a learning experience that cannot be presented better in another media?

Ten (or 10.9% of) respondents stated the following.

Respondent 1:

--Don't just mimic another medium, such as pages of a text (that's a book) or a series of pictures (that's a film).

Respondent 2:

--Employ good use of the computer, i.e., not something that could be done as well in a book.

Respondent 3:

--The program is not an 'electronic textbook'--it must make an appropriate use of computers to enhance the educational experience.

Respondent 4:

--Is the material most effective on computer, or would another media be preferable?
--Does it take full advantage of the computer's capabilities?

Respondent 5:

--The software should use the computer's capabilities.

Respondent 6:

--It must be appropriate for use on a computer.

Respondent 7:

--Has there been sufficient research done to insure that the subject matter can be enhanced via CAI?

Respondent 8:

--Is the computer the appropriate medium to handle the content?

Respondent 9:

--Have features that speak highly of the software's technical implementation, e.g., state of the art graphics.

Respondent 10:

--The software must take full advantage of the computer.

Integration into Classroom Use. Can the program be effectively and easily integrated into classroom use? Does the software lend itself to use within a classroom time frame? Are effective and appropriate teacher support materials available? Can the program be easily used by a teacher?

Eight (or 8.8% of) respondents stated the following.

Respondent 1:

--Is the program consistent with educational practices in the classroom?

Respondent 2:

--Is there a need in the educational marketplace for the product and will it integrate into the classroom?

Respondent 3:

--The software must be appropriate to the classroom environment and target students.

Respondent 4:

--The software must demonstrate effective use of student and teacher time.

Respondent 5:

--The program must be suitable for classroom use.

Respondent 6:

--Don't let the lesson last a long time without a temporary exit for the student.

Respondent 7:

--The software must be flexible for classroom use.

Respondent 8:

--Does the software fit with in the school environment?

Ease of Use. Is the program user friendly?

Six (or 6.6% of) respondents stated the following.

Respondent 1:

--Insure your program is 'user friendly.'

Respondent 2:

--Provide quick access to the most common segments in the lesson.
--Provide a quick path to directions, explanations, game rules or help.
--Provide clear menus and instructions that make the software easy to use.

Respondent 3:

--The software should be very simple and easy-to-use.
--We always assume no previous knowledge of computers.

Respondent 4:

--We want user friendly tools that prompt the user.

Respondent 5:

--We look for ease of use and management.

Respondent 6:

--We want powerful, easy-to-use products that are totally original.

Curriculum Congruence. Does the content directly support the curriculum?

Ten (or 10.9% of) respondents stated the following.

Respondent 1:

--Does the software meet an unfulfilled need!
--Does the software fit within the school environment and can be practically implemented in the school environment?

Respondent 2:

--Does the software fit into the curriculum?

Respondent 3:

--Is there a need in the educational marketplace for the product?

Respondent 4:

--Does it fill a need?

Respondent 5:

--Does the software have importance to curricular goals and teacher needs?

Respondent 6:

--Is there a market need?
--Does the material have a large enough audience...does it fit the curriculums of must school districts in the country?

Respondent 7:

--The software should fit curricular needs.

Respondent 8:

--How will the software's content fit into the curriculum?

Respondent 9:

--Programs must be curriculum-based.

Respondent 10:

--You must adapt content to fit curricular needs.

Interaction. Is interaction effectively achieved for the target audience? Is there a sufficient amount and a sufficiently high quality of interaction to promote learning?

Five (or 5.5% of) respondents stated the following.

Respondent 1:

--Do provide information to help the student make decisions and evaluate progress.
--Do provide alternative paths through a program for student and instructor use.
--Do produce examples from a general model. This encourages concept learning as students repeat the lesson. Apply this technique to exercises also.

Respondent 2:

--Include interactive techniques.

Respondent 3:

--We want step-by-step interactive tools.

Respondent 4:

--The program is far more than 'drill and practice'--it challenges the user's knowledge and teaches critical thinking skills.

Respondent 5:

--Include a complete description of the program, how it functions, and how students interact with it.
--Does it use the computer in an interactive manner with branching and specific remediation and reinforcement?

Content Sequence/Levels. Are there multiple levels of difficulty with appropriate incremental steps between the levels so that the development sequence and the difficulty of the levels is appropriate to the target audience?

Five (or 5.5% of) respondents state the following.

Respondent 1:

--The software must be appropriate to the classroom environment and target students.

Respondent 2:

--Can the program accommodate students with different levels of ability?

Respondent 3:

--The software must state intended audience age or grade levels.

Respondent 4:

--The software must have an appropriate reading level for the target age.

Respondent 5:

--Do provide ways to change the anticipated sequence of events within a program.
--Do plan for novice and expert users. Novices will need more guidance (instructions, examples, and help). Experts should not be subjected to novice cues.

Reliability. Is the program free from programming and technical errors?

Five (or 5.5% of) respondents stated the following.

Respondent 1:

--Do be prepared to handle common disk error interrupts.
--Do use subroutines as much as possible.
--Don't embed control codes in literal strings.
--Do include error trapping as the first statement of each program.

Respondent 2:

--We consider programming to be of great importance. The software should be technically sound.

Respondent 3:

--The software should run without bugs.

Respondent 4:

--The content should be clearly presented without bugs.

Respondent 5:

--The software should be functional and work on the computer.

User Control of Program. Can the user (teacher or student where appropriate) control the rate, amount, and sequence of presentation?

Two (or 2.2% of) respondents stated the following.

Respondent 1:

--What kind of control does the user have over the program?

Respondent 2:

--Allow student control of lesson pacing.
--Provide paths through a program for student and instructor to use.
--Don't let the lesson last a long time without a temporary exit for the student.

Feedback (General). Does the program correctly assess student input and provide appropriate and effective feedback messages?

Three (or 3.3% of) respondents stated the following.

Respondent 1:

--What kind of feedback will users receive (e.g. reinforcement, error messages)?

Respondent 2:

--Does it use the computer in an interactive manner with branching and specific remediation and reinforcement?

Respondent 3:

--Do check for expected wrong responses. Prepare constructive feedback tailored to these errors and provide digressions, if necessary.
--Do allow for misspellings, unless correct spelling is a lesson objective. In your feedback call attention to the correct spelling.
--Do reinforce correct answer with informative feedback.

--Do provide concise and courteous messages. Attempts at humor often don't come off well and become stale quick.
--Do guide the student with help, hints, review, and corrective feedback, e.g., feedback that increases the probability of the student performing better if given the question or task again.

Objectives. Are objectives clearly stated and are they met?

Six (or 6.6% of) respondents stated the following.

Respondent 1:

--We look for fulfillment of stated objectives.

Respondent 2:

--Does the software teach what its manufacturer claims?

Respondent 3:

--Include objectives of the program.

Respondent 4:

--The software should present the topic.

Respondent 5:

--What are the menu contents of your program? How will the contents address the needs of the proposed audience?

Respondent 6:

--Regardless of format, the software and manuals should include the statement of objectives.

Motivation. Is the program motivational?

Seven (or 7.7% of) respondents stated the following.

Respondent 1:

--Do provide progress reports frequently.
--Don't overuse fancy displays (animation, color, sound) for motivation.
--Do use competition with good students, but remember that competition is counter-productive with students who lose.

Respondent 2:

--The software should be motivational in nature.
--Students should WANT to use it.

Respondent 3:

--The program is motivational and innovative.
--The program is fun to use and the educational value is superb.

Respondent 4:

--The software must be motivational for students.

Respondent 5:

--We want products that encourage thinking and abstract learning and will inspire the user to want to learn more.

Respondent 6:

--Will the student feel challenged and successful after using the product?

Respondent 7:

--Does it allow the student to work basically unsupervised? (This should be accomplished by keeping the student's interest and offering a flexible pace.)

Branching. Are there branches to provide facility for individualized instruction according to the student's needs?

Three (or 3.3% of) respondents stated the following.

Respondent 1:

--Does it use the computer in an interactive manner with branching and specific remediation and reinforcement?

Respondent 2:

--Can the program accommodate students with different levels of ability?

Respondent 3:

--Don't force the learner to go though several menus to reach segments in the lesson.
--Do let good students bypass easy sections.

Negative Feedback/Help. Are corrective feedback messages or help screens provided as needed?

One respondent (or 1.1% of respondents) stated the following.

Respondent 1:

--Do provide concise and courteous error messages. Attempts at humor often don't come off well and become stale quickly.
--The software should be infinitely patient (and non-threatening.)
--Do provide a quick path to directions, game rules help.

Content Modification. Can the content be modified by the teacher?

One respondent (or 1.1% of respondents) stated the following.

Respondent 1:

--Don't force the learner to go through several menus to reach segments in the lesson. Instead, provide quick access to the most common segments in the lesson.
--Do maintain previous parameter settings, so students can ask 'what if' questions of the simulation and then compare their results.

Content Bias. Is the content free from bias (race, sex, cultural, ethnic, stereotyping, and violence)?

No respondents addressed Content Bias.

Teacher Documentation. Is the documentation comprehensive, easy to understand and well organized?

Four (or 4.4% of) respondents stated the following.

Respondent 1:

--Many packages require both a student manual and an instructor manual...If there is information only the instructor must see (for instance, answers to discussion questions or technical notes on how to change program parameters), you may have to create two documents...
--The following is a checklist of some essentials of good documentation. In preparing the manuals for your package, you should pay special attention to each of these points: Educational Purpose, Lesson Plans, Instructions or Tutorial, Background Information or

Theory, Exercises, Format, Language, Summary or Reference.

Respondent 2:

--Please explain the nature of the printed material, if any, that will accompany the software. How will it help users get the program up and running? What will be the coverage and organization of the printed material? What published length do you anticipate? What type and how many illustrations do you anticipate?

Respondent 3:

--Documentation should be available, but not necessary.

Respondent 4:

--The package should include helpful and accurate documentation.
--Direct teacher recommendations should be included.

User Support Materials. Are user support materials present? Where present, are they appropriate and effective?

Three (or 3.3% of) respondents stated the following.

Respondent 1:

--Explain the material that will be accompanying the program.

Respondent 2:

--If your package is aimed primarily at the high school market, you should include a set of lesson plans or detailed directions for the instructor on how the programs can be used in the classroom.

Respondent 3:

--Direct teacher recommendations should be included.

Color, Sound, Graphics, and Animation. If these features are present, are they used effectively to enhance the program?

Five (or 5.5% of) respondents stated the following.

Respondent 1:

--Have features that speak highly of its technical implementation, e.g., state of the art graphics.

Respondent 2:

--Make full use of the computer's technical capability, sound, color, and graphics.

Respondent 3:

--The software should offer meaningful graphics. The drawings should be as precise and accurate as well as necessary to the program.

Respondent 4:

--Will you use graphics, color, or sound? If so, how?

Respondent 5:

--Do be graphic, interactive, and adaptive.
--Don't use flashing text. Use inverse video sparingly, it at all.
--Use visuals (through graphic display of relationships).
--Don't overuse fancy displays (animation, color, sound) for motivation. These can distract the student from essential information.

Screen Displays. Are screen displays effectively and appropriately formatted?

Two (or 2.2% of) respondents stated the following.

Respondent 1:

--Do use areas of the screen in a consistent manner.
--Don't clutter the screen. Instead, help the learner's eyes focus on new and important information on the screen.
--Do build screens in a natural reading order: left to right and top to bottom.

Respondent 2:

--Every screen should be prompted.

Management System. Is there a management system which provides an effective means for recordkeeping and/or assignment control?

Two (or 2.2% of) respondents stated the following.

Respondent 1:

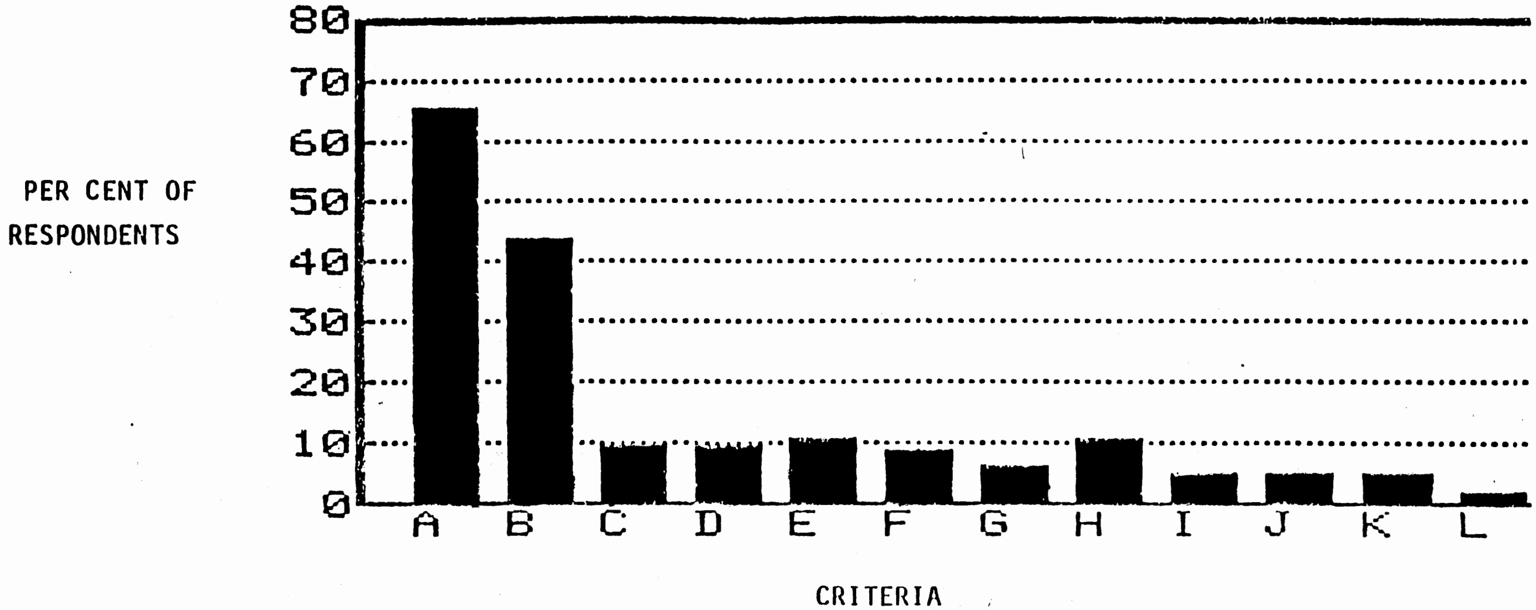
--Do keep student records for student restart and progress reports.

--Do keep student records for teacher evaluation of students and of the lesson.
--Do provide a print option for instructors and students.

Respondent 2:

--Include a data management system so teachers may determine student progress.

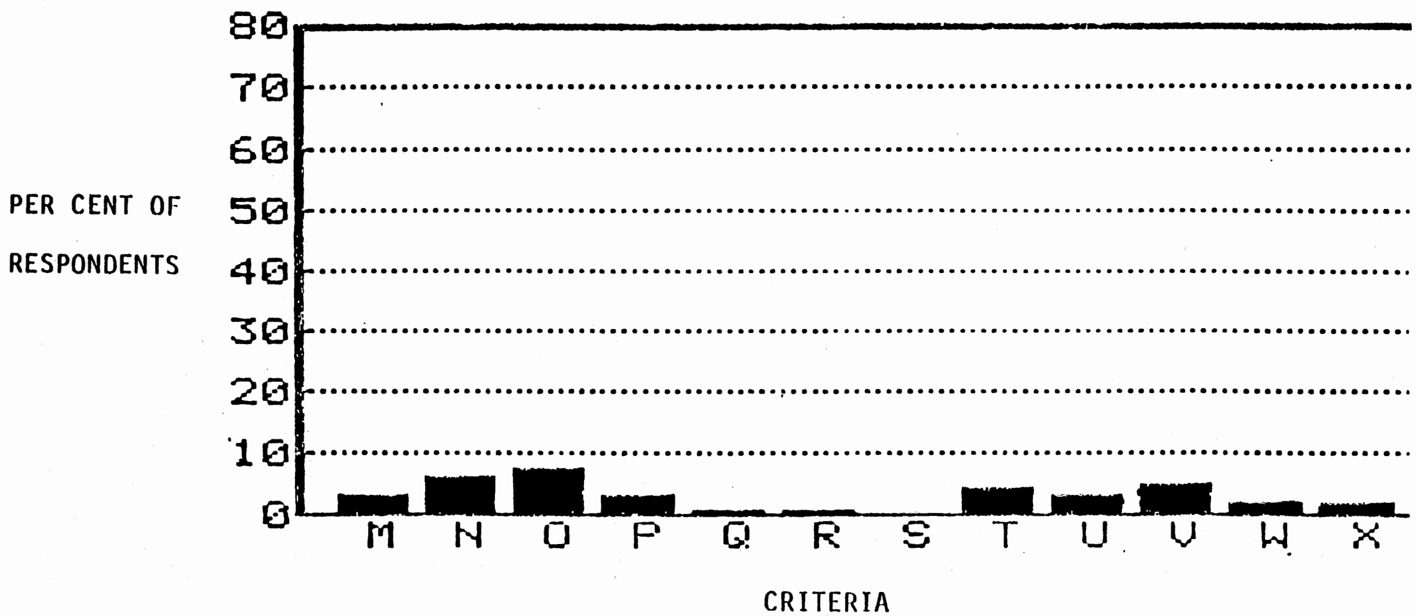
The bar graphs in Figure 1 represent percentages of respondent use of each criteria.



DATA:

A -- No Standard Criteria 67.0
B -- Consider Each Product Separately 44.0
C -- Correctness of Presentation 9.9
D -- Content Presentation 9.9
E -- Use of Technology 10.9
F -- Integration into the Curriculum 8.8
G -- Ease of Use; 6.6
H -- Curriculum Congruence 10.9
I -- Integration into the Classroom 5.5
J -- Content Sequence/Levels 5.5
K -- Reliability 5.5
L -- User Control 2.2

Per Cent of Respondents Employing Criteria



DATA:

M -- Feedback (General)	3.3	S -- Content Bias	0.0
N -- Objectives	6.6	T -- Teacher Documentation	4.4
O -- Motivation	7.7	U -- User Support Material	3.3
P -- Branching	3.3	V -- Color, Sound, Graphics, Animation	5.5
Q -- Negative Feedback	1.1	W -- Screen Displays	2.2
R -- Content Modification	1.1	X -- Management System	2.2

Per Cent of Respondents Employing Criteria

Unanticipated Findings

Several respondents included criteria they use in developing and selecting software that were not included in the Bitter study (1986).

These criteria include:

Problem Solving. One respondent (or 1.1% of respondents) stated the following.

Respondent 1:

--We are interested in software that sharpens reasoning skills, critical thinking, and logic.

Marketability. Three (or 3.3% of) respondents stated the following.

Respondent 1:

--We try to determine marketability. Does the product fit into our market area? Is it too competitive with other products we have under development?

Respondent 2:

--Frankly, we want software that will make us money, that's the bottom line.

Respondent 3:

--Is there a market and is this market willing to pay the high cost of development and marketing the software?

Uniqueness. Four (or 4.4% of) respondents stated the following.

Respondent 1:

--We look at the uniqueness of the idea and/or overall quality of the programming effort.

Respondent 2:

--We look at originality.

Respondent 3:

--The software should be needed. If a package exits that does what yours does, then your package should offer a significant improvement or it should not be published. Crowding the field with more of the same thing is unnecessary.

Respondent 4:

--The program should not be similar or closely related to existing software products, unless there is a clear advantage to the proposed program.

Total Cost to Produce. Two (or 2.2% of) respondents stated the following.

Respondent 1:

--We consider programming, royalties, coordination, documentation, packaging and distribution to determine a total cost proposal.

Respondent 2:

--Is the undertaking financially viable?

Future Product Development. One respondent (or 1.1% of respondents) stated the following.

Respondent 1:

--Does the software lead to more software products and ideas?

Summary

Overall, the majority (67%) of respondents do not have a formal or standard set of criteria to guide in software development or employ in the selection of educational software for publication. A very small number (8.8%) of publishers have formal criteria or policies regarding the development and selection of educational software for publication.

Those respondents that do have formal or some criteria (56.1%) do not employ the same criteria that educators deem important in the development of educational software. Of those respondents having formal or some criteria:

--thirty-two percent employ the criterion of Use of Technology and Curriculum Congruence
--thirty-one percent employ the criterion of Correctness of Content Presentation and Content Presentation
--twenty-six percent employ the criterion of Integration into Classroom Use
--twenty-three percent employ the criterion of Motivation
--nineteen percent employ the criterion of Ease of Use and Objectives
--sixteen percent employ the criterion of Interaction, Content Sequence/Levels, Reliability, and Color, Sound, Graphics, and Animation
--thirteen percent employ the criterion of Teacher Documentation
--ten percent employ the criterion of Feedback (General) and User Support Materials
--six percent employ the criterion of User Control of Program, Screen Displays, and Management System
--three percent employ the criterion of Negative Feedback/Help and Content Modification
--zero percent employ the criterion of Content Bias.

Similarly, a large number (44% of respondents) request that potential programs be submitted to them for evaluation on an individual basis. This evaluation is conducted by some respondents on the basis of marketability, uniqueness, total cost to produce the product, future product development and whether or not the software teaches problem solving skills.

CHAPTER V

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

SUMMARY

As the use of microcomputers for instructional purposes continues to grow, the potential impacts of the technology are becoming apparent in the areas of retention of material, development of positive student attitudes toward computers, and the reduction of time for content delivery. Although there is much enthusiasm for computer use in schools and the research does confirm a positive impact of computers on education, educators still have many concerns regarding the development and publication of educational software.

Many of the current software packages have left teachers dissatisfied and frustrated. A 1981 survey of computer use revealed that educational software was viewed as little more than electronic flashcards and workbooks (Gold, 1984). There was a general sense among educators that software was dull, unimaginative, and of questionable pedagogical soundness (Ingersoll, Smith, and Elliot, 1983). Similarly, a study of teachers using computers revealed that the majority of teachers were disappointed with the amount and quality of software available (National Education

Association, 1983). The literature is full of reports regarding the need and criteria for good quality software. However, few reports concern what software publishers are doing to meet these educational needs. This study is intended to bring to light the criteria publishers employ when developing and selecting educational software for publication.

This study assessed the educational criteria used by manufacturers in the development and publication of educational software and compared it to an evaluation system used by educators. In this study, the educational criteria for software evaluation are those used by members of the California Software Evaluation Consortium, which is constituted of approximately thirty member groups who routinely evaluate software.

The 132 subjects were educational software manufacturers, developers, and publishers identified by The 1986 Educational Software Preview Guide. Subjects were initially contacted by letter requesting the procedures and criteria used by the subjects to select educational software for publication. The data received from subjects was classified and percentages calculated based upon their compliance with the following criteria (Bitter, 1986):

--Correctness of Content Presentation
--Content Presentation
--Use of Technology
--Integration into Classroom Use
--Ease of Use
--Curriculum Congruence
--Interaction

--Content Sequence/levels
--Reliability
--User Control of Program
--Feedback (General)
--Objectives
--Motivation
--Branching
--Negative Feedback/Help
--Content Modification
--Content Bias
--Teacher Documentation
--User Support Materials
--Color, Sound, Graphics, Animation
--Screen Displays
--Management System

Of the 132 publishers of educational software identified by The 1986 Educational Software Preview Guide, 91 or 69 per cent of the publishers responded. Of the 91 respondents, 49 responded by letter and 32 responded by telephone contact. The participants in this study represented twenty different states and Canada.

After an analysis of the data received from participants, percentages of respondent use for each criterion was calculated:

--Correctness of Content Presentation	9.9%
--Content Presentation	9.9%
--Use of Technology	10.9%
--Integration into Classroom Use	8.8%
--Ease of Use	6.6%
--Curriculum Congruence	10.9%
--Interaction	5.5%
--Content Sequence/Levels	5.5%
--Reliability	5.5%
--User Control of Program	2.2%
--Feedback (General)	3.3%
--Objectives	6.6%
--Motivation	7.7%
--Branching	3.3%
--Negative Feedback/Help	1.1%
--Content Modification	1.1%
--Content Bias	0.0%
--Teacher Documentation	4.4%
--User Support Materials	3.3%

--Color, Sound, Graphics, Animation	5.5%
--Screen Displays	2.2%
--Management System	2.2%

Conclusions

The purpose of this study was to assess the educational criteria used by manufacturers in the development and publication of educational software and compare it to the criteria for the evaluation of software conducted by educators.

Overall, the great majority of educational software publishers do not have a formal or standard set of criteria to guide in software development or employ in the selection of educational software for publication. However, the small number of publishers employing formal or some criteria or policies regarding the development and selection of educational software for publication do not employ the same criteria that educators deem important in the development of educational software.

Similarly, many of manufacturers request that potential programs be submitted to them for evaluation on an individual basis. This evaluation is conducted by some publishers on the basis of that manufacture's individual agenda.

Recommendations

Recommendations for Publishers

The following recommendations are not a comprehensive listing of all the considerations which publishers should take into account when developing and selecting educational software for publication. However, they do represent a first step in the development of a comprehensive approach to improvement of educational software:

1. In consultation with educators and based on current learning research, publishers should develop a standard set of criteria and procedures used to develop and select software.

2. Review boards consisting of educators and programmers should evaluate software before field testing.

3. Field testing products with students and teachers should be conducted before products are marketed. The results of field testing should be included with the package information.

4. Educators should be involved in identifying areas of future software development.

5. Recommendations for integration into curricular areas should be included, along with lesson plans for each product.

In summary, publishers and educators should work together, each contributing their expertise, to advance and improve the quality of educational software.

Future Research:

While the opportunities for research in computer assisted instruction are enormous, the following recommendations must first be addressed before significant improvement of educational software will take place:

1. Survey teachers to find what types of software are needed but not available.

2. Conduct studies to determine the extent to which field testing of educational software is needed to produce high quality educational software.

3. Conduct studies to determine why certain software has a high popularity.

4. Conduct studies to determine what teachers need to implement computers properly in the classroom.

5. Conduct studies that result in a listing of current software that meets all curricular areas.

6. Conduct studies that determine further uses for the computer in instructional settings.

A SELECTED BIBLIOGRAPHY

Becker, Jay. "How Schools Use Microcomputers." Classroom Computer Learning, Vol. 3 (Sept., 1983), pp. 41-44.

Berg, Paul and William J. Bramble. "Computers and the Future of Education." AEDS Journal, Vol. 17 (Fall/Winter, 1983), pp. 101-108.

Bialo, Ellen R. and Lisa B. Erickson. "Microcomputer Courseware: Characteristics and Design Trends." AEDS Journal, Vol. 18 (Summer, 1985), pp. 227-236.

Bitter, Gary and David Wighton. "The Most Important Evaluation Criteria Used Within the Educational Software Evaluation Consortium." (Paper presented to the Educational Software Evaluation Consortium, Menlo Park, California, December, 1986.) Tempe, Arizona: Arizona State University, 1986.

Bork, Alfred. "Education and Computers, The Situation Today and Some Possible Futures." THE Journal, Vol. 12 (Oct., 1984), pp. 92-97.

Burns, P. K. and W. C. Bozeman. "Computer Assisted Instruction and Mathematics Achievement: Is there a Relationship?" Educational Technology, Vol. 21 (Sept., 1981), pp. 32-39.

Cacha, Frances B. "Glamourizing and Legitimizing Violence in Software: A Misuse of the Computer." Educational Technology, Vol. 23 (Mar., 1983), pp. 7-9.

Cohen, Vicki Blum. "Criteria for the Evaluation of Microcomputer Courseware." Educational Technology, Vol. 23 (Jan., 1983), pp. 9-14.

Czechowicz, C. I. "English Basics Part II: Concepts in Language Arts." The Journal of Courseware Review, Vol. 1 (Jan., 1981), p. 10.

Edwards, J., S. Norton, S. Taylor, R. van Dusseldrorp, and M. Weiss. "Is CAI Effective?" AEDS Journal, Vol. 7 (Summer, 1974), pp. 122-126.

Eggers, John R. and John F. Wedman. "The Growing Obsolescence of Computers in Education." Educational Technology, Vol. 14 (July, 1984), pp. 27-29.

Fiske, E. "Computers Alter Life of Pupils and Teachers." The New York Times, April 4, 1984,1, p. 42.

Gallini, Joan. "What Computer-Assisted Instruction Can Offer Toward the Encouragement of Creative Thinking." Educational Technology, Vol. 23 (Apr., 1983), pp. 7-11.

Gagne, R. M. and L. J. Briggs. Principles of Instructional Design. New York: Holt, Rinehart and Winston, 1974.

Goforth, William P. and Paul M. Nassif. "A Comparison of Microcomputer-Based Computer-Managed Instruction (CMI) Software Programs (with an Evaluation Form)." Educational Technology, Vol. 14 (Jan., 1984), pp. 28-30.

Golas, Katharine C. "The Formative Evaluation of Computer-Assisted Instruction." Educational Technology, Vol. 23 (Jan., 1983), pp. 26-28.

Gold, Patricia Cohen. "Educational Software -- New Guidelines for Development." AEDS Journal, Vol. 18 (Fall, 1984), pp. 41-50.

Grabowski, Barbara and William Aggen. "Computers for Interactive Learning." Instructional Innovator, Vol. 29 (Feb., 1984), pp. 27-30.

Hassett, J. "Computers in the Classroom." Psychology Today, Vol. 14 (Sept., 1984), pp. 22-28.

Hechinger, F. "About Education: Computer Software Found Weak." The New York Times, April 20, 1982, 1, p. C4.

Holznagel, Donald C. "Evaluating Software." AEDS Journal, Vol. 17 (Fall/Winter, 1983), pp. 33-40.

Hord, Edwin V. "Guidelines for Designing Computer-Assisted Instruction." Instructional Innovator, Vol. 29 (Jan., 1984), pp. 19-23.

Ignatz, Mila E. "Suggestions for Selecting Science Educa-Software." Journal of Computers in Mathematics and Science Teaching, Vol. 5 (Fall, 1985), pp. 27-29.

Ingersoll, G., C. Smith and P. Elliot. "Microcomputers in American Public Schools: A National Survey." Educational Computer, Vol. 2 (Oct., 1983), p.28.

Jay, Timothy B. "The Cognitive Approach to Computer Course ware Design and Evaluation." Educational Technology, Vol. 23 (Jan., 1983), pp. 22-26.

Jensen, Carl B. "Using What We Have Learned in the Past to Improve the Future of Courseware Design." AEDS Journal, Vol. 19 (Fall, 1985), pp. 29-47.

Kamouri, Anita L. "Computer-Based Training: A Cognitive Framework for Evaluating System's Designs." Journal of Educational Technology Systems, Vol. 12 (Apr., 1984), pp. 287-309.

Kennedy, Patricia H. and Ruth A. Camuse. "Selecting Software on a Budget." Instructional Innovator, Vol. 29 (Jan., 1984), pp. 24-28.

Kulik, J. A., C. C. Kulik and P. A. Cophen. "Effectiveness of Computer-Based College Teaching: A Meta-Analysis of Findings." Review of Educational Research, Vol 50 (Apr., 1980), pp. 525-544.

Lathrop, Ann, ed., The 1986 Educational Software Preview Guide, California Department of Education (Sacramento, 1986).

Lawton, Johnny and Vera T. Gerschner. "A Review of the Literature on Attitudes Towards Computers and Computerized Instruction." Journal of Research Development in Education, Vol. 16 (Jan., 1982), pp. 29-32.

Morris, John M. "Computer-Aided Instruction: Toward A New Direction." Educational Technology, Vol. 23 (May, 1983), pp. 12-15.

National Education Association. A Teacher Survey NEA Report: Computers in the Classroom. Washington D.C., 1983.

Neibauer, Alan. "The Computer Literacy Myth." THE Journal, Vol. 12 (Feb., 1985), pp. 88-90.

Norris, Cathleen M. and Barry Lumsden. "Functional Distance and the Attitudes of Educators Toward Computers." THE Journal, Vol. 11 (Jan., 1984), pp. 129-132.

Otte, Richard B. "Courseware for the 80's." THE Journal, Vol. 12 (Oct., 1984), pp. 89-91.

Papert, Seymour. Mindstorm--Children, Computers and Powerful Ideas. New York: The Harvester Press, 1980.

Pliske, Daniel B. and Joseph Psotka. "Exploratory Programming Environments for Designing ICAI." Journal of Computer-Based Instruction, Vol. 13 (Spring, 1986), pp. 52-57.

Poore, Jesse H. and John W. Hamblen. "Why Computer-Based Education Is Making Slow Progress: An Analysis of Costs and Other Impediments." AEDS Journal, Vol. 17 (Spring, 1984), pp. 42-52.

Roblyer, M. D. "Toward More Effective Microcomputer Courseware Through Application of Systematic Instructional Design Methods." AEDS Journal, Vol. 17 (Fall/Winter, 1983), pp. 23-32.

Roblyer, M. D. "When is it 'Good Courseware'? Problems in Developing Standards for Microcomputer Courseware." Educational Technology, Vol. 21 (Oct., 1981), pp. 47-54.

Salisbury, David F. "How To Decide When and Where to Use Microcomputers for Instruction." Educational Technology, Vol. 14 (Mar., 1984), pp. 22-24.

Schiffman, Shirl S. "Software Infusion: Using Computers To Enhance Instruction. Part One: What Does Software Infusion Look Like?" Educational Technology, Vol 26 (Jan., 1986), pp. 7-11.

Shoemaker, H. and H. Holt. "The Use of Programmed Instruction in Industry." in Teaching Machines and Programmed Learning. 2nd Ed., R. Glaser (Ed.), National Education Association, Washington, D.C., 1965.

Skinner, B. F. The Technology of Teaching. New York: Appleton-Century-Crofts, 1968.

Smith, Patricia L. and Barbara Ann Boyce. "Instructional Design Considerations in the Development of Computer-Assisted Instruction." Educational Technology, Vol. 14 (July, 1984), pp. 5-11.

Steffin, Sherwin A. "Computer Simulations: A Key to Divergent Thinking." Media and Methods, Vol. 7 (Oct., 1981), pp. 12-13.

Steffin, Sherwin A. "A Suggested Model for Establishing the Validity of Computer-Assisted Instructional Materials." Educational Technology, Vol. 23 (Jan., 1983), pp. 7-10.

Steinberg, Esther R. "Reviewing the Instructional Effectiveness of Computer Courseware." Educational Technology, Vol. 23 (Jan., 1983) pp. 17-19.

Tamir, Pinchas. "Current and Potential Uses of Microcomputers in Science Education." Journal of Computers in Mathematics and Science Technology, Vol. 5 (Winter, 1986), pp. 18-28.

Taylor, R. P. (Ed.). The Computer in the School: Tutor, Tool, Tutee. New York: Teachers College Press, 1980.

Tetenbaum, Toby J. and Thomas A. Mulkeen. "Computers as an Agent of Change." Computers in the Schools, Vol. 2 (Winter, 1986), pp. 91-103.

Toffler, Alvin. The Third Wave. New York: Bantam Books, 1981.

Truett, Carol. "Field Testing Educational Software: Are Publishers Making the Effort?" Educational Technology, Vol. 14 (May, 1984), pp. 7-12.

Tyler, John G. "Your Prescription for CAI Success." Instructional Innovator, Vol. 28 (Feb., 1983), pp. 25-26, 40.

Vinsonhaler, J. F. and R. K. Bass. "A Summary of Ten Major Studies on CAI Drill and Practice." Educational Technology, Vol. 12 (Aug., 1972), pp. 29-32.

Wade, T.E. "Evaluating Computer Instructional Programs and Other Teaching Units." Educational Technology, Vol. 20 (Nov., 1980), pp. 32-35.

Watt, D. "Computers in Education." Popular Computing, Vol. 7 (Apr., 1983), pp. 83-84.

Waugh, Michael L. and Dawn Currier. "Computer-Based Education: What We Know and Need to Know." Journal of Computers in Mathematics and Science Technology, Vol. 5 (Spring, 1986), pp. 13-15.

Wedman, John. "The Future of Computers in Education: What are the Right Questions." THE Journal, Vol. 11 (Sept., 1983), pp. 147-148.

Wedman, John and Marvin Heller. "Concerns of Teachers About About Educational Computing." AEDS Journal, Vol. 18 (Fall, 1984), pp. 31-40.

APPENDIX A

INITIAL CONTACT LETTER

Oklahoma State University

CLEARING HOUSE OF INFORMATION
ON MICROCOMPUTERS IN EDUCATION

STILLWATER, OKLAHOMA 74078-0146
108 GUNDERSEN
(405) 624-6254

October 13, 1986

Dear Sir or Madam:

Your company was identified by the 1986 Educational Software Preview Guide developed by the Educational Software Evaluation Consortium as publishing software of high quality. In my position as editor of the CHIME Newsletter and Oklahoma State University's representative to the Educational Software Evaluation Consortium, I am conducting a study of procedures and criteria used by software publishers and their review boards to select educational software for publication from contributors. The results of this study will be published in the CHIME Newsletter and presented to the members of the Consortium.

I ask that you please send me the procedures and criteria used by your company and review board to select educational software for publication. At no time will individual publishers be identified by software or by name with respect to the criteria used. However, cooperating publishers will be listed as participants in the study.

If you have any questions regarding the above request, please do not hesitate to contact me. Your cooperation is appreciated by CHIME and the members of the Educational Software Evaluation Consortium.

Sincerely,

Susan Kies Roth
Editor
CHIME Newsletter

CENTENNIAL
DECADE
1980 • 1990

APPENDIX B

SECOND CONTACT LETTER

108 Gundersen
Stillwater, OK 74078

Dear Sir or Madam:

Please send me the procedures and criteria used by your company to develop and select educational software for publication.

Susan Roth
Editor of CHIME

APPENDIX C

PARTICIPANTS IN THE STUDY

PARTICIPANTS IN THE STUDY

Addison-Wesley Publishing Co.
Advanced Ideas, Inc.
Aldus
Ann Arbor Softworks
Apple Computer
Artworx Software
Barron's Educational Series, Inc.
Baudville
Beagle Brothers
Behavioral Engineering
Borland International
Broderbund Software
Bytes of Learning
C & C Software
CBS Interactive Learning
Classroom Consortia Media, Inc.
Conduit
Cygnus Software
D.C. Heath & Co.
Davidson & Associates
DesignWare, Inc.
Developmental Learning Materials
Didatech Software
Earthware Software Services
Educational Publications Concepts
Educational Materials & Equipment
Educational Activities
Electronic Arts
Encyclopaedia Britannica
Field Publications
Floppy Enterprises
Focus Media, Inc.
Freesoft Co.
Gamco Industries
Gessler Educational Software
Grolier Electronic Publishing
Hartley Courseware, Inc.
Hayes Microcomputer Products, Inc.
Hi Tech of Santa Cruz
High Technology Software Products
Holt, Rinehart and Winston
Houghton Mifflin Company
HRM
Interlearn
J & S Software
JMH Software of Minnesota, Inc.
Kent Publishing Co.
Koala Technologies Corp.
Krell Software Corp.

PARTICIPANTS (Continued)

Lawrence Hall of Science
Learning Technologies, Inc.
Learning Well
Lightspeed Software
Logo Computer Systems, Inc.
Lotus Development Cop.
Macmillan Publishing Co.
Mark Davids
Marshware
MECC
Media Materials, Inc.
Microcomputer Workshops/CBS
Midwest Publications Co.
Milliken Publishing Co.
Mindplay
Mindscape, Inc.
Quality Educational Designs
Radio Shack
Rand McNally & Co.
Satellite Software International
Science Research Associates, Inc.
Scott, Foresman & Co.
Sir-Tech
Society for Visual Education
Sorcium/IUS Micro Software
South Coast Writing Project
South-Western Publishing Company
Spinnaker Software
Springboard Software, Inc.
Strategic Simulations, Inc.
Sublogic
Sunburst Communications
Synergistic Software
Telos Software Products
Terrapin, Inc.
Tom Snyder Productions
Unison World, Inc.
United Software Industries
Vernier Software
Versa Computing, Inc.
Wadsworth Electronic Pub. Co.
Walt Disney Non-Theatrical Co.

APPENDIX D

NECESSARY ATTRIBUTES TO CONSIDER IN DESIGNING COURSEWARE FOR THE MICROCOMPUTER

TABLE I

NECESSARY ATTRIBUTES TO CONSIDER IN DESIGNING COURSEWARE FOR THE MICRCOMPUTER

Generic to Instruction Design:

- -Target audience specified
- -Learner entry competencies specified
- -Rationale, goals, and objectives specified
- -Objectives stated behaviorally
- -Objectives stated in terms of the learner
- -Objectives include higher-order skills
- -Learners informed of objectives
- -Range and scope of content adequate to achieve program's intents
- -Preinstructional strategies used:
  - Pretests
  - Advance Organizers
  - Title at beginning of unit
- -Instructional test formatted for easy reading
- -Concept learning employed in instructional approach
- -Vocabulary used appropriately for learner
- -Graphics embedded in content
- -Graphics used appropriately
- -Demonstration of the exercise provided
- -Teacher's Manual provided
- -Instruction clearly stated for student
- -Evaluation Components provided

TABLE I (Continued)

Necessary for the Design of Courseware:

-Curriculum role used:

Adjunct

Mainline

Management only

Other

-Mode of interaction employed:

Drill and Practice

Tutorial

Game

Simulation

Problem Solving

-Student sequenced through the content:

Nonlinear

Varied by teacher/student

-Instructional text formatted for screen display

-Graphics embedded in the content

-Graphics used appropriately

-Cues and/or prompts used

-Action occurs on the screen

-User control granted to learner

-Computer-Managed Instruction employed

-Feedback used appropriately

-Records stored on magnetic devices for future retrieval

TABLE I (Continued)

-Content designed to be altered
-Random generation used
-Packaging designed for component parts
-Teacher's Manual and Student Manual provided
-Technical design used:
Quick response time
Quick loading time

2

VITA

Susan Kies Roth

Candidate for the Degree of

Doctor of Education

Thesis: EDUCATIONAL CRITERIA USED BY MANUFACTURERS IN THE DEVELOPMENT AND PUBLICATION OF EDUCATIONAL SOFTWARE

Major Field: Education

Biographical:

Personal Data: Born in Aurora, Illinois, July 1, 1956, the daughter of Neal J. and Clara F. Kies. Married to Timothy J. Roth on March 24, 1979. Two children, Gregory G. Roth born on January 15, 1983 and Karen M. Roth born on July 10, 1985.

Education: Graduated from Rosary High School, Aurora, Illinois in June, 1974; received Bachelor of Science degree from the University of Illinois, Urbana, Illinois in May, 1978; received Master of Science degree from Oklahoma State University, Stillwater, Oklahoma in May, 1983; completed requirements for the Doctor of Education degree at Oklahoma State University, Stillwater, Oklahoma in May, 1987.

Professional Experience: Classroom Teacher, Mt Zion High School, Mt Zion, Illinois, August, 1978 to June 1979; Classroom Teacher, Washington Junior High School, Washington, Illinois, August 1979 to June 1981; Classroom Teacher, Cushing Middle School, Cushing, Oklahoma, August, 1981 to June, 1982; Graduate Assistant, Oklahoma State University, Stillwater, Oklahoma, June 1982 to May 1987; Editor of The CHIME Newsletter, Oklahoma State University, Stillwater, Oklahoma, January, 1985 to present.