DOCUMENT RESUME

ED 277 169 EC 191 374

AUTHOR Weisgerber, Robert A.; Blake, Patricia L.
TITLE The Evaluation and Selection of Instructional Software for Use with the Learning Disabled. CREATE Monograph Series.
INSTITUTION American Institutes for Research in the Behavioral Sciences, Palo Alto, Calif.
SPONS AGENCY Special Education Programs (ED/OSERS), Washington, DC.
PUB DATE Sep 84
GRANT G008302861
NOTE 47p.; For the other monographs in this series, see EC 191 371-376.
PUB TYPE Reports - Descriptive (141)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Computer Assisted Instruction; *Computer Software; Computer Software Reviews; Courseware; *Educational Technology; Elementary Secondary Education; *Instructional Material Evaluation; *Learning Disabilities; *Media Selection

ABSTRACT

Fourth in a series of six monographs on the use of new technologies in the instruction of learning disabled (LD) students, this paper explores issues related to the evaluation and selection of instructional software for LD students. Topics discussed include the following: (1) criteria for instructionally useful software (e.g., flexibility and unique display capabilities); (2) rationale for improved selection processes; (3) factors influencing selection (e.g., nature of documentation that accompanies the program); (4) producers of software; (5) evaluation forms and review procedures. A final section discusses elements of a comprehensive evaluation and selection instrument, including types of information to be gathered (vendor/program data, program description, instructional design features, and appropriate applications). (JW)

Research and Educational Applications Center for the Handicapped
Occasional Papers from
REACH

The Evaluation and Selection of Instructional Software for Use with the Learning Disabled

Robert A. Weisgerber
Patricia L. Blake

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

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

CREATE Monograph Series

September 1984

The research reported herein was performed pursuant to Contract No. G008302861 with the U.S. Department of Education. Contractors undertaking such projects under government sponsorship are encouraged to express freely their professional judgment in the conduct of the project. Points of view or opinions stated do not, therefore, necessarily represent official Department of Education position or policy.

Table of Contents

Introduction

The market for educational software is very large and growing. Quality Educational Data, Inc., (Denver, CO) reports that in 1981-82, for example, there were 3,989 Apple computers in the schools, a year later there were 15,773 and by 1983-84 there were 36,781 (Apple has 54% of the market). Indeed, 75% of the schools in the country now have microcomputers. Where there is that kind of growth in computers there is automatically a powerful attraction for software development. By 1983-84 there were 1,947 separate companies who considered themselves educational software publishers and some 153 distributors of their materials. The volume of business connected with microcomputers has grown so rapidly and in such diverse ways that there are now 285 companies providing consultants and market research services.

Given this remarkable growth and the certainty that it will continue in the foreseeable future, every effort should be made to open lines of communication between publishers and educational consumers. This would allow the publishers to make their products known and the consumers can be afforded the benefits that come from selecting high quality, instructionally targeted materials for use in their instructional programs.

This paper is intended to serve as a baseline for Project CREATE (Center for Research in the Applications of Technology in Education), a baseline from which we can begin to develop prototypic selection and evaluation procedures to (1) help instructional staff to make effective choices among alternative materials, (2) encourage appropriate use of software in the curriculum so that their maximum cost-effectiveness is realized, and (3) provide feedback to publishers about features that they might want to incorporate as they develop new software for the educational market.

CREATE's efforts will involve an analysis of the "attractive" properties of selected software and the development of evaluation forms that can be used to systematically identify the qualities that are important for learning disabled children.

1. <u>Instructional Usefulness of Software</u>

The instructional usefulness of software is generally defined by two criteria: (1) how well a program addresses the specific needs of students, and (2) how well a program fits into the school's curriculum. As simple as this statement appears, only a small number of programs actually satisfy the definition. This is not to say that educational software that does meet the criteria for instructional usefulness cannot be "good" software. The term, good software, often is taken to mean that it is technically sound, easy to learn, or fun to use, but as an effective instructional tool it must be more than that.

As suggested by Chambers and Sprecher (1983), a primary requirement for effective educational software is that it be developed by individuals who are knowledgeable in the subject matter content, computer technology, learning theory, and motivation of learners. In other words, the developer should understand and creatively plan for those variables that will (1) enable a teacher to reach a wide range of learners with varied needs, and (2) build into the program the flexibility necessary to correspond with different instructional objectives. CREATE has identified several features of creative programming that are desirable. These include: flexibility, ability to modify the program, exploitation of the computer's unique display capabilities, gaining and holding attention, real-time feedback, "patient" tutorial potential and learner control.

In addition to the instructional aspects of software, programs that permit recordkeeping, report generation, and profile building are valuable because of their usefulness for administrative monitoring of individualized instruction.

The individual needs of a student may vary from enrichment to remediation. For educational software to effectively address the special needs of handicapped learners, the software package must be flexible enough to construct "bridges" between various levels of learning abilities. Since a computer is capable of being programmed to provide such bridges, educators/

programmers are able to utilize this tool in ways that complement more traditional methods of education.

Flexibility. Software flexibility can be thought of as a means by which a program adjusts to the way an individual "best" learns. Among other things, flexibility includes adjustment of the rate of presentation, the size of the text, and the capacity to work with alternative input devices. Learning disabled students, for example, may need to slow the rate of text presentation to effectively discriminate stimulus features such as letter shapes. Students with poor motor control may need to adjust the rate of presentation to match their response rate. Along with adjusting the rate of presentation, the capability of adjusting type size can be beneficial. In addition to being useful for the visually impaired, enlarging the type size would help to focus a learner on a specific aspect of a task. In this way, the computer display can function as a "place-marking card," where relevant stimuli are isolated from textual distractions in the display.

Another important aspect of flexibility can be seen in those software programs that "branch" to a learner's ability level. Placement may initially be arbitrary (as a "best guess") but is then adjusted on the basis of performance by the learner in comparison to pre-set criteria. The rationale for branching is that if a learner repeatedly gives incorrect responses, the program will branch to a less difficult level of instruction. Only when the student has mastered the content to criterion will the program branch to a more difficult level.

Since most programs determine whether a student has mastered a level based on correct performance, the program is not directly assessing how a student is learning. Senf (1983) points out that performance-oriented programs are particularly limiting for learning disabled students because they do not determine why a student is giving incorrect responses. As Senf suggests, a student may be having difficulty with a certain reading task because he or she is "analyzing each word phonetically by syllable." Clearly, sending the student back to a less difficult level of instruction or review of the same material would not in itself teach the student to read efficiently. Instead, the program should assess incorrect responses by dentifying the underlying prerequisite skills, then branch the learner into review or training of these skills.

Suppose, for example, a program is designed to record error rates in eading of line copy, and it is determined that most errors occur at the ime of line change (where the eyes must return to the left margin and ontinue reading smoothly). The program might branch the learner into a training mode" where a window highlights the text to be read on a line-y-line basis. The learner could then refine his or her vision skills, such s coordination of foveal detail with peripheral awareness. In this example, ndividualized instruction means that a teacher could set instructional arameters so that they are matched with the appropriate parameters for the ndividual student. That is, a teacher could "tell" the program to branch o the vision skills module under one set of conditions or to branch ahead f the criteria for mastery are satisfied by that particular student.

Modification of the Program. The ability to modify a program allows a eacher to tailor the instructional content. As just mentioned, this can be ccomplished through adjustment of the instructional parameters or by eacher selection of the degree of learner control, or through a provision or inserting or deleting information.

The modification of program content is usually realized through the use f authoring systems. An authoring system is a program that allows users to reate individualized lessons without programming knowledge. For example, he program could provide a "user-friendly" way to design a quiz, present aterial for drill and practice, or create a vocabulary list in which words an be added or deleted. Authoring systems vary in both sophistication and rice. For example, a program such as "The Adaptable Skeleton" (Micro Power & Light Company, $34.95), allows users to create lessons within a multiple hoice format. In contrast, a program such as "Assisted Instructional Development System" (Skillcorp Software, $395.00), includes such capabilities as branching at three levels, timed lessons, and a management system that allows generation of reports.

It is becoming more common to see educational software companies include "mini-authoring" systems with their programs (e.g., Advanced Ideas). The capacity for quick and easy insertions is ideal for learning disabled students. A teacher could, for example, develop spelling lists based on particular letter patterns to facilitate word recognition as well as spelling practice. A benefit of using a mini-authoring system is that program adjustments can be made that enable individualized instruction while maintaining the foundation of the program (i.e., the motivational properties of an instructional game).

Unique Display Capabilities. There has been much criticism of the early use of computers as mere "page turners." Such criticism appears justified when considering the potential of using this medium in unique applications. Aside from the "real-world," only with a computer can students rotate a three-dimensional object or simulate driving a car. Unlike a static chalkboard or text, the computer can be programmed to be dynamic.

To illustrate, consider a hypothetical program where a student is taught to arrange a group of sentences into an expository paragraph. The objective would be to locate the sentence that contains the main idea and organize the sentence components so that they relate to the main idea (topic sentence) in proper sequence.

After a learner completed a specified number of lessons, he or she would play a pinball game with the game balls earned during the lesson(s). (The number of turns would be determined by the number of balls earned during the paragraph lesson.)

Figure 1 is an illustration of how this program might appear on a computer display. The left side of the screen would be used for displaying the disorganized paragraph. The learner would select the sentences to build an organized paragraph on the right side of the screen. The number of game balls earned during a lesson would be displayed at the top of the screen. The pinball game, as illustrated in Figure 2, would be designed to improve

Figure 1

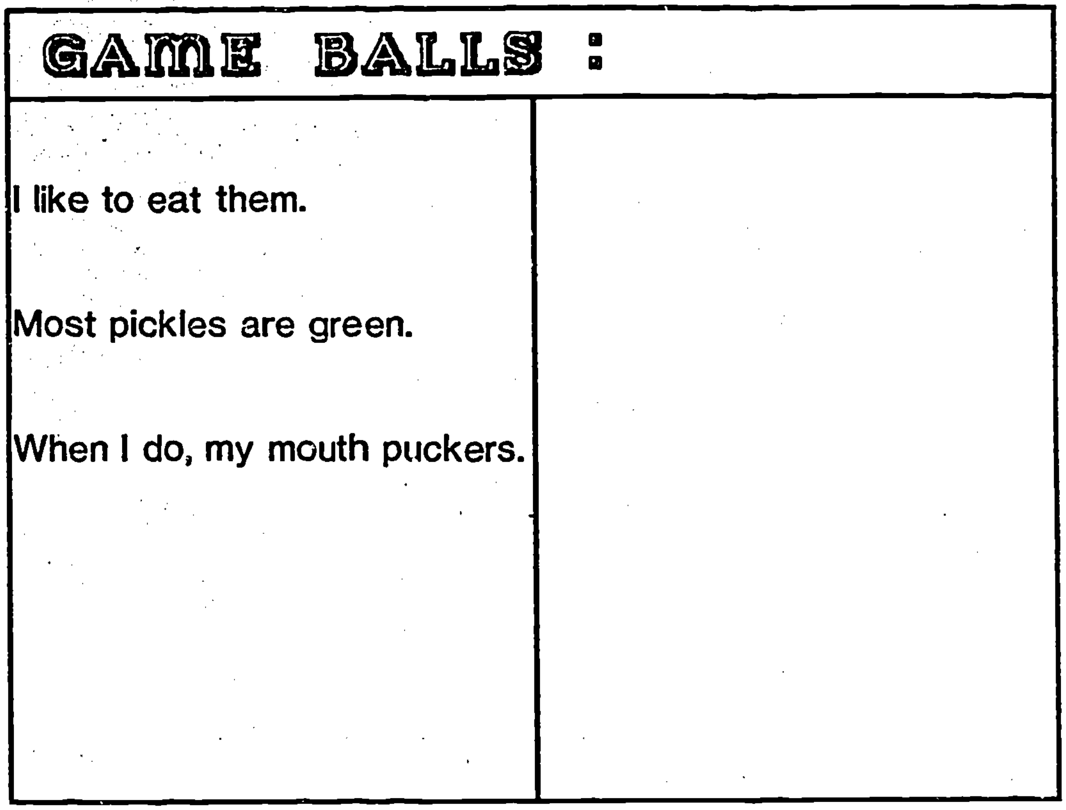
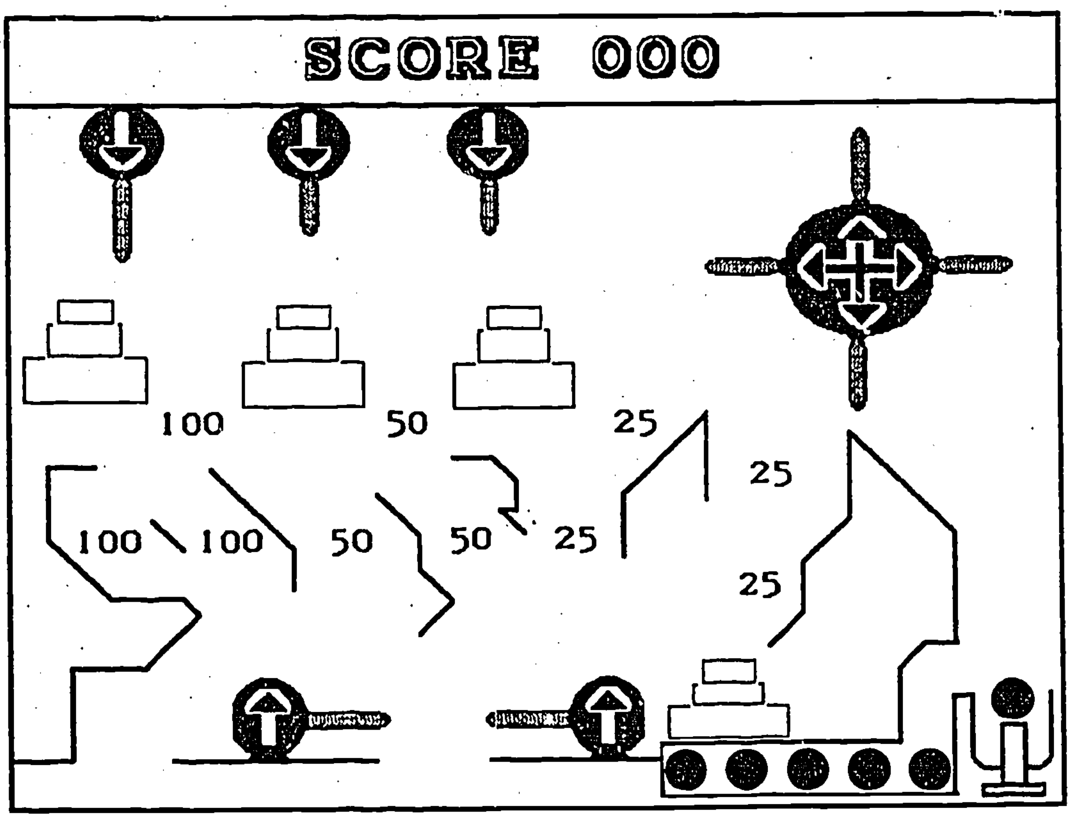


Figure 2



eye-hand coordination by instructing the learner to use the "flippers" to direct the ball and keep it in motion. The speed of the ball could be set by the learner. Let's see how this program might work:

- The lesson would begin with the presentation of a main idea sentence and two or more supporting sentences displayed in random order. The learner would be instructed to bring the sentences (one at a time) to the organizer area and place them so that they relate properly to the main idea.

- By using the cursor keys on the keyboard (or by using a joystick or graphics tablet) the learner would "pull" the sentences one-by-one into the paragraph organization area.

(The simplicity in manipulating content makes the computer not only easy to use by individuals who have writing difficulty but it is exemplary for showing how components relate to a whole.)

- The learner would earn a game ball (for play after the lesson) for choosing the proper sequence for each paragraph.

- Incorrect responses would cause the program to "coach" the learner by highlighting the critical phrase in the sentence, e.g., tastes sour), and asking the learner for answers to key questions (e.g., "<u>What</u> tastes sour?")

- The instructional parameters would be flexible so that a teacher could adjust the maximum number of "coached" responses allowed to earn a game ball.

The ability to manipulate and move objects provides a sense of control over the environment. Interacting with and commanding a computer is a powerful and entertaining experience. This is particularly true for children who feel little control over how well they learn. There have been many reports of handicapped learners improving their self-concept through the ego-boost they get from mastering a computer (Hagen, 1984; Turkel & Podell, 1984; Loebl & Kantrov, 1984).

Interest and Attention. To gain a learner's attention programs often use a beep or bell sound. While this may be an effective means for gaining attention, holding attention is more difficult. Using game show sounds that are meant to suggest, "This is only the beginning . . . . Stay tuned for more excitement," can provide a sense of anticipation. ("What will I get if I win?"). Programs that have mini-authoring systems could allow students to select their own graphics and compose their own music to accompany their lessons. It could be quite exciting to be rewarded with one's own creation.

The bright colors available on some computer systems are attention-catching and eye-pleasing. There is something satisfying about seeing a representation of a yellow flower, or the darkness of space, as we expect to see them. If a school has only monochrome displays, the graphics should be as uncluttered and distinctive as possible.

As Jerome Bruner put it, "The work of children is play." Particularly effective is the use of animation in software for the elementary grade levels. Animation is familiar to young children and it is associated with fun and entertainment. When a cartoon character appears on the computer screen and says "Let's spell a word!," it seems like fun.

The quality and style of graphics may be a more critical concern for students at the secondary level. Primitive looking graphics can be insulting to young adults. Programs aimed at the secondary grade level should provide graphics that are age-appropriate if they are to gain and maintain attention.

Real-time Feedback. Turkel and Podell (1984) in discussing the learning environment of Logo, give an excellent description of the advantage of real-time feedback:

The immediate feedback of a design on the screen enabled students to modify their instructions to the computer. By making modifications, students could change what the turtle was doing; this control provided the motivation to continue with the activity (p. 261).

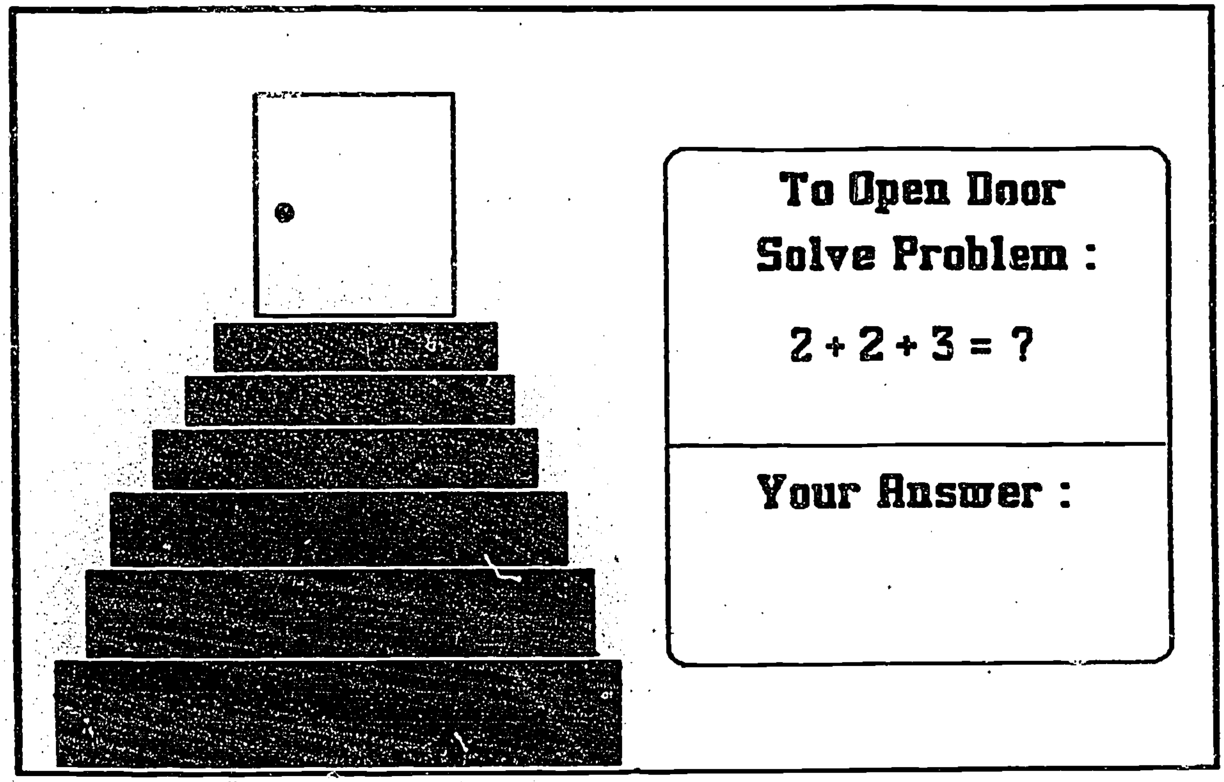
Instructional simulations, which are often presented in the form of problem-solving software, are ideal for maintaining Logo-like feedback and control within a variety of structured instructional environments. That is, a student can make decisions based on variables that the student controls.

For example, a student may be simulating the role of an American Indian who plans to sign a peace treaty with a representative of the United States government. In traveling to the treaty-signing, he or she could encounter a range of barriers necessitating various decisions. These decisions lead to consequences, some of which could impact on the meeting. Cause and effect relationships become immediately apparent. If a strategy proves ineffective the student might call for help to find out why. Mistakes can be explained as soon as they occur and before they are repeated. Similarly, successes are rewarded immediately to ensure that they will be repeated.

One of the real strengths of educational software is its potential for various feedback modes. This strength is especially realized when the reinforcement schedules are consistent with learning theory and the rewards are interesting and different enough to motivate and hold attention.

To illustrate how a program might use a variety of forms of feedback, consider a hypothetical math program. In this example, the student plays the part of a "math detective" who must find the solutions to a series of math problems in order to earn honor, success, and computer-money (points that add up in the "bank"). The learner moves through a series of rooms where, by correctly following certain mathematical operations, "new doors" are opened until the final goal is reached. (See Figure 3)

Figure 3



A possible sequence is as follows:

- The game begins with the student/detective receiving an assignment: "Solve the mysterious case of the missing--" (e.g., cat, ring). The learner would then be instructed to search the rooms for the missing item. In order to enter a new room the learner must correctly solve a math problem.
- The student would receive randomly generated feedback (e.g., Good Job!; That's Right--Great) and graphics (a cat's toy might appear on one of the steps) as rewards for completing steps correctly.
- The first two incorrect responses at each step might be met with a hint (such as highlighting the step that resulted in the incorrect solution) and a message to "try it again." After the next unsuccessful try, the problem could be completed step-by-step (by the program) along with the message, "Now you try it."

- All of the math operations would be highlighted and checked number-by-number so that the student could see where they were making an error.
- Further errors might indicate that the student needs work on number relations, activating branch for special instruction . . . "How to be a math detective."
- The difficult problem would appear again later in the session.
- After correctly completing a certain segment of the math problems, the student would be rewarded (intermittently) with both music and graphics.

By pressing a designated key a learner could receive help or clues at any time. The type of math problems and time allowed to complete them would be determined by the student. More earned points would be allotted to shorter solution times and harder problems.

Handicapped learners need immediate feedback that is both non-judgmental and encouraging. As pointed out by Torgeson (1977), the learning disabled are typically passive in their approach to learning because they have received little or no payoff. The computer has tremendous potential for motivating and helping handicapped learners feel in control of learning. A characteristic of "good" software is that it allows a student to "attack" a lesson without fear of failure by offering plenty of hints and cues to minimize frustration.

Learner control. Selecting an activity is one way students can assess their own strengths and weaknesses. It is important for students to discover that "I'm good at this" and "I don't like that task." A well designed program will motivate students to try more difficult problems without fear of failure. A motivated learner may be determined to repeat an activity until his or her own criteria for success are met. "I'm going to win this time"; "Next time I won't type in a 'p' because I know the right answer is a 'b.'"

Success for learning disabled students success may be, in part, knowing that learning itself can be controlled. A student may know, for example, that when text moves across the screen at a slower rate, he or she can solve the problem correctly.

Educational management. In addition to instruction, programs can perform tasks that relate to instructional planning. Computers are excellent for accumulating information. Programs that update current records of students' strengths and weaknesses can build a profile for diagnostic purposes or aid a teacher in planning individualized instruction.

Daily and cumulative reports are helpful to both the teacher and the student. A teacher can look at daily performance data and track a student's progress over time. For learning disabled students, the charting of skill development is one way of demonstrating that learning is actually taking place.

Identifying software that is "good" is not easy. Some "showy" features of the software may dominate the child's or teacher's impression of it, but other educationally important features may be missing. To help in selecting software that is cost effective, procedures must be used that are systematic and objective.

2. Rationale for Improved Selection Processes

Today's schools are faced with difficult choices about how to spend their resources most effectively to achieve quality education. Computers represent an exciting but potentially expensive investment for them, and the potential for waste in expenditures is a real one. For example, the rapidity with which the computer field is changing means that some machines will become obsolete and be replaced by faster and more versatile machines. What happens to the investment that has been made in the software for the old machine? Is it still usable on the new machine? Clearly, the estimated "life" of materials can be significant to the wise use of the school's limited financial resources.

A similar case can be made for investing in software that can be used in more than one way. Obviously, if it does "double duty" then its value is marginally greater, even when its expected "life" is relatively short. One example would be the adaptation of "regular" instructional software that has the potential to be used effectively with the handicapped. Perhaps a program for teaching spelling can also be used to teach eye-hand coordination in a special needs student. Or a program that is designed to teach math operations (adding or subtracting four-digit numbers) could be used to teach scanning by rows and columns to an individual who has trouble with horizontal and vertical spatial relationships.

Another way in which maximum cost-benefits can be realized is to prioritize on the acquisition of software in areas that teachers feel are critical to remediation of learners' problems or rapid advancement in a subject matter. That is, if it can be demonstrated that there is a potential for "breakthrough," as might be possible by involving an autistic child in a particular learning game, then it is hard to deny the value of the material. Certainly, frequent teacher needs and learner needs must be given extra weight in the selection process.

As school districts begin computer-supported instruction there is quite often a nucleus of dedicated and well-informed individuals who lead the movement, or, if it is a small school, the impetus can come from one person. Selection practices in this early adoption stage often tend to reflect the interests and biases of the individual or individuals involved. Thus some subject matter areas will be relatively "rich" in programs while others will be "poor." Similarly, there is sometimes an emphasis on one aspect of computer use, such as creative expression with LOGO, which reflects the perspective of the individual who is nominally in charge.

Fortunately, there is increasing recognition that school policy needs to be established to guide the selection process. The policy should not be cumbersome, overly restrictive, or time consuming to implement, but should be set forth, in writing, in such a way that teachers are encouraged to become involved. One method for encouraging teacher involvement is to have them participate in setting the policy. Another is to provide a stipend in the summer for a team of teachers to engage in systematic review and evaluation of materials that can enhance the program in the coming school year.

According to the Software Merchandising Directory, December 1983 (Eastman Publishing, Encino, CA), the top ten "best sellers" in the educational marketplace are:

Bank Street Writer	Word Processing	Broderbund
Bumble Plot	Learning Program - Number Pair Frequency Through Plotting	The Learning Company
Facemaker	Elementary Programming Through Games	Spinnaker
Gertrude's Puzzles	Shape/Color Patterns	The Learning Company
Master Type	Typing Skills	Scarborough/Lightning
Snooper Troops I	Detective-Type Games Library Skills Maps, Vocab	Spinnaker
Snooper Troops II	Reasoning Info Classification Note Taking	Spinnaker
Story Machine	Language Arts-Writing Sentence, Paragraphs	Spinnaker
Type Attack	Typing Arcade Approach	Sirius
Typing Tutor II	Typing Monitors Keyboard	Microsoft

Even a casual inspection of this list shows the high priority being given in the schools to programs that (a) call for word use and expression, (b) have game-like characteristics, and (c) try to encourage thinking.

3. Factors that Influence Selection

Realistically, the most important limiting factor in selection is the school budget. Only a small percentage of the school district's operational expenses can be devoted to the purchase of software, so the implication is clear—either restrict purchases to the very best and most needed items, supplement the acquisition budget through fund-raising efforts of one kind or another, or seek out public domain materials. Materials in the public domain can be elusive, however, because even though some may be quite well done, their authors have no marketing budget and knowledge of their availability usually depends on word-of-mouth or bulletins circulated by "cooperative" user groups.

When high quality materials are identified, there is a natural inclination to want more than one copy so that the program can be used simultaneously in more than one class or school. This brings up another problem. One copy may be affordable, but multiple copies may not. It should be noted that copyright law <u>does</u> apply to author's and publisher's rights in the case of software. Piracy, while known to exist, is not a viable option for the schools. Special purchase agreements can take the need for multiple use into account, either through discounts on additional copies or through negotiated rights to distribute the software through a local area network (LAN) from a central location.

Another factor that influences the selection and utilization of software is the nature of the documentation that accompanies the program. Is the software "ready to go" or is there a need for considerable study by teachers to implement it smoothly and effectively? If this question is not addressed by the buyer before purchase, the material may never reach its full potential for use in the school. It is disappointing but true that poorly written documentation has kept some otherwise useful software from receiving the attention it deserves. To reduce dependence on accompanying materials that may be of questionable quality, there is a trend toward making the program itself "teach" the user through "help" options. Of course, there is a limit to which this can be done without using up the available memory in the machine. As time goes on, newly emerging hardware with much greater storage capacity and software that is integrated will counterbalance the requirement that users be taught through separate print materials.

An obvious factor, and at least partly a function of the initial district decisions, limits selection to the range of software that will operate on the school's machines. This is not an insignificant factor, for unlike a book that can be substituted when a new edition comes out, the program must be compatible. While most of the very popular software is available for more than one type of computer, there are still many fine programs that can only run on equipment that the school doesn't possess--or exists only in a school other than the one in which the software is needed. It is the availability of hardware that has caused such a strong domination of the school software market by Apple. Recent promotional policies by IBM (particularly at the higher education level) are challenging that domination.

Assuming compatibility and the establishment of priority needs, a factor that should not be underestimated is the importance of critically appraising the quality of instructional design in a given program. Whether it is intended as a central part of the curriculum, such as teaching touch-typing in a typewriting class, or as adjunct material, such as the use of "brain buster" puzzles to occupy students productively during earned "free time," there is a need to judge whether the program will hold student interest and if it is technically correct. Regardless of whether evaluation is teacher-based, committee-based, or student-based, it should involve systematic evaluation. More will be said about this in a subsequent section of this paper.

4. Producer Variables and Sources for Software

Teachers and schools have not yet developed a high set of standards demanding polish in microcomputer software in the same way they have for textbooks. While printed materials that are commercially acquired must be "accepted" by review committees using criteria that are tough for publishers to comply with, software still enjoys a kind of forgiveness and an over-looking of deficiencies among many districts. As the market fills, however, the reliability of the publisher in using high standards during product development will become much more important.

The tremendous amount of variation in the quality of education software is paralleled by variation in the size and experience of the producing organizations. That is not to say that "bigness" of the firm is necessarily correlated with the "goodness" of the program, but rather that if the development investment is large enough, the purchaser may more reasonably assume a greater amount of internal and external review to find "bugs" prior to publication than is usually the case for an individual operating independently.

Major developer-publishers of software include divisions of traditional educational publishers such as Addison-Wesley, Silver-Burdett, and Scott Foresman. With some exception, the strategy that is favored by these large firms is to develop a block of curriculum materials that are demonstrably correlated with the scope and sequence of "mainstream" instruction in the language arts and math areas, and more recently in social studies and science. These materials are aimed at the majority of school children who are "on track" in their progress through the grades and course work. Apart from several curriculum adaptation projects supported by the Office of Special Education Programs, U. S. Department of Education, these large publishers are not concentrating on the special education market, for this thin market is not able to support the magnitude of investment that they make in royalties, duplication, packaging, marketing, including brochures, "free" examination copies, and convention demonstrations. In any case, the focus of instruction in these large scale curriculum centered programs is typically on academic skills and not on underlying special needs such as training in perceptual processes.

At the next level of magnitude are the publishers who specialize in supplementary materials. Some prominent examples are Sunburst Communications, DLM, Hartley, Spinnaker, and the Learning Company. The strategy here is not to develop a complete curriculum in microcomputer format, but rather to offer a potpourri of titles that will appeal to a variety of teachers in a school district. There are also a large number of vendors of software who handle the products from different software publishers. Examples include K-12 Micromedia and Opportunities for Learning, Inc. These firms are distributors, not publishers, and apart from exercising some selectivity about the items they choose to carry, do not directly affect the introduction of new products for the marketplace.

Some developer-publisher firms are striving to fill a particular niche in the market. Koala, for example, emphasizes software that can be used with the touchpad that it manufactures. A very few are exploring the use of microcomputers and laser videodisc technology to develop interactive technology at a high level of instructional sophistication. An example is MicroEd, Inc., a midwest firm that is headed by Thorwald Esbensen, a long-time educator and advocate of individualized learning. In an unusual marketing strategy, MicroEd has made available a number of software grants to school districts in which up to $10,000 worth of its educational products are donated to set up elementary school computer-assisted libraries.

Additionally, the development of educational software is by no means restricted to well-funded publishers of instructional materials, but also includes a large number of "cottage industry" types, people who generate creative ideas and follow through in developing them. Unfortunately, they often give insufficient thought to incorporating features of design that lead to successfully marketed products.

Yet another level of software production is by teachers themselves. As indicated previously, some teachers work independently, then share their products through a coordinating center or agency which emphasizes public domain software. A prime example of the cooperative producer-user approach is SOFTSWAP, a joint project of the San Mateo (CA) County Office of Education and Computer Using Educators (CUE). Some 6,000 educators are involved and programs are available for Apple, Atari, IBM, Commodore, and other hardware.

Other teachers may be part of a network of interlocking schools and administrative units such as the Minnesota Educational Computing Consortium (MECC). There are differences in these two approaches, of course, with one of the most important being the opportunity to pilot and field test materials within a consortium of schools before they are marketed nationally. The MECC strategy has worked so successfully that it can be thought of as an "educator's publishing house" rather than a commercial publishing house. Illustrative of the MECC type of product is "Elementary Volume 7," a nine disk package for non-readers and beginning readers (see the review of this package in the Journal of Learning Disabilities, May 1984).

Lastly, one must consider the students and faculty within one's own district as an untapped resource for authoring new software. Over the long term it is likely that "friendly" authoring systems that continue to be developed will encourage teachers and students to create instructional materials just as they now prepare quizzes, handouts, and exercises for homework. (There is debate on this point, however. Some writers seem to feel that teachers' time will always be so limited that authoring will be unlikely. We take the view that as the systems become more and more user-friendly, more and more people will be tempted to participate.)

The preceding discussion of sources has been concerned with the educational software marketplace in general. The special education marketplace represents a fraction of the larger market, of course, and is concentrated in two areas. On the one hand, there are clearly a variety of programs to help in the IEP process. On the other hand, much of what is being evaluated and reviewed by teachers of the learning disabled* are really not materials targeted specifically at the learning disabled population but rather at the general school population. For example, drill and practice of academic content used for special children may not be "special" at all. There are very few developers who are preparing software uniquely designed to overcome or ameliorate specific learning disabilities.

Cartwright (1984) in his article entitled "Computer Applications in Special Education", cites a study by Chaing at the Cupertino Union School District in California. The purpose of the study was to determine the effectiveness of computer assisted instruction with mildly handicapped students. Throughout the course of the study teachers created 975 lessons

*In the Journal of Learning Disabilities, for example, the "LD Scan" reviews are contributed to by panels of teachers all reviewing the same product.

in reading, math and language arts. The teachers then tried out the materials with 200 handicapped, elementary-level children. Comparison with a control group indicated that local level teacher-initiated development of programs could be effective.

Walker and Hess (1984) suggest that software producers field test an early version of a program (under realistic field conditions), where they can observe its use and identify unforeseen problems. Similarly, Singer (1983) recommends field testing as a way to improve the quality of software. Singer suggests that both teachers and students use separate evaluation forms to provide feedback about a program.* This formative evaluation strategy is designed to compare teachers' and students' perceptions of a program's strengths and weaknesses.

Truett (1984) conducted a survey of educational software producers across the country to ascertain the extent to which they used field testing. Her findings must be viewed with caution since there was an extremely low rate of return to the questionnaire (56 returned of 406 mailed). Keeping in mind the limitations on representativeness that the low return implies (for example, the data probably overstate the real situation), it is nevertheless interesting to examine some aspects of the survey results. Of those who responded:

- 69% of the producers did some type of field testing
- 51% involved six teachers or less
- 44% used local schools
- 46% involved fewer than 50 students
- 37% relied on teacher evaluations
- 32% indicated that an in-house review took place

Truett properly acknowledged that there was a strong possibility that the non-respondents were biased toward no field testing. She has requested funding to investigate this possibility. If one assumes that the data are meaningful, then it is immediately apparent that the extent of field testing is quite small given the number of copies that may be sold and the number of

---

*See The Computing Teacher, 1983, 11(3), page 36, for the Singer Student Software Questionnaire and Singer Teacher Software checklist.

users (teachers and students) who would be affected over time. Considering this limitation, it is most important that school districts use care in the acquisition of software.

Given the diversity of sources for materials and such unevenness in their products in terms of (a) their specificity with regard to the needs of the learning disabled, (b) their "solidness" in the degree to which they are integrated with the curriculum, and (d) their merit as examples of the art of programming, it is obvious that evaluation and selection procedures should be in place in every school district that elects to serve its LD population of students via microcomputer. The next section will discuss important aspects of evaluation and selection as they are currently being practiced and should be practiced in the schools.

5. Evaluation Forms and Review Procedures

Table 1 is a comparison of evaluation forms used by different software review organizations. This table is not meant to represent an exhaustive survey, but rather the intent is to illustrate the major elements considered in present evaluation and selection practices.

The table shows that most evaluation instruments concentrate on a description of vendor/program data, an evaluation of a program's clarity and usefulness, and the instructional design features of a program. These elements are primarily concerned with the general effectiveness of a program. That is, how well the program "performs."

The "appropriate applications" comparison in the table is concerned with the identification of the underlying skills necessary for effective information processing. A program may teach these skills directly (expressed as the educational objective) or indirectly (such as the visual tracking that is involved in playing many video games). Unfortunately, appropriate applications data is almost entirely absent from most educational evaluation forms. Consequently, teachers who are looking for software that could be adapted for use with special populations can only surmise whether the software is "right" for teaching basic, underlying skills.

Table 1

COMPARISON OF EVALUATION FORMS

	AIR	National Evaluation Organizations					State and Regional Education Coordinating Agencies			Local California Education Agencies	
	CREATE	EPIE	SCHOOL MICRO-WARE	MICRO-SIFT	SOFT-WARE REPORTS	COUNCIL FOR EXC. CHILDREN*	STATE OF CALIF: TECC	STATE OF MINN: MECC	REGIONAL: SAN MATEO SHERC	HIGH SCHOOL: FREMONT	ELEMENTARY SCHOOL: CUPERTINO
I. VENDOR/PROGRAM DATA											
A. Product and Source Description	x	x	x	x	x	x	x	x	x	x	x
B. Hardware Requirements	x	x		x	x	x	x	x	x	x	x
C. Educational Objective	x	x	x	x	x	x	x	x	x	x	x
II. PROGRAM EVALUATION											
A. Clarity and Usefulness of Documentation	x	x	x	x	x		x	x	x	x	x
B. Clarity and Usefulness of Program	x	x	x	x	x	x	x	x	x	x	x
C. Externally Reviewed	x						x	x			
D. Field Testing by Producer	x	x			x		x	x			
E. Synopsis of Content	x	x	x	x	x	x	x	x	x	x	x
III. INSTRUCTIONAL DESIGN FEATURES											
A. Instructional Strategy	x	x	x	x	x	x	x	x	x	x	
B. Learner Control	x	x	x	x	x		x	x	x	x	x
C. Feedback/Reinforcement	x	x	x	x	x	x	x	x	x	x	x
IV. APPROPRIATE APPLICATIONS											
A. Vision Skills Involving Neuromuscular Functioning	x										
B. Enabling Skills Involving Perception	x										
C. Processing Skills Involving Cognition	x					x					
D. Performance Skills Involving Subject Matter	x	x			x	x					
V. OTHER		Records Management Evaluation					Description of Student Response to Program	Student Evaluation Form	Description of Student Response to Program	Checklist of Student Response to Program	

*Recommended by Council for Exceptional Children

In order to select software that can adapt to the needs and ability levels of a wide range of learners, the evaluation instrument must have some provision for addressing the key question: "How can this piece of software be used effectively in special education?"

Many commercially available programs have the potential to be used effectively with the LD population. For example, a program could be adapted for use with special learners simply by having teachers develop introductory activities to help students practice the skills necessary for mastering the software content. Clearly, special education teachers should be involved in the evaluation of "regular" software along with regular teachers so that they can identify skill areas in the software that may be used effectively with special learners.

Ideally, school evaluation and selection procedures should take into account field test results supplied by developers and narrative descriptions from experienced computer-using educators. Reviews from third party sources (e.g., TECC, EPIE) could also provide schools with a perspective that facilitates the selection process.

Below are descriptions of two commercial publications that can help in the development of an evaluation/selection process at the school district level.

The Software Catalog: Microcomputers. Published by Elsevier Science Publishing, Inc., (52 Vanderbilt Ave., New York, NY 10017), the Software Catalog is a 794 page reference catalog of educational software. The catalog contains general vendor/program data. Two unique categories are: (1) date of release (useful for keeping track of programs under development), and (2) source code availability. (The source code is available when a program has been written in a high-level language (i.e., Pascal). The source code enables users to modify parts of a program.)

The Software Catalog is published every six months and is continuously updated with new software information (updated three months after each publication.)

Hively's Choice: School Year 1983-84. This comprehensive volume has been edited and co-published by Wells Hively. The volume is available through Hively's Choice Publications, Inc., and The Continental Press, Inc., Elizabethtown, Pennsylvania 17022.

Hively's Choice is a collection of selected quality software that has been reviewed and listed by instructional objectives, instructional ideas and examples (based on field testing), curriculum connections, and preparation time. Hively's Choice proposes to be a great time-saver for teachers and parents who are looking for quality programs that perhaps meet a particular instructional objective. Subsequent volumes are planned for each year.

The following is a short description of those commercial organizations that were presented in the previous table:

School Microware Reviews, published by Dresden Associates (P.O. Box 246, Dresden, Maine 04342). School Microware Reviews is a periodical of educational software evaluations by three or more reviewer/educators. Each issue includes a narrative evaluation of at least 50 software packages--two to five programs in each major academic area. The publication is available three times a year by subscription or through a public access database called Resources in Computer Education (RICE).

EPIE (Educational Products Information Exchange). In 1982, the EPIE Institute joined with Consumer's Union of the U.S. to evaluate both hardware and software products. EPIE has provided educators with extensive training in using their evaluation form. The form uses both a numerical rating scale and descriptive sentences, and it is 12 pages in length.

Once certified by EPIE as a "Courseware Analyst," the educator provides EPIE and Consumer's Union with software evaluation reports. The reports are published in "Microcomputer Courseware PRO/FILES" and "Microcomputer Hardware PRO/FILES" and are available to schools on a subscription basis. (see following subsection "on-line databases" for the scheduled on-line implementation.) For further information write to: EPIE Institute, 1018 Keith Avenue, Berkeley, CA 94708

Software Reports. Published by Allenbach Industries, (2101 Las Palmas Drive, Carlsbad, CA 92008), Software Reports is a compilation of almost 400 educational software programs that have been evaluated by an "independent software review board" called The Evaluation Committee (TEC). The programs have been divided into 14 different subject areas. There is a separate section for special education.

Each program was evaluated by five criteria: (1) documentation, (2) ease of use, (3) program content, (4) instructional technique, and (5) educational usefulness. Programs are listed by subject category. An overall evaluation "grade" (A, B, C, D) is given to each program. Subscribers can receive future updates at a reduced cost by completing a survey questionnaire.

MicroSIFT. Housed at the Northwest Regional Educational Laboratory, in Portland, Oregon, MicroSIFT is a clearinghouse for educational software. The software evaluations are performed by the MicroSIFT Dissemination Network, SIFTnet, a group of over 200 educational agencies located across the nation.

The MicroSIFT evaluation instrument is comprised of two forms: (1) the "Courseware Description," which consists of a two-page checklist and short narratives of Vendor/Program Data, and (2) the "Courseware Evaluation" form, which includes a two-page evaluation using a 4-point rating scale (SA=strongly agree; SD=strongly disagree) and short narratives.

The software reports are the composite of the evaluations made by two teachers and a "computer education specialist." These software reviews are made available to educators through the MicroSIFT Dissemination Network and the on-line database, Resources in Computer Education (RICE).

On-line Databases

Producing objective software evaluations and disseminating them by means of print-media is costly to the publisher, and therefore to the buyer. Since most review publications do not accept advertising to maintain impartiality, they must rely on subscriptions for financial support. However, with individual subscriptions ranging from approximately $50 to $200 a year, many schools cannot afford them. A case in point was the periodical "Courseware Report Card," which recently ceased publication due to financial losses (see Educational Technology, May 1984).

The critical issue is: "How can published software evaluations be disseminated so that schools can afford to use them in their evaluation/selection procedures?". Many print-media periodicals are also offering on-line databases. These databases provide access to extensive searching capabilities for a wide range of information about educational software.

Below are descriptions of two on-line educational software databases:

RICE. A product of the Northwest Regional Educational Laboratory, the RICE database has been available through Bibliographic Retrieval Services (BRS) since 1982. RICE offers users a software database with 2400 records which can be searched by computer type, producer, grade level, subject, or application. A free text searching option allows users to locate character strings imbedded within the record, such as descriptive phrases.

EPIE On-line. Soon to be released through CompuServe is an educational software database called "EPIE on-line". This database will provide on-line access to approximately 5000 software programs listed in The Educational Software Selector (TESS). (TESS is available in print form from Teachers' College Press.) Information can be searched by type of computer, subject, grade, title, use in school (regular, remedial, special education, and administration), price, supplier, record keeping, network, review, and EPIE review. Information is delimited by using successive individual search characteristics from the above list.

In summary, in order to select software that can adapt to the needs and ability levels of a wide range of learners, the evaluation instrument must have some provision for addressing the question: "How can this piece of software be used effectively in special education?"

The key to informed evaluation and selection practices lies in the development of review standards that are both systematic and comprehensive. The next section will discuss in greater detail the elements of a comprehensive evaluation instrument and how it might be implemented in the schools.

6. Elements of a Comprehensive Evaluation and Selection Instrument

The volume of software being produced and the tremendous amount of overlap that exists between information channels about educational software makes it obvious that no teacher can stay abreast of what is currently available or know whether it is of high quality or not. Even if the largest part of the teacher's day was not already devoted to the teaching process, the task would be insurmountable. Consequently, teacher judgments must often be made on the strength of what someone else says about software that is available and worthwhile.

What "someone else" should the teacher trust? Who has the time, capacity and required knowledge to undertake the "right" kind of evaluation--evaluation that is meaningful and thorough?

For the sake of illustrating the problem and its solution, suppose three teachers agreed to pool their evaluative reviews of software. One of the first things they would have to do would be to agree on what they should look for and how they would judge it. They would, in effect, need to establish criteria that could be applied with consistency so that each person has a basis for judging the merits of an evaluation conducted by the other(s).

Assuming that our three teachers are then able to enlist the help of a fourth individual who may not be skilled in the subject matter but nevertheless is interested in education and willing to help, they would probably ask that person to take on those parts of the evaluation activity that do not demand an educational background such as the gathering of information about unit cost or the identification of the source and type of microcomputer on which the program would run.

To carry our example one step farther, let us imagine that each of the cooperating teachers has special skills in an area that the others lack or in which they aren't particularly interested. For example, one teacher might be keenly interested in perception as it relates to the act of reading. That teacher would naturally want to make direct, personal judgments about the potential of the software to help reach particular perception-training objectives. A second teacher might concentrate on evaluations of software for comprehension-building and the third on spelling. Presumably, the software identified by the three teachers could be used in different, but complementary, ways.

In essence, then, it is clear that (a) different persons can be usefully engaged in the review process, (b) if the results of their efforts are to be interpreted reliably their evaluative procedures should be uniform, and (c) the same software can be used in more than one way.

The present state-of-the-art in evaluation and selection of software is definitely <u>not</u> uniform. Moreover, it seems likely that comprehensive evaluations will be precluded unless complementary roles are adopted during the process of review. Unless important changes are made in practice, the role that chance plays (e.g., what salesman comes to call, or what conversations occur at a professional meeting) will continue to play a major role in what software a teacher selects.

We believe that a generalizable evaluation/selection form and protocol is needed that (1) distinguishes the types of information that need to be gathered for appropriate educational decisions about applications, (2) permits that information to be gathered by a team of people at different times, and (3) can be used by teachers to identify important program features. A prototype version of an AIR/CREATE Software Evaluation form is attached to this paper.

Types of information to be gathered

We envision four major types of information about software that should be collected:

1. Vendor/Program Data
2. Program Description
3. Instructional Design Features
4. Appropriate Applications

1. Vendor/Program Data

The following three categories of information do not require professional judgment or expertise but rather are based primarily on information supplied by the vendor. Accordingly, they can comprise an introductory section in the evaluation form. This information can be filled out by clerical staff, classroom aides, or volunteers in advance of critical review by faculty.

A. Regardless of whether a product is eventually selected or not, a record should be maintained of the product name, the source from which it is obtained, and cost-per-unit information.

B. Information needs to be recorded that clearly identifies the brands/models of equipment that the software can be run on as well as memory requirements or peripheral equipment needs.

C. Vendor literature often provides sufficient information to identify the instructional purpose of the material, that is, whether it is for regular instruction, for remediation, or for enrichment and general interest. The age/grade level (primary, intermediate, secondary, postsecondary) should be evident from the promotional literature as should the overall nature of the program (drill and practice, game, testing, authoring, etc.)

2. Program Description

The next sections of the evaluation can be completed by an assigned review committee within the district or across districts. These would contribute toward building a basic, descriptive file that individual teachers could turn to for initial indications of program quality.

A. A synopsis of content should be written. This should entail a brief description, together with strong points and weak points.

B. Although local evaluation is always important, it is often helpful to note the evaluations by others that might have appeared in professional journals or in publications that specialize in software reviews.

C. If the producer (developer-publisher) has made field testing data available, this should be noted. However, if the testing is questionable or claims seem exaggerated, only limited importance should be attached to it.

D. If the software has been tried within the district this should be noted as well as the extent of testing and the results obtained.

3. Instructional Design Features

Next, a section of the evaluation form should address instructional design features. These are central to the educational value of the program. Without question, they will be important to any final judgment about the appropriateness of instructional material to the particular needs of each learning disabled child. One evaluative criterion that should not be overlooked is the quality of the documentation and its completeness as a teacher guide. To the extent that the documentation gives clear directions about how the software fits into the curriculum, indicates instructional objectives, provides pre- and post-measures, and suggests supplemental activities, the teacher can see ways in which it can be integrated into the instructional program. When these elements are missing it is harder to integrate the software in a timely and effective way. (Indeed, when students discern that software is "off topic," they tend to view it as an escape from class rather than part of an overall program.)

Consider, for example, a child whose frustration tolerance is low, who is having problems in mastering a specific learning task. Is the software sequential in nature? Is provision made in the program for branching to remedial information or is the same item played back until the right guess is made (with little, if any, change in the learner's understanding of why the answer was correct)? Similarly, with the short-attention-span child, does the program allow the learner to enter and exit specific parts of the program without laboriously going back through the whole sequence or waiting until the program is complete before stopping?

A. The completeness of the educational experience involves such elements as its relevance to the curriculum, statements of purpose, worksheets and related activities suggested, accuracy and correctness of information, and records of performance.

B. User friendliness is always important, but for the handicapped student, simplicity may be crucial. Instructions must be clear and readable, it must be easy to get started, the program should not "crash," and graphics, color, and sound should be used in a way that assists the learner.

C. The program should respond to individual differences. Ideally, it should motivate, be adaptable, adjust to ability levels, and give appropriate "hints."

D. Under learner control, the program should allow selection of activities and easy access as well as the setting of appropriate rates and help options.

E. Learners should be given positive reinforcement and kept informed of their progress. Explanations should be given so that errors are not repeated.

Notes may be required on software that is nontraditional in its design properties, such as unstructured, creative programs in which input is neither right nor wrong, but simply incremental.

4. Appropriate Applications

In order to maximize the value of the evaluation-selection procedure to teachers, it is necessary to think in terms of how teachers would prefer to initiate contacts with administrators or specialists when they are seeking leads to software. When teachers use software with their special students, they should have in mind a specific goal for each learner. Teachers might well ask, "I'm interested in some material that might help with Johnny's letter reversal problem. What do you have to suggest?" This implies the need for a section of the evaluation form that takes into account the variety of instructional goals that teachers may ask about.

The goal may be improvement in basic vision skills, cognitive processing skills, perceptual enabling skills, or academic performance skills. It is important, therefore, that each software item be evaluated in terms of its utility in one or more of these goal areas.

A. A much overlooked area of instruction that can be especially important to some learning disabled children is vision skills training. Vision skills that lend themselves to training by computer are neuromuscular in nature, such as binocular fusion, accommodation, and tracking and fixation. Vision skills do not refer to lens-correctable conditions, to surgical correction, or to degenerative pathologies.

B. A type of instruction that is fundamental, but often neglected, encompasses those enabling skills that frequently cause learning disabled students to perceive their environment in a faulty or inefficient way. These include such underlying skills as being able to distinguish meaningful cues on the basis of features or patterns, establish laterality and orientation in space, recognize part-whole relationships, and attend, decode, and classify perceived stimuli. Even though much of the software that is currently in the schools may not address these learning goals as their primary purpose, there is every possibility that teachers of the learning disabled will be able to use some software creatively to achieve these secondary purposes.

C. At a higher level, software can be effective in instruction aimed at developing cognitive processes. These are the intellectual skills that are essential to memory processes, analytic processes, development of meaning and context in comprehension, and general thinking involving reasoning and problem solving. Until these skills are developed, any learning of academic subject matter is likely to be laborious and fraught with errors. Although some software exists that addresses cognitive processes directly, sometimes software in the subject matter areas can be used to build these skills.

D. Most educational software calls for performance of academic skills such as reading, writing, math, science, social studies, and vocational and business subject matter. Other programs deal with art, foreign language, social skills, and computer literacy. More precisely, programs are often aimed at some specific aspect of a subject matter; in math this might be fact acquisition, number relations, math operations, problem solving, logic, or other topics. Again, secondary uses of the software may be very worthwhile and need to be identified during the evaluation process.

Building an evaluation database

It should be clear that during the review process it may be recognized that the same software can be used in a variety of ways--the clever teacher and the creative special education department will take advantage of this opportunity to maximize the instructional return on their dollars.

For example, in the premier issue of Computers, Reading and Language Arts, Kuchinskas (1983) suggests 22 ways in which microcomputers can be used in a reading/language arts curriculum. Spanning instruction, testing, and data storage, she identifies the following applications:

1. Drill and Practice
2. Tutoring
3. Assessment
4. Record Keeping
5. Prescriptions
6. Interactive Language Programs
7. Readability
8. Language Analysis
9. CLOZE Passage Generation
10. Vocabulary List Generation
11. Test Item Generation
12. Objectives Production
13. Inventories
14. Word Processing
15. Data Banks
16. Videodiscs
17. Simulations
18. Computer Programming
19. Staff Development
20. Action Research
21. Computer Literacy
22. Other

District staff can often anticipate some of the likely questions teachers may pose about available software. Use of the evaluation form will help to identify which software does in fact deal with those kinds of instructional focus. However, there remains a problem of readily storing the information so that it can be accessed at the time the teacher needs it. If the amount of information to be stored were not so large (a variety of software that could be used for specific applications) it would be tempting to simply ask a knowledgeable district-level supervisor to recall the information. That approach is not really feasible given the great amount of software available and the numerous variations that are possible for its use.

Suppose, for example, that the teacher wants software to help improve a learner's attention span. The computer could be used as a database, allowing teachers to make inquiries (through a keyword system) so that relevant software possibilities identified with the descriptor "attention span" would be identified. These eligible pieces of software could be delimited further according to age/appropriateness, or computer-brand by simply imposing additional terms as search constraints.

By using a file management program, a district-level team could define a database using a keyword system. The evaluation instrument should serve as the foundation for selection of keywords. As each piece of software is evaluated it would be entered into the database and "flagged" by keywords (e.g., attention; discrimination; problem solving) as checked on the evaluation form.

The advantage of implementing a local level database is that a teacher could efficiently locate software that: (1) meets the search criteria the teacher establishes, (2) is adequate according to a comprehensive evaluation procedure, and (3) is currently available within the district.

Summary

During the initial evaluation and review of any particular piece of software, it is unlikely that any one person will be able to make the "final" judgments about all of the educational applications that are feasible. One strategy would be to use a committee approach to the review of the material so that its possible value can be evaluated separately for (a) vision skill training, (b) perceptual enabling skills training, (c) cognitive processing skills training, and (d) academic performance skills training. Whether the applications aspect of the software evaluation is carried out by an individual or a committee, however, the intent should be to identify a variety of ways in which the particular software item can be used, not just the purpose as stated by the vendor. The goal should be to think of the material as a resource that, if added to the district's collection, should be used to the fullest to assure the best dollar return on investment.

By developing and using a form that provides a checklist approach to the listing of appropriate applications, information about the software can be readily encoded into a database. The database could then be accessed by teachers and specialists in answer to the question, "What do you have that will help me teach "---?".

Staff in CREATE have developed a preliminary version of an evaluation and selection form for use in the schools (it is attached). It will be field tested in the 1984-85 school year in cooperating schools around the country.

REFERENCES

Cartwright, P.G. Computer applications in special education. In D. F. Walker and R. P. Hess (Eds.), Instructional software: Principles and perspectives for design and use. Belmont, CA.: Wadsworth, 1984.

Chambers, J. A., & Sprecher, J. W. Computer-assisted instruction: Its use in the classroom. Englewood Cliffs, N.J.: Prentice-Hall, 1983.

Hagen, D. Microcomputer resource book for special education. Reston, VA: 1984.

Kuchinskas, G. 22 ways to use a microcomputer in reading & language arts classes. Computers, Reading and Language Arts, 1983, 1(1), 11-16.

Loebl, D., & Kantrov, I. Micros in the special ed. classroom. Electronic Learning, 1984, 3(5), 38-39.

Senf, G. M. Learning disabilities challenge courseware. The computing teacher, 1983, 10(6), 18-19.

Singer, B. How do teacher and student evaluations of CAI software compare? The Computing Teacher, 1983, 11(3), 34-36.

Torgeson, J. K. The role of nonspecific factors in the task performance of learning disabled children: A theoretical assessment. Journal of Learning Disabilities, 1977, 10(1), 27-34.

Truett, C. Field testing educational software: Are publishers making the effort? Educational Technology, 1984, 24(5), 7-12.

Turkel, S. B., & Podell, D. M. Computer assisted learning for mildly handicapped students. Teaching exceptional children, 1984, 16(4), 258-262.

Walker, D. F., & Hess, R. D. Evaluation in courseware development. In D. F. Walker and R. D. Hess (Eds.), Instructional software: Principles and perspectives for design and use. Belmont, CA.: Wadsworth, 1984.

EDUCATIONAL SOFTWARE EVALUATION FORM

AIR/CREATE

I. VENDOR/PROGRAM DATA

A. PRODUCT AND SOURCE DESCRIPTION

1. Program Name ________________________________
2. Distributor Name ________________________________
3. Address ________________________________
____
4. Telephone (   ) ____________________
5. Single Program ___ Part of a Series ___
6. Cost $________
7. Terms for updated versions ____________________
8. Back-up Policy
   Includes back-up copy______
   Terms for multiple copies____________________
   Hard disk version available ______

B. HARDWARE REQUIREMENTS

1. Microcomputer (Enter model)
   Apple______ Atari______ Commodore______ IBM______ Other____________________
2. Memory Required __________k
3. Storage Medium
   Diskette___ Tape cassette___ ROM cartridge___
4. Equipment (Enter R for required, O for optional)
   Printer______
   Disk drive ______1
   Disk drive ______2
   Color monitor______
   Graphics tablet ______
   Joystick______
   Mouse______
   Voice recognition device______
   Voice synthesizer______
   Other ____________________
5. Optional Devices
   Joystick____ Mouse____ Voice synthesizer____
   Graphics tablet ____ Other__________________

C. EDUCATIONAL OBJECTIVE

1. Instructional Purpose
   Remediation___ Regular Instruction___ Enrichment___
2. Suggested Grade/Ability Level
   Primary___ Intermediate___ Secondary___ Postsecondary___
3. Presentation Mode
   Drill and practice___
   Educational game___
   Problem solving___
   Simulation___
   Tutorial___
   Information retrieval____
   Testing____
   Authoring system____
   Classroom management____
   Game (for fun only)____

II. PROGRAM DESCRIPTION

A. SYNOPSIS OF CONTENT

1. Description______________________________

2. Strong points______________________________

3. Weak points______________________________

B. EXTERNALLY REVIEWED

1. Source______________
2. Rating______________
3. Cautions______________________________

C. FIELD TESTED BY PRODUCER

1. Results are available_____
2. Number of students involved_____
3. Claims made______________________________

D. LOCAL TRYOUT

1. Who/where tried______________________________
2. Number of students involved_____
3. Results______________________________

III. INSTRUCTIONAL DESIGN FEATURES

	YES	NO	NA
A. EDUCATIONAL COMPLETENESS			
1. Integrated with curriculum	___	___	___
2. Names prerequisite skills	___	___	___
3. States instructional outcomes clearly	___	___	___
4. Suggests related activities			
Before using computer	___	___	___
After using computer	___	___	___
5. Information presented is accurate/correct	___	___	___
6. Includes worksheets	___	___	___
7. Provides for testing			
Pretest	___	___	___
Posttest	___	___	___
8. Provides performance records	___	___	___
Based on sound learning theory	___	___	___
B. USER FRIENDLINESS			
1. Operating manual is clear	___	___	___
2. On-screen instructions are easy	___	___	___
3. On-screen text is readable	___	___	___
4. Menu driven (learner can begin easily)	___	___	___
5. Safeguarded against "crashes"	___	___	___
6. Graphics and color assist rather than distract	___	___	___
7. Sound can be turned off/on	___	___	___
C. LEARNER SENSITIVITY			
1. Learner motivation is sustained	___	___	___
2. Program can be adapted to individual needs	___	___	___
3. Automatically adjusts to student's ability level	___	___	___
4. Gives clues, hints to encourage correct answer	___	___	___

D. LEARNER CONTROL

	YES	NO	NA
1. Can control rate of presentation	___	___	___
2. Can select activity (without repeating completed segments)	___	___	___
3. Can repeat any activity	___	___	___
4. Can exit at any time	___	___	___
5. Can get help on-line at any time	___	___	___

E. FEEDBACK/REINFORCEMENT

	YES	NO	NA
1. Positive reinforcement is immediate	___	___	___
2. Incorrect answers are explained	___	___	___
3. Remedial loops are built in	___	___	___
4. Students are rewarded only for correct responses	___	___	___
5. Student is informed of success rate	___	___	___

Notes ____

IV. APPROPRIATE APPLICATIONS

A. VISION SKILLS INVOLVING NEUROMUSCULAR FUNCTIONING

1. Binocular fixation (coordination of the two eyes)_____
2. Accommodation (focusing a clear image) _____
3. Tracking (appropriate eye movements and fixation) _____

B. ENABLING SKILLS INVOLVING PERCEPTION

1. Discrimination
   a. of features___
   b. of patterns___
2. Attention
   a. span ___
   b. concentration___
   c. alertness___
3. Classification
   a. grouping___
   b. figure/ground___
4. Cue Identification
   a. searching___
   b. differentiation___
5. Coordination and Integration
   a. visual/motor___
   b. auditory/motor___
   c. visual/auditory___
6. Part-whole Relationships
   a. verbal___
   b. graphic___
7. Orientation
   a. lateral___
   b. spatial___
8. Decoding of Letter Shapes
   a. typed or printed___
   b. written___
   c. reversals/inversions___
9. Decoding of Words
   a. letter sets___
   b. word length___
   c. word shapes(ascenders/descenders)___

C. PROCESSING SKILLS INVOLVING COGNITION

1. Analytic Techniques
   a. In reading (e.g., key ideas, sentence structure) ___
   b. In math (e.g., conservation, relative magnitude) ___
   c. In science (e.g., cause/effect, hypothesis checking)___
   d. In social studies (e.g., timelines, political interactions)___

2. Understanding and Meaning
   a. vocabulary and definitions ___
   b. use of context___
   c. comprehension ___

3. Recall of Information
   a. chunking and grouping___
   b. mnemonic techniques___
   c. rehearsal ___

4. General Purpose Thinking
   a. reasoning___
   b. problem solving___
   c. study strategies___

D. PERFORMANCE SKILLS INVOLVING SUBJECT MATTER

1. Reading
   a. speed___
   b. whole word recognition___
   c. phonics and blends___
   d. comprehension___
   e. other____

2. Writing and Composition
   a. spelling___
   b. punctuation___
   c. capitalization___
   d. word usage___
   e. parts of speech___
   f. organization of ideas___
   g. syntax___
   h. proofreading___
   i. other____

3. Math
   a. fact acquisition___
   b. number relations___
   c. math operations___
   d. problem solving___
   e. logic___
   f. graphing___
   g. other____

4. Science
   a. fact acquisition___
   b. observing and measuring___
   c. data interpretation ___
   d. formulating generalizations and hypotheses ___
   e. problem solving___
   f. other____

5. Social Studies
   a. fact acquisition ___
   b. data interpretation ___
   c. identification of concepts and issues ___
   d. evaluation of evidence ___
   e. formulating generalizations and hypotheses ___
   f. other ____

6. Vocational/Business
   a. technical skills___
   b. work readiness___
   c. career awareness___
   d. other ____

7. Foreign Languages
   a. vocabulary___
   b. rules and patterns___
   c. comprehension___
   d. other ____

8. Arts
   a. perception of visual properties___
   b. analysis and interpretation___
   c. technique and eye-hand coordination___
   d. creativity of design and form___
   e. other____

9. Social Skills
   a. handling stress and conflict___
   b. taking responsibility___
   c. communicating effectively__
   d. analyzing feelings___
   e. group participation___
   f. other____________________

10. Computer Literacy
   a. general computer use___
   b. programming___
   c. graphics___
   d. other____

11. Other Subject Matter
   a. ____ _____