The Educational Effectiveness of Mathematics Computer Software for the Blind and Partially Sighted

Lynn Manconi-Menendez

A Thesis

in

The Department

of

Education

Presented in Partial Fulfillment of the Requirements
For The Degree of Masters of Arts at
Concordia University
Montreal, Quebec, Canada

June 1989

ISBN 0-315-51371-3

ABSTRACT

The Educational Effectiveness Of Mathematics
Computer Software for the Blind And Partially Sighted

Lynn Manconi-Menendez

The purpose of this study was to conduct a series of summative evaluations of mathematics computer software designed for the blind. The evaluations are based on explicitly stated criteria of educational values and field-testing. At present, most mathematics software developed for the blind is limited in quality and educational effectiveness. The major problem is that the software is not being field-tested before its distribution. The study begins with a review of 33 mathematics software programs designed for the blind based on a checklist incorporating the blind user's special needs. A grid analysis was conducted in order to rank the software according to three different categories: good, fair and poor. Three programs were chosen, one from each category, and tested in a field study to assess their suitability and educational effectiveness with six visually handicapped subjects. The study suggests that most mathematics software is neither effective nor appropriate for the blind or partially sighted audience. Guidelines are offered for instructional designers and manufacturers to follow in remedying this situation.

ACKNOWLEDGMENTS

I wish to thank my family, friends and colleagues for their continual support and encouragement throughout the various stages that have led to the completion of this thesis. I would like to acknowledge the following people for their valuable assistance.

For his entrusted confidence, understanding and expertise, I wish to thank Mr. J.K. Evans, principal of the Philip E. Layton School.

To the teachers of Philip E. Layton School, I would like to thank them for their sound advice and unconditional support.

For imparting his knowledge, guidance and time I am grateful to my advisor, Dr. Jon Baggaley.

I would also like to thank Dr. Gary Boyd and Dr. Gary Coldevin for their interest and helpful guidelines.

Lastly, I would like to thank the visually impaired students for their participation in this study.

DEDICATION

To my husband, Victor Menendez

and my son, Victor-Martin Menendez

The Educational Effectiveness of Mathematics Computer Software For The Blind And Partially Sighted

Table of Contents

Chapter 3: Instructional Design Adequacy

Chapter 4: Technical Adequacy

List of Tables

CHAPTER 1

Evaluation Approaches

PURPOSE OF THE THESIS

This thesis aims to apply summative evaluation methods in the testing of learners' responses to educational software. Summative evaluation is utilized after an innovation has been developed and its procedures defined. Data are then gathered to assess the innovation's effects on the target population. The data serve as input for decision making by the developer (Scriven, 1967).

Katz and Morgan (1974) suggest that it is quite feasible that many materials which have failed summatively might have succeeded if information from formative evaluation had been available during program development. Formative evaluation is a process by which data are collected concerning the effectiveness of a product during its development. The purpose of such an approach is to guide revisions which will improve the product prior to its final production (Patterson & Bloch, 1987). According to Dick (1980) there is a lack of research on formative evaluation, and even less research has been conducted on the formative evaluation of computer-assisted instruction (CAI). Baker and Alkin (1973) suggest that publishers who employ formative evaluation in the development of their products have found that this approach is an effective way to detect and correct weaknesses in the materials. Baker and Alkin's findings suggest that the use of formative evaluation by software developers might increase the quality of software being produced.

Baggaley (1987) has stressed that summative evaluation data may be used within a research design, for formative purposes. This perspective is consistent with the views of Edwards (1974) who distinguishes between research and evaluation on the basis of their objectives.

> Research and evaluation are different both in terms of their objectives and their relation to change. Research may be conducted for the purpose of testing theory, expanding concepts, or explaining behavior. Evaluation on the other hand, is carried out to provide information for decision making. While research is aimed at increasing knowledge, evaluation is aimed at achieving a practical goal with an emphasis on utility. Thus, while evaluation may employ research knowledge and research methodology, it is not research. (Edwards, 1974, p.378).

A summative evaluation, according to Russel and Blake (1988), is able to "determine the effectiveness, efficiency, value or worth of the materials" (p.25). In order to achieve these results, several stages are necessary.

The subjects should be pretested to determine their entry skills. Also, the instructional materials which will be reviewed should be intact and used as they would be used under actual instructional conditions. A post-test should be administered after the instruction has taken place. Attitudinal measures and questionnaires may be used to verify the subjects' affective behaviors (Russel & Blake, 1988).

In analysis, the difference between the subjects' pretest and postest scores may be calculated in order to determine whether any learning or attitude change has taken place. The assumption is that if learning did not occur for the majority of the subjects, then it is the fault of the materials (Bloom, Hastings, and Maudaus, 1971).

As Carey (1980) points out, "Summative comparisons among competing programs require careful experimental controls to ensure program fairness and the validity of conclusions." (p.18). In the attempt to ensure predictive validity the materials must be presented in their final form and the subjects should be representative of the intended target population. To avoid experimenter bias, the individual who will be conducting the summative evaluations should not be one of the developers of the software (Russel & Blake, 1988).

As suggested by Fitz-Gibbon and Morris (1975), formative research can address the greatest limitation in formative evaluation: the dilemma of what to do after a problem has been detected in instruction. Often, designers do not know what changes are necessary when given advice from evaluators to 'revise appropriately'. In most cases, the manufacturers have used their experts' best knowledge to produce the product, and it may not be apparent to them what needs to be revised. A theory-based approach could direct the data collection and interpretation of results, and make it more feasible for the software developers to determine the aspects of the software program which have failed in terms of instruction efficiency.

In today's market, software is being produced at an astonishing rate, but many studies reveal increasing dissatisfaction with its educational quality (Lewis, Harrison, Lynch & Saba, 1988; Naiman, 1987; Owston, 1987; and Jay, 1983). Naiman (1987) goes so far as to say that, "Good software is expensive to make, is rarely produced on schedule, is never completely debugged before it goes to market, and represents an investment risk" (p.198).

Thus, low quality may be attributed to economic reasons. Naiman suggests that most textbook publishers are reluctant to publish software. They have developed only minimal amounts of courseware so that they may continue to profit from selling textbooks.

Ploch (1986) arrives at a similar conclusion, stressing that publishers feel insecure about the future of the software market and hesitate to invest in it, especially in view of the widespread problem of software piracy.

The development and production of software for the handicapped is notoriously slow and expensive. Often to stay in business, software producers must concentrate on marketing software which will sell widely (Sturdivant, 1984). However, little quality software for the disabled is being produced.

> Much of the computer software is designed as it is marketed. Many manufacturers of aids for the disabled are among the worst offenders in this respect. For some companies, the difference between research and development, and production and marketing is apparently obliterated. (Laurer & Mowinski, 1986, p.911).

Providing access to standard software programs for individuals who cannot 'see' the monitor is particularly difficult. In many cases, the more powerful standard software takes complete control of the computer when it is loaded. Therefore, with limited DOS, the program does not support attachment of braille, synthetic speech or other peripheral devices intended to provide access to the visually impaired. In addition, the standard programs themselves are often 'locked' and the source code is unavailable, making any direct modification of the programs impossible (Young, 1984).

Conventional programs are designed for individuals who can see the screen. The screen presents information to the user in a 'parallel format' which implies that the information on the entire page is presented to the user all at once. Blind users who are utilizing a voice synthesizer to 'see' the screen, hear only a single word or a character at a time (Goodrich, 1984). For sighted users this would be like looking at the screen through a straw, and struggling to read and make sense of the words. Eventually, the sighted users would be able to decipher the messages on the screen. However, the organization and presentation of the information would be far from optimal.

Needless to say, programs written specifically for use by the blind should follow different strategies for organizing and presenting material to its users (Schreier, DeWitt, Goldberg & Leventhal, 1987). The present thesis utilizes a research approach to summative evaluation in order to assess if available software for the blind is appropriately designed for this group's instructional needs. It attempts to generate guidelines for future software developers, and to qualify as 'formative research' consistent with the definition of Baggaley (1987). The study excludes programs which permit access to the blind user as a secondary consideration. Rather, the author is interested in finding out if developers are using appropriate strategies in programs which they claim to have designed for the blind specifically.

THE ISSUE OF FIELD-TESTING

The design of field-testing procedures is crucial to the success of software evaluation. Field-testing methods can determine the user response and reaction, and whether the product achieves any student learning or significant change in behavior (Truett,1984).

As the literature indicates (Owston, 1987; Patterson & Bloch, 1987; Steffin, 1983) most reviews on CAI are summative evaluations with little or no prior-testing . According to Dick (1980), "at least one revision of instructional materials results in significantly improved learning over no revision at all... revisions based on knowledge of student data are better than revisions based on the subject matter expertise of the content writers" (p.5).

Dick (1977) proposes three stages to the formative evaluation of instructional materials: one-to-one evaluations (with one to three learners); small group evaluations (five to fifteen learners); and field-tests (twenty or more learners). Golas (1983) sees the following main advantages of field-testing a large number of learners: defining managerial problems and the need for additional resources. However, Golas goes on to point out that conducting a full-scale test of CAI is probably not a necessity, for the cost and amount of time needed by such an evaluation can outweigh the value of the information collected.

Weston (1986) is critical of Golas' approach and questions the validity of revisions based on data from just a few students. Several authors (Jay, 1983; Truett & Ho, 1986; Dick, 1980) agree with Weston that a sample of twenty or more students is essential for field-testing of computer software.

Golas (1983) points out that conditions exist in CAI testing that would not be present in the testing of print materials. The greatest distinction between evaluation of print materials and CAI revolves around the hardware aspects of the computer. This includes both the input device (keyboard, mouse, touch panel, etc.), and the output device (voice synthesizer, monitor, printer, etc.). With printed materials there is no hardware aspect except for "hands-on" training where materials are used conjointly with a specific type of hardware.

Briggs and Wager (1981) note that hardware and software malfunctions can result in a postponement of the evaluation trial, and can lead the learners to become frustrated and bored if it becomes impossible for them to recapture their place when operation resumes.

Another problem arises from the learners having very little or no computer experience. The target audience for whom the instruction is being developed may not be familiar with the medium or may experience difficulty in understanding the text, e.g., via the voice synthesizer.

According to Goodrich, Bennett, Paul & Asley (1980), most voice synthesizers require that the user spend about one to two hours practice to understand the speech.

Golas (1983) suggests a formative evaluation approach which is similar to the three stages proposed by Dick (1977), yet geared specifically to encounter the difficulties found in CAI. Golas' proposal begins with one-to-one evaluations with one student and handwritten frames. Once the revisions from the trial has been made, the actual programming is implemented. This method reduces the excessive costs of reprogramming. Golas then recommends that the program be evaluated with one student at the computer terminal. Afterwards, a small-group evaluation with three students at the computer terminal is conducted, since hardware constraints preclude a larger number at any one time.

In Truett's (1984) survey of 56 educational software producers and publishers, 31 companies reported "field-testing" all of their software. Truett's findings revealed that most of the field-testing that does take place is generally teacher evaluation, rather than based upon an assessment of student performance. Truett also indicates that most commercial publishers do not employ a formative evaluation approach, nor conduct extensive summative field studies. The CAI software business is still in its infancy, and the cost of such studies is high. Truett adds that,

"...many producers are not even convinced that such efforts enhance marketability of their software" (1984, p.12).

Truett's study was replicated in 1986 by Truett and Ho, revealing once again that many producers place a low value on field-testing their products. Out of 125 software producers and publishers, only 83 conducted field-tests. Truett and Ho's findings also showed that fewer than six teachers were involved in these field-tests, that they involved fewer than fifty children from a narrow range of local schools.

Truett and Ho recommend that pre and posttest scores of student achievement should be included in educational software documentation to assess whether a change in student achievement has occurred as a result of using a particular program. Inevitably, "student achievement is still the bottom line in educational evaluation and accountability" (Truett & Ho, 1986, p.25). Throughout the literature on software evaluation, the most popular approach is to use a summative evaluation based upon teachers' reviews rather than on results obtained from field-testing students working with the software (Lewis, Harrison, Lynch & Saba, 1988; Owston & Wideman, 1987; Allan, 1984).

The National Council of Teachers of Mathematics (1985) recommends that only two or three students be observed using a program, with the emphasis of the results being based upon the teacher's evaluation of the software program.

Owston and Wideman (1987) provide an alternative viewpoint suggesting that, "Educators may have the expertise to assess the quality of a program's content and instructional design, but it seems unreasonable to assume that they will be able to judge with any precision how students will respond to it." (p.296).

Several authors' (Signer, 1983; Preece and Jones, 1985) findings support Owston's and Wideman regarding the credibility of teachers' opinions concerning software evaluation. Signer (1983) found that students were more critical of a software's pacing, instructional and motivational value than their teachers were. Preece and Jones (1985) found that even teachers knowledgeable in the field of computers were often uncritical of the software they evaluated.

Weston (1987) suggests that a combination of experts and learners be involved in the evaluation of software due to the different types of feedback each can provide. But what defines an expert? Steinberg (1983) insists that persons who have taught the target population are the ones best qualified to judge the suitability of courseware for that group. Steinberg goes on to say that content experts are not necessarily familiar with the target population's needs, and that an instructor who is conscious of the specific attributes of his/her handicapped students may be in a better position than 'subject experts' to justify the relative importance of the software's instructional characteristics (p.18).

Baker and Alkin (1973) suggest that revisions based on knowledge of student data are better than revisions based on the subject matter expertise of the content writers. A study conducted by Hannaford and Taber (1982) on computers and the handicapped, emphasized that many handicapped youngsters tend to have a low self-esteem. Hannaford and Taber point out that a teacher of the handicapped may be more aware of the use of sarcasm or abrasive feedback in software (e.g. "No! Wrong again, silly!"). This feedback may be counterproductive for handicapped students who may be deeply offended.

A DEVELOPMENTAL TEAM APPROACH

Much of the software designed for the handicapped is not educationally effective due to the programmers' lack of knowledge about educational principles and the specific needs of the handicapped (Taber, 1983). Brandon (1988) acknowledges the difficulties which hinders the production of high quality instructional software. He adds that software development requires many skills and designers are expected to be good programmers and instructional designers as well as experts in the subject matter.

Several authors have indicated, (Weston, 1986; Smith & Boyce, 1984; Walker & Hess, 1984) that a developmental team consisting of a subject matter expert, an instructional designer and a programmer can be valuable in the development of courseware. This team approach could provide the necessary expertise and avoid an over-emphasis on one person's errors or biases. The subject matter expert is able to contribute knowledge of the content and also to offer knowledge of the learners' attributes (Weston, 1986). An instructional designer can guide the development by providing a systematic plan for the instruction by applying principles of learning throughout the process (Romiszowski, 1981). Finally, the programmer's role would be to describe the capabilities of the computer; and to implement the subject matter expert's and instructional designer's ideas into the program (Smith & Boyce, 1984).

When software is developed by an individual rather than a team, the individual may not have an adequate background to contend with all the variables involved in courseware design. A team approach appears to be more advantageous when a systematic planned process is utilized. It is likely that this approach can produce far more effective and efficient instruction (Gagne & Briggs, 1977).

EVALUATION CRITERIA

The creative possibilities of the computer are endless. Unfortunately, much of the software currently on the market appears trivial in terms of this potential (Brandon, 1988). Technical excellence is widespread today, and therefore any software which seems anything less than excellent must be rejected as unacceptable. Efforts are being made in special education to capitalize upon recent hardware and software advances, and to try to build quality software based upon sound design principles.

In the subsequent chapters, a series of important design principles will be described which are regarded as crucial in effective educational software. Three distinct areas exist in the field of courseware evaluation: software for the blind, instructional design adequacy and technical adequacy. A chapter will be dedicated to each of these areas.

CHAPTER 2

Software For The Blind

In this chapter factors pertinent to the visually handicapped will be discussed. This will include the target population, the rationale of the medium and learners' attributes. Affective criteria will also be considered. These will include sources of self-esteem for the visually handicapped, feedback and motivation.

TARGET POPULATION

The group of people roughly categorized as 'handicapped' is at least as heterogeneous as a nonhandicapped group. The divergent learning characteristics of handicapped students require different procedures depending upon the severity and type of nandicap (Cartwright, 1984).

Until the 1980's little literature was available on the potential of computer instruction with visually impaired children (Ashcroft & Bourgeois, 1980). The underlying assumption of available research is that there appears to be a high degree of motivation on the part of blind and partially sighted students to use computers (Ruconich, 1984; Vanderheiden, 1982; Lewis, Harrison, Lynch & Saba, 1988; and Laurer & Mowinski, 1986). However, a recent surge of interest in the development of electronic aids for the visually impaired has led to increasing research (Sanford, 1984).

The software programs which will be evaluated in the current study were designed specifically for blind and partially sighted anglophones with normal intelligence who are in an upper elementary level, whose only access to the software is via adaptive hardware like a voice synthesizer.

RATIONALE FOR MEDIUM SELECTION

The same type of educational reasearch is currently being addressed to CAI as to the earlier educational media. Substantial differences are apparent between media. For example:

> (Television) is mostly a one-way medium of communication, basically designed for entertainment and converted (or subverted) to education;...[the computer] is a multipurpose, semi-intelligent, interactive tool" (Salomon & Gardner, 1986, p.13).

Yet, despite the differences between technologies, questions regarding instructional effectiveness, optimal design, cost, users' attitudes and techniques are valid for each. When evaluating software, instructional characteristics are often the most important attributes to verify according to Lathrop and others (Lathrop & Goodson, 1983; Steinberg, 1983; Bitter & Wighton,1987; Owston, 1987).

Other authors assert (Coopersmith, 1967; Maslow, 1970; Rogers, 1951; Kirtley, 1975; Tuttle,1987) the affective components of software appear to be equally important, especially for handicapped students. Does one medium teach better than another? It is naive to assume that any one medium can affect learning in isolation, and more profitable to assume that learning occurs because of some special attribute or quality which is associated with each specific medium (Salomon & Gardner, 1986).

Is the computer a suitable medium for all learning outcomes? For specific outcomes such as knowledge and comprehension the computer appears feasible as a medium for instruction. In situations where the computer is required to make a judgement about the learner's input, Smith and Boyce (1984) recommend,

> ...the computer would not be the appropriate instructional medium as its judging capabilities would be inadequate,... [Evaluation] would be better done by a live teacher. Other learning outcomes including motor skills acquisition and attitude formation would also be unlikely candidates for CAI delivery (Smith & Boyce, 1984, p.5-6).

Romiszowski (1981) suggests a multi-level design process for selecting an appropriate medium for instruction. His analysis for the starting point of the media selection process lists the following main objectives which should be easily implemented by the chosen medium.

> The media that we use for instruction must be capable of transmitting all the information, supplying all the instructional stimuli which the lesson content requires, and, second, the media should also help the learner to engage in the appropriate learning activities (Romiszowski, 1981, p.340).

Since the presentation and judgement capabilities of the computer are limited, a selection decision should precede the detailed specification of objectives and their related test items (Smith & Boyce, 1984). CAI, when properly applied, offers an invaluable addition to traditional learning tools, both for handicapped and nonhandicapped students. CAI is able to provide non-threatening, individualized instruction which allows students to proceed at their own rate of speed through sequential steps. It also has the potential to extend the teacher's expertise in defined subject areas (The Trace Research and Development Center For The Severely Communicatively Handicapped, 1982).

Rather than assume that the computer can control or direct the learners' performance, research suggests that individuals are likely to influence how the technology will affect them (Salomon & Gardner, 1986). Walker and Hess (1984) encourage evaluators to go beyond the assessment of direct student outcomes and to review some of the more subtle impacts which may influence students' behavior when using a computer. Walker and Hess propose the following questions to include in an evaluation of learners' attitudes towards computer software.

> Does the program make the learners want to use computers more or avoid them? Do users leave the program feeling confident of their ability to use computers or are they intimidated by them? Is their self-esteem enhanced or degraded? (Walker & Hess, 1984, p.213)

Such questions may prove difficult to answer because the effects sought are not directly observable. Also, the effects may be produced by a combination of other factors apart from the program being reviewed. Walker and Hess urge the inclusion of learners' attitudes towards courseware as an essential criterion for a thorough evaluation.

LEARNERS' ATTRIBUTES

When evaluating courseware, the evaluator should obtain some information regarding the target audience's computer sophistication, e.g. familiarity with the keyboard as well as their level of apprehension about computers (Smith & Boyce, 1984). Leiker (1982) stresses the importance of acknowledging the diversity of user characteristics and learning in CAI.

> All users are subject to boredom, frustration, apprehension, forgetting and errors. Learning is also an important consideration, any given system action might produce various effects depending on the experience and training of the user. For example, the same response time might produce apprehension or boredom, depending on the user and his level of experience. (Leiker, 1982, p.1)

Schneidermann (1980) describes two aspects of user characteristics which are important to consider when evaluating CAI. A user's attitude toward a computer can affect his or her performance and learning (Schneiderman, 1978). Experimental research shows that users who felt at ease and displayed confidence when using a system, committed fewer errors and took less time than the users who did not feel confident (Dickson, Senn & Chervany, 1977).

Anxiety regarding CAI is another factor which may affect a user's performance. Schneidermann (1980) discusses a number of such fears. These include apprehensiveness regarding response times which are either too long or too short, the fear of losing a file or damaging the hardware, and the fear of being supervised and making errors. The extent to which these tensions are manifested depends on the individuality of the user and the amount of computer experience. Such fears can reduce the users' short-term memory capacity (Schneidermann, 1979).

Krolick (1984) suggests that the anxiety of using a computer can be reduced for persons with a visual handicap if they are given the opportunity to feel the computer and discover as much as possible about the machine beforehand.

Salo...on and Gardner (1986) recommend that researchers should focus on how the unique qualities of instructional software might be adapted to various learning styles. Currently, for blind users the software which is available restricts individual learning styles. Much of the adapted software organizes and presents information as unadapted software does (Goodrich, 1984). It is not likely there will be much improvement in this respect (Vanderhelden, 1982).

SOURCES OF SELF-ESTEEM FOR THE VISUALLY HANDICAPPED

In order to fully understand a visually handicapped person's sources of self-esteem, the first step is to examine the impact a physical impairment can have on the self. One must first understand that visually impaired persons are not necessarily abnormal, maladaptive or in need of long-term therapeutic counselling. Visually handicapped persons are usually normal people who are responding to life's demands with the added stress of a physical impairment (Tuttle, 1987).

Maslow (1970) advocates that people strive for personal adequacy, a universal basic need, by engaging in self-enhancement activities. A person's ability to cope successfully with life's demands depends on his/her personal adequacy. Personal worth is not optional; it is mandatory for the emerging self-concept. Coopersmith (1967) defines self-esteem as the affective component of self-concept. A person's self-esteem can influence how the individual will view his/her social and physical environment. "A healthy self-esteem results from valueing the self and being valued by others" (Coopersmith, 1967, p.54).

Rogers (1951) agrees with this theory and includes self-acceptance as an integral part of self-esteem. A person's attitudes towards him or herself determines his/her attitudes towards others. An individual must first experience a measure of self-love and self-acceptance before he or she is able to love and accept others. Acceptance of others depends upon a person's self-acceptance and his/her ability to interrelate with others. Therefore, self-esteem, self-acceptance and acceptance by others are all inextricably intertwined (Rogers, 1951).

Both Coopersmith (1967) and Rogers (1951) contend that self-esteem is influenced by two factors: reflection from others and judgements of one's own competence. "Handicapped individuals experience a wider range of predominantly negative reflections, which results in lowered self-esteem"

(Kirtley, 1975, p.42). Tuttle (1987) emphasizes that the blind students' self-concepts are particularly influenced by significant others' expectations, judgements, standards and values. These expectations are rarely congruent with the visually handicapped students' aspirations for themselves.

In the light of this discussion on self-esteem, four major problems arise which blind persons must face when using computers. First, in order to feel competent and adequate, blind persons must develop good coping skills and adaptive behaviors. For instance, blind students need to be able to understand the voice synthesizer since it is the medium used for accessing information on the screen (Goodrich, Bennett, Paul & Wiley, 1980).

Secondly, the students must deal with the task of maintaining a sense of high self-esteem in the face of predominantly negative reflections. Due to the rapid development in the field of microcomputer technology, visually handicapped individuals' efforts to acquire and have access to computer programs often prove to be futile. The programs which they have learned are frequently outdated (Young,1984). Status is often based on knowing the 'latest thing' which of course, has not yet been adapted for blind persons' needs.

The third problem rests on blind people's ability to maintain control over situations. Visually impaired computer users must know much more about computers and computer aids than most of their sighted counterparts since they have fewer resources on which to depend (Goodrich, 1984). The fourth problem has to do with their dependence on others who can 'see' the screen. (Laurer & Mowinski, 1986, p.911).

FEEDBACK AND MOTIVATION

The Council of Ministers (1985) describes effective feedback as follows: "Feedback should be motivational and sensitive to the users' needs, negative feedback should not be more attractive than positive, and the computer should make use of its capabilities by providing corrective feedback which shows the steps in the solution or re-explains the concept" (1985, p.13).

A critical quality of a software program's motivational aspect is the attractiveness or rewardingness of the content (Williams & Williams, 1985). Ideally, this should prevail even when the content becomes difficult. The maturity level of the students should also be considered, for as students become more directly interested in the content of the material, they require less encouragement. Many software designers seem to favor feedback statements which to some users may be inappropriate, even condescending. For instance, hearing "You're so smart, I want to marry you! " everytime an older blind user solves a step in a math problem may become distracting rather than rewarding. Williams and Williams (1985) suggest "the type and degree of reinforcement must be selective to the grade level" (p.117).

Jaegar (1988) and Dempsey and Wager (1988) argue that careful planning of feedback types and timing in software programs needs to be of primary interest to developers. Perhaps students only want to know if their answer is correct, and that is all which is required to reinforce them. Jaeger recommends that both publishers and purchasers of software avoid the flashy glitter presented in screen displays and begin focusing on the "embedded learning theory, instructional methodology, subject scope and content" (Jaegar, 1988,p.22).

As will be seen, ineffective feedback appears in most of the software reviewed in the present study. In one example, whenever the correct answer is input, the program repeats the question with the answer. However, if the user inputs an incorrect answer, it does <u>not</u> repeat the question! A user may feel motivated to get the answer wrong so that the question will not be repeated. The author has also found other examples in which feedback is misused. Many derogatory comments were found (e.g. "I know you can do better!"); unduly time consuming musical tunes; loud comments such as "NO! NO! NO!"; inappropriate responses such as "You're A Wizz!" after four attempts; and worst of all, graphic displays of which the blind user is totally ignorant.

Another problem which arises is that since all the instruction is being heard via the voice synthesizer, often the feedback distracts other users as well. Therefore the ability to control feedback is an important feature for software designed for the blind.

CHAPTER 3

Instructional Design Adequacy

This chapter will present important software design considerations which are relevant to visually handicapped and nonhandicapped students alike. These considerations include educational objectives, the content range/entry skills, content sequence, depth, accuracy, bias, readability, design of error messages and general instructional strategies.

EDUCATIONAL OBJECTIVES

A common fault of educational software currently on the market, according to the Council of Ministers (1985), is the lack of specifically stated educational objectives. The procedures for writing educational objectives and test items for CAI are similar to the procedures used in the development of other instructional materials. With regard to CAI, Owston (1987) points out two methods to be followed when studying a program's instructional objectives.

One method is to clearly state the program's objectives in behavioral terms so that changes in student's behavior can be measured. For example, if the software is designed to teach a simple skill such as the addition of two one-digit numbers, it would be possible to detect changes in a student's behavior from the beginning to the end of the program. However, when faced with a complex set of skills it becomes difficult to define the program's objectives in behavioral terms. As Owston contends, "other kinds of valuable learning take place that are not necessarily manifested by changes in student behavior. Such examples include changes in attitudes and the development of complex cognitive strategies" (1987, p.7).

The type of question items used in software programs are somewhat restricted by the judgement capabilities which can be programmed into the computer (Smith & Boyce, 1984). The types of question items which are feasible and in present use in CAI are as follows: true or false; multiple-choice; short answer; completion; matching, and constructed response (Keller, 1987).

In Bloom's (1956) "Taxonomy of Instructional Objectives in the Cognitive Domain" knowledge is the simplest of the learning process. It basically involves the recall of previously learned material. Comprehension goes one step beyond simply remembering the material. It consists of a number of activities which involve understanding: a) interpreting the material; b) translating it into another form; c) summarizing it; or d) predicting consequences.

Application is a higher process in Bloom's taxonomy, requiring a greater level of understanding. It refers to the ability to use learned materials in new, concrete situations, as in the application of rules, methods, laws and theories. Multiple-choice, short answer, completion and matching items may use application.

Analysis represents an even higher level because it requires an understanding of both the content and the structural format of the material. The learner has to break down the material into its component parts so that its organizational structure can be understood. This includes identifying the parts of the material, as well as understanding their relationship to each other and how they fit into the organizational structure. Except for the constructed response, all the other question strategies use analysis: the learner classifies the answer.

Synthesis is the learning outcome which stresses creative behaviors with an emphasis placed on the ability to put parts together into a new whole. This may include the production of a unique communication, i.e. a speech; a plan of operation such as a research proposal or the development of a set of abstract relations. Only constructed responses enable a learner to use this higher thinking skill.

Lastly one must consider the learning outcome entitled evaluation. This level is considered by Bloom to be the highest in the cognitive hierarchy because at this level a student has to demonstrate all the skills from the other taxonomic categories, while making a conscious value judgement based on clearly defined criteria. Once again, only constructed response items provide a student with the capacity to evaluate.

Most of the math software programs to be evaluated in the present study use only questions promoting lower learning outcomes such as knowledge and comprehension.

CONTENT RANGE AND ENTRY SKILLS

Poor content range is a common weakness of CAI software packages. In many cases, the programs are attempting to teach too much material in one session. Another problem is that programs often fail to begin with a review of the prerequisite skills, and expect the learners to cover content that may not have been learned (Council of Ministers, 1985). For instance, a program may begin with a multiplication problem using two digit numbers, while failing to ensure that the learners' entry skills are adequate to the task.

In some of the programs reviewed in this study, the only allowance for the learners' entry skill level is that the learners could choose amongst four different levels, ranging from simple to difficult. Other programs were examined in which this option did not exist at all.

CONTENT SEQUENCE

The purpose of having a content sequence in a program is to enable that the learners acquire a skill or concept with maximal ease. The content sequence presents information ranging from simple to difficult and concrete to abstract notions. The Council of Ministers (1985) have found through their courseware evaluation that the content sequence of many programs are poor. Common traits which the Council found were a lack of sequence and the omission of important steps following a new concept.

An example of a poor sequence is when students answer addition questions correctly but the program does not proceed with more difficult material; instead it stays at the same level. Conversely, the items may prove to be too difficult but the program does not respond to the learners' errors by providing easier problems. Few of the programs tested in the present study had such branching capabilities. Students could not alter the course of a program. According to Brandon (1988), these procedures have become standard in CAI software during the 1980's. If this is so, why is software designed for the blind a decade behind?

DEPTH

Depth refers to the detailed instruction given when new content is presented (Council of Ministers, 1985). The instruction might include supplementary explanations, rules, exercises, drills, etc. Often courseware does not provide the learners with enough depth. The students are presented with information on one screen page and required to advance to the next step without sufficient practice of the new skill. This is contrary to conventional classroom procedures, which recognize that extended practice needs to take place if the new skill is to become conceptualized by the individual.

ACCURACY

Usually, one can assume that instructional materials are free of both factual errors and errors in grammar or spelling. However, this does not appear to be true with much of the early software. It is still necessary to review carefully the factual content of each program for accuracy.

Ideally a program with grammatical or spelling errors should not be used (Lathrop & Goodson, 1983).

BIAS

As with accuracy, software which is free from biases should be the norm and not the exception. Yet, incidences of overt or subtle biases have been found in many programs. Lathrop & Goodson (1983) define bias as negative comments regarding sex, religion, age or ethnic background. With regard to sex, software's attributes such as symbols used, overall presentation and manner of response to input may unintentionally attract boys more than girls to computers. Fisher (1984) and Marrapodi (1984) admit that little research has been conducted in this area, but strongly warn that subliminal messages of sexism can be detected in commercial software packages. According to Fisher, software which is overly game-oriented may employ symbols which are socially preferred by boys. Marrapodi urges awareness of the types of reinforcers used in educational software. Packages which use aggression, spot allusions, or other traditionally gender- related rewards will probably appeal more to boys.

The Council of Ministers (1985) cites another example of content bias which is predominant in software produced in the United States: the absence of Canadian material. Very often math software packages do not employ the metric system in their content, and, for a Canadian student, this could prove to be a disadvantage.

Lastly, Rose (1984) states that all software should be reviewed for equity. This includes, "the complete or relative exclusion of a particular group in the content of the material" (Rose, 1984, p.51). Instructional materials representing disabled persons in a variety of roles may influence how others perceive the disabled and provide a positive role model for the handicapped students.

READABILITY

'Readability' refers to the text, its vocabulary level, sentence structure and amount of text presented on a screen page. For sighted students, the main concern is that the text's readability be at the reading level of the student. However the manner in which text is displayed on the screen may cause certain difficulities for blind users. Just and Carpenter (1980) noted that, unlike sighted readers, blind learners dependent upon a voice synthesizer cannot skip over material, or change the presentation rate to match comprehension, or pause over particular words.

Michaelis and Wiggins (1982) recommend solutions to these problems. They suggest that the amount of text shown on a screen page should be kept short and simple. To ensure even better comprehension, instructions should be given in the present tense and require an immediate response.

ERROR MESSAGES

Error messages are often encountered by computer users and yet the careful design of these messages is usually overlooked (Mizokawa & Levin, 1988). For example, a range error in one of the math programs reviewed in this study requires the user to input alphabetical material in a multiple choice format. If the user inputs a numerical answer, the program will not accept it, and provides no instuction to the user to input a letter instead. A built-in response evaluation scheme should be provided to check the input against the range of permissible responses.

Mizokawa and Levin recommend guidelines for the design of error messages. If an error message is given, it should have to be acknowledged by the user. At all times, error messages should avoid sarcasm, ridicule and harsh criticism. As well, comments such as "Oops!" can become annoying after only a few repetitions and therefore should be avoided. Lastly, the error messages must provide the users with information, rather than punishment or entertainment (Mizokawa & Levin, 1988).

INSTRUCTIONAL STRATEGIES

The term 'strategy' lends itself to many different interpretations. Faust (1977) defines strategy as " being composed of a series of displays which are presented to students and from which he (sic) is supposed to learn " (p.18). Merrill (1975) maintains that a strategy is comprised of " a sequence of specific tactics...[Tactic is defined as] ...a given display or presentation to the student of rather short duration" (p.220). Gropper (1974) on the other hand defines an instructional strategy as "the solution of an appropriate type of practice to meet the distinctive requirements of each different type of objective" (p.5). Hatfield (1975) views an instructional design strategy as "a set of interrelated and sequential activities, each serving a particular purpose for teaching a designated instructional objective " (p.109). Two different perspectives evolve from these definitions. Faust and Marrill imply that a strategy should be exclusively viewed in terms of computer display design. Gropper and Hatfield's interpret strategy as an activity.

These authors' definitions use concepts such as 'activities', 'displays', or 'practice' to define strategy. Ullmer (1988) resents limiting the word strategy to mere procedures. He claims that it strips the term of its communicative power. Ullmer credits Gerlach and Ely (1971) for recognizing the potential of the word strategy in the educational context.

Gerlach and Ely describe an instructional strategy as including a teacher's mode of disseminating information, the choice of resources and the definition of the students' role (in Ullmer, 1988). This definition also applies to CAI, where the computer can be substituted for the teacher. Ullmer also adds that with the onset of CAI there now exist several instructional strategy options: drill and practice, tutorials, games, simulations, dialogues, discovery learning and problem solving. For adapted software the choices become more limited. In the software available for evaluation during this study, only three instructional strategies were used: a) drill and practice (29 programs); b) tutorials (3 programs); and c) problem-solving (1 program).

Drill and Practice

The drill and practice strategy mostly comprises presenting questions and answers to the learner. In general, the best drill and practice programs make use of the computer's unique capabilities. Lathrop and Goodson (1983) find the computer's ability to handle random numbers or statements to be an advantage for use in drill and practice programs, since this permits the students' responses to be varied and interesting without causing any distraction to the subject at hand. The ability to generate an unlimited number of examples allows for diversity which can help to maintain the learners' interest.

Lathrop and Goodson maintain that computerized drill and practice programs in their many formats offer definite advantages over traditional methods of instruction. These advantages include comprehensive drills which test the students so that their entry level is known. Students can repeat a drill as many times as necessary in order to master the intended objectives. Wrong answers are discovered immediately, which reduces the possibility that an error may be learned.

Merrill (1985) is not so impressed with drill and practice programs. He prefers that a computer program act as a "participant in a conversation" instead of a replicated textbook (p.25). In the math software programs which were evaluated for this thesis, 27 out of 33 programs were drill and practice programs with varying formats. These programs do not possess all the qualities mentioned previously such as branching and scoring. However, these programs are more efficient to use than a brailler. The subject matter of these drill programs consisted of basic math skills within the four operations: multiplication; division; addition and subtraction.

Tutorials

The difference between a tutorial program and a drill and practice program, is that a tutorial is usually designated for individual use (Lathrop & Goodson, 1983). In order to be effective, tutorials must take full advantage of the interactive capacity of the computer. Tutorials play the role of a tutor to the user. The programs should first of all gather information on the learner's entry skills. Then the program should begin instruction at the particular level suited to the learners' knowledge of the subject matter. Incorrect responses cause the program to re-explain a concept and then to return to the main menu.

The three tutorial programs reviewed in the present study employed similar techniques for instruction. Yet they all share a common flaw. A problem is given at the beginning of each lesson. The program then asks the user to identify each new step. After each step is identified, the problem is repeated. Perhaps the developers felt that blind users would need to hear the problem frequently repeated? In the author's opinion, to hear a question repeated six times in one short program is an exaggerated burden.

Problem-Solving

The basic problem-solving strategy, as occaisonally encountered in math software makes little use of the computer capabilities, but rather presents math problems as they would appear in a regular textbook. The only computer feature used is the feedback given after each step in the solution. This feedback can become counter-productive since it may interfere with the user's concentration.

CHAPTER 4

Technical Adequacy

In this chapter factors as ease of use, the use of the computer's capabilities, available peripherals, reliability and programming errors will be discussed.

EASE OF USE

Can the intended user easily and independently operate the program? This is a question addressed by the Council of Ministers, (1985). For sighted users, most programs may prove simple enough to operate. However, for blind users they often require the assistance of a sighted person to read the 'Menu' or repeat the question for them. This not only diminishes the blind users' sense of accomplishment in being able to operate a computer, but can lead blind users to assume a dependent role (Laurer & Mowinski, 1986).

USE OF THE COMPUTER'S CAPABILITIES

Bitter & Wighton (1987) compiled a list of 22 educational software characteristics which were ranked major educational software evaluation agencies from the most to the least important. The agencies placed a strong emphasis on the types of criteria discussed in the previous chapter:

content, pedagogy and integration, with the highest ranking being awarded to the accuracy of the content. The second choice was on the effectiveness of the content presentation. The third choice focused on the appropriate use of technology, and the fourth choice looked at content again, with the integration of the program into the classroom. Among the least significant characteristics as chosen by the agencies was the use of computer features such as screen displays, color, sound, graphics and animation. A management system of record-keeping ranked lowest of the 22 criteria.

In Bitter & Wighton's study, the criteria were assessed for sighted users. Yet, when evaluating software for a specific audience such as the blind, characteristics which are ranked least significant for sighted users may turn out to be very important. An example of this is the use of technical features such as color, sound and graphics which may interfere with the instruction (Schreier, DeWitt, Goldberg & Leventhal, 1987). The program's instructional strategy should be vested in the audio mode of instruction, enabling users to practice their basic skills in an efficient and time-saving manner. The extent to which a program fully utilizes the computer's memory is also important to the blind population. Many of the attached peripherals or specifically designed software cannot operate with a limited amount of memory available (Young, 1984).

AVAILABLE PERIPHERALS

Attaching a voice synthesizer is not a complete, easy answer to the use of a computer for the blind (Krolick, 1984). As mentioned earlier, many problems arise when using a voice synthesizer is used. One problem is that virtually all the information displayed on the screen is echoed by the synthesizer. This often includes video attributes such as highlighting and blinking text, colors, graphics and ASCII codes! It is logical to recommend that programs which are specifically designed for the blind population should avoid such video features in order to permit blind users to operate a program without receiving confusing messages.

Synthetic speech involves linking electronically generated sounds to form words. Some methods of producing synthetic speech use a computer program to generate the speech. Others employ a special terminal and computer hardware for this purpose. In general, all text received from the computer can be spoken as words, spelled letter by letter, and/or reviewed (Ruconich, 1984).

As Laurer and Molinski (1984) explain, blind people must approach computer use in a different manner from sighted people. Voice output is not exactly comparable to screen output. As computer software is becoming more and more screen-oriented, it is becoming increasingly difficult for blind users to access these programs without much frustration and hardship. For instance, it is simple for a sighted person to skim quickly through a 'help menu' on the screen to locate commands. As for the blind user, listening to a long list of commands and trying to remember the right one may not be such an easy task to accomplish. One way to alleviate the task would be to have handy braille reference cards at the users' disposal, but this still does not address the main problem. "There is still a need for specialized software designed for efficient voice output" (Laurer & Molinski, 1984, p.4).

The most common way to make a computer talk is to plug a voice synthesizer into an external port or card slot on a personal computer. For the Apple II, two systems are available: the Echo II and the Votrax Type 'N Talk. The Echo II uses the Texas Instrument speech synthesizer which was initially introduced as a teaching toy. The Echo II can only be attached to Apple computers (Frantz & Wiggins, 1982). On the other hand, the Votrax Type 'N Talk is compatible with any microcomputer. Neither of these two synthesizers produce output which is truly natural. A particular limitation of voice synthesizers is the relatively low quality of speech output, demanding a period of learning and accomodation before understanding is guaranteed (Goodrich, Bennett, Paul & Wiley, 1980). Recently several firms have manufactured devices capable of storing human utterances. This enables the user to choose from a variety of voices, thus preserving the natural qualities of speech (Foulds, 1982). However, the flexibity of message creation is lost in the process.

The Echo II voice synthesizer developed by Street Electronics is a low cost synthesizer for the Apple II Plus and the Apple IIe. The Echo II features screen review which allows the user quickly to change modes of operation. This synthesizer has been very popular among blind students due to its low price of approximately $200.00 (Holladay, Navy, Kaysen, 1984). However it has limitations. It speaks whatever information is sent to it; also, in order to provide all the users' needs, such as repeating information or spelling words, the computer must first be programmed to perform these functions (Holladay, et al., 1984). Another inconvenience arises when words are pronounced correctly even though they are spelled wrong; conversely, some correctly spelled words are mispronounced. Fortunately, there exist devices which allow users to make program changes correcting particularly troublesome mispronunciations (Ruconich, 1984).

Lacking programming skills could constitute another drawback of using synthesized speech. If users did not write the program, it is unlikely that they will know how to modify the software to make program changes or to command the computer to recognize certain keys as function keys (Ruconich, 1984). Another problem is that the Disk Operating System, extended by the voice synthesis routines, often becomes too large, leaving inadequate memory for the operation of common application programs.

The device which provides visually impaired students with the most accurate picture of computerized output, including its line-by-line orientation on the screen, is the braille printer. Although braille printers range in price, they are all fairly expensive. However, the considerable flexibility these printers offer to blind users makes the price worthwhile (Lauer, 1984). One of the least expensive is the Cranmer Modified Perkins Brailler which has a price range of $3000.00. It is able to provide output in braille from one to twenty-five lines at a time according to instructions given from the keyboard (Maryland Computer Services Inc, 1983).

The Cranmer Brailler's flaws consist of not being able to handle fanfold paper, but rather individual sheets of braille paper which need to be hand-fed. It also has a slow printing speed of only 10 characters per second. It is extremely noisy when it is functioning, and there have been many complaints regarding its mechanical unreliability (Goodrich, 1984). This printer serves only to produce final hardcopy for braille readers and does not offer any options for editing.

Mid-range braille printers exist, such as the MBOSS-1 and VERSAPOINT, which cost about $5000.00 each. The advantages of these printers are that they are compatible with most Apple and IBM software without modification and are quite easy to master. As well, a continuous form-feed paper tractor can be used, operating at an acceptable noise level (VTEK, 1985; Telesensory Systems Inc, 1986). The VERSAPOINT printer designed by Telesensory Systems Incorporated (1986) also has the capacity to produce graphics and print in a foreign language. The disadvantages of the two systems are that they print at the slow rate of 10-20 characters per second and have a regular tendency to break down.

Much faster embossers are available such as the LED-120 or the THIEL, whose printing speed runs at 120 characters per second. Their $15,000.00 price restricts access to many blind users. These braillers appear to be geared towards commercial rather than individual users, since they are able to produce entire text rapidly, work at an acceptable office noise level and seldom break down (Ruconich, 1984).

When deciding which printer to buy, several important factors should be taken into account. As when purchasing a computer, the user has to decide which features of the printer are necessary: ease of use, noise level, speed and ability to print in both braille and print (Brunken, 1984). A vital consideration is the availability of service and whether the company provides resource people trained in accessories for the blind. This proves to be an essential consideration since professional assistance is often required in order to begin braille production. Also, the convenience of having the company's service department located in the same city can outweigh initial cost factors since a printer does not have to be sent to a different country for repairs (Scheur, 1984).

Following is an extract from the Raised Dot Computing newsletter about a blind person's experience in ordering a braille printer from a well-reputed company. The advice which Scheur (1984) offers could be a warning to other users regarding the intentions of companies which sell hardware to the blind.

> Recently, I found out that an established company that 'serves the blind' has been charging substantial premiums on items readily available to the sighted market. These premium prices would not be so bad if delivery was prompt....Because this printer is top of

> the line, I was not surprised at not finding it carried by local dealers. But, I knew I wasn't stuck because TSI [Telesensory Systems Inc] was selling a Radix 10 'turn-key' system [a braille printer]... I was told that delivery of the Radix turn-key system would take four to six weeks, and that the cost was $1,025.00.
>
> In contrast, I knew that some dealers could get the printer on special order in just a few days, charging only $925.00...Is it through laziness, indifference, lack of knowledge, or all three, that TSI selling this printer to the already beleaguered blind community, cannot afford a competitive price or timely delivery?...One is left with the feeling that TSI, long the leader in blindness technology, has become one more company engaged in the fine art of 'ripping off the blind', either by purpose or practice. Like any other, it must be dealt with warily. It seems to me that the TSI sales focus is not on the individual. Rather it is on the agencies and companies which employ or provide services for blind people. But those organizations get some cues from blind consumers about what to buy...(Scheur, 1984, p.7).

A great amount of research (Vanderheiden, 1982; The Trace Research & Development Center for the Severeley Communicatively Handicapped, 1982; Holladay, Navy & Kaysen, 1984; Schreier, DeWitt, Goldberg, Leventhal, 1987) has been undertaken to find technological means of providing the blind with adequate CAI accessibility. However in many school settings, the lack of sufficient funds and information about the results of this research has hampered the implementation of this knowledge.

PROGRAMMING ERRORS

All software programs must be technically sound; this implies free of programming errors (Williams & Williams, 1985). Of the 33 mathematics software programs reviewed, three had such serious programming errors that they were rejected out of hand, as representing an unethical imposition on test subjects (see next chapter).

A common error amongst all the programs is the inappropriateness of the voice synthesized language. For example, the basic statement "twelve times five equals sixty" once synthesized, may become "one, two, x (as in the letter x), five, equal sign, six, zero." It seems that even though the software is designed to teach mathematics, the voice synthesizer is utilizing a word-processing program. Other confusing symbols sounded by the software are as follows: the division sign is pronounced "slash"; and the subtraction sign is "hyphen" or silence.

Some of the software programs reviewed require the use of a brailler to answer the addition, subtraction and multiplication problems. Usually, it takes a considerable amount of time to type in a mathematics problem. The problems are typed in a vertical format and a learner may often need to roll the paper up and down to verify the numbers. Thus, the amount of time spent having to use the brailler defeats the purpose of using a mathematics computer program; which is to learn mathematics in a more efficient manner.

CHAPTER 5

Study A: The Preliminary Software Review

The following software evaluation study was based on the explicitly stated criteria of educational values discussed in the earlier chapters, and on the author's own expertise as a teacher of the blind. As a result of the study, three software packages will be selected representing contrasting levels of program quality for subsequent 'field-testing'. The use of the word 'field-testing' in this study implies that the testing will take place in a natural setting using realistic procedures with a small group of subjects. This is distinct from Dick's (1977) use of the term 'field-testing' by which he refers to testing of large groups with twenty learners or more.

RATIONALE FOR EVALUATION INSTRUMENT

The numerous checklists that are available for software evaluation are targeted towards a sighted audience (e.g. MicroSIFT, 1985; The Council of Ministers, 1985; California Library Media Consortium for Classroom Evaluation of Microcomputer Courseware, 1984). In order to generate a checklist which incorporates the specific needs of the blind, the author began by adapting evaluation criteria from both the "The Evaluator's Guide For Microcomputer Based

Instructional Packages" (ICCE, 1984) and The California Library Media Consortium For Classroom Evaluation Of Microcomputer Courseware, (1983). Criteria which the author judged to be pertinent to blind students' needs were added into the checklist. Steinberg (1983), Hannaford and Taber (1982) support the author's approach in this respect, indicating that a teacher of the handicapped may be more aware of the target population's needs than the software publishers themselves.

FORMAL AND WIDE AUDIENCE EVALUATION APPROACH

National software clearinghouses such as MicroSIFT and The California Library Media Consortium (CLMC) both use a formal wide audience evaluation approach in their evaluations (Owston, 1987). The first step in this approach is to devise a checklist incorporating criteria which the clearinghouse staff deem to be important for software evaluation. The second step includes locating and instructing teachers on evaluation procedures, which the teachers implement when they conduct the software reviews. In order to ensure uniformity amongst the evaluators, a monitoring process is used which ensures that established procedures are followed correctly. The evaluation results are then widely distributed via the electronic media (Owston, 1987).

MICROSIFT

MicroSIFT can be regarded as a pioneer in the field of software evaluation, currently reviewing over 500 software titles. Today, MicroSIFT is one of the largest and most influential clearinghouses for microcomputer-based courseware and related information (Lathrop and Goodson, 1983). It produces quarterly evaluation reports compare software packages within a specific area. This information is then sent to the RICE on-line database which is accessed by various educational resource service agencies (Owston, 1987). The fact that MicroSIFT's evaluation guidelines have been thoroughly field-tested as part of its project and are readily available, may account for its influence on other software evaluation approaches and its widespread use (Owston, 1987).

THE CALIFORNIA LIBRARY MEDIA CONSORTIUM

The CLMC's primary concern in developing their checklist was to encourage teachers to participate in the evaluation process by making it as short and simple as possible. The form was designed to be self-explanatory and accordingly, there is no accompanying documentation. A large portion of the form is devoted to evaluation criteria, divided into five areas: general design, ease of use, content, motivational devices used, and documentation available. Evaluators are asked to review critically various characteristics of the program within each area. In addition, a section on instructional content and objectives, asks for an assessment of the program design; and a further section asks the evaluator to describe the students and their responses to the program. Although this form was designed for use by teachers at a workshop, it has proved to be an excellent evaluation tool in general, owing to its simplicity and pertinence.

METHODOLOGY OF THE REVIEW

Based on the information reported in the previous chapters, and on aspects of the MicroSIFT and the CLMC checklists, a software evaluation checklist was developed for use with blind students specifically. The checklist was divided into three different areas: a) program operation and technical quality, listing 12 attributes; b) content and instructional quality, listing 9 attributes; and c) motivation, listing 6 attributes (Table 1).

Table 1 Checklist Criteria

Program Operation & Technical Quality:

1) No programmming errors present.
2) User informed which keys to use.
3) Wrong key doesn't abort program.
4) Accepts abbreviations for common answers.
5) Instructions given to quit.
6) Instructions can be skipped.
7) Able to modify program.
*8) Uses appropriate mathematical language.
*9) Sound of voice is comprehensible.
10) Uses computer capabilities well.
11) User can use program independently.
*12) Use of repetition is acceptable.

Content & Instructional Quality:

1) Follows sound educational theory.
2) Follows sound educational methods.
3) Content is accurate and unbiased.
4) Amount of learning justifies amount of time.
*5) No use of graphics and animation.
6) Objectives are clear and concise.
7) Questions are clear and concise.
8) User controls level of difficulty.
9) Timing is paced and flexible.

Motivation:

1) Effective feedback for correct answer.
2) Effective feedback for wrong answer.
3) Positive feedback more attractive than negative.
*4) Graphics not used to reinforce.
5) Timing doesn't distract user.
6) Instruction is user-friendly.

* Applies specifically to the blind
and partially sighted.

THE SOFTWARE REVIEWED

Thirty-three mathematics software programs were located for review, from The American Printing House For The Blind (Appendix I). Henceforward, the mathematics software programs will be referred to by their designated letter. Three of the programs were rejected due to serious programming errors. Four other programs used only graphics for instruction and were therefore judged useless for a blind user.

Following are examples of the programming errors which were detected. One program would stop functioning if a user input the wrong answer; another would state "Unidentified error in 300"; a third would stop a user from participating 2 out of 3 times and display a big "ZAP!" on the screen. Other programming problems included: a long loading time; the use of graphics only for instruction; no voice capability; always returning to the menu after only one question and the tendency to repeat a question and to answer too many times when a user inputs an incorrect response.

The remaining twenty-five software programs were assessed using the 27-item checklist. A "y" was recorded in Table 2 when a program successfully complied with one of the checklist attributes. If the program did not meet the criteria, a blank space was left beside the item. The numbers at the bottom of the Table refer to the total number of attributes each program contains.

GRID ANALYSIS

In order to increase the reliability of the checklist and provide for an objective selection of which programs to fieldtest, the data were inputed into the repertory grid analysis program of Higginbottom (1980). The variant used was a cluster analysis, as appropriate to the binary data of Table 2. The checklist was divided into the three areas included in the Table (program operation & technical quality; content and instructional quality; and motivation) and three separate grid analyses were conducted, one for each area.

In each analysis, the individual program attributes were considered as single variables. When a software program had one of the checklist attributes, it was given a positive response, a score of one. When a program did not have the attribute in question, it was given a negative response and a score of zero. Cluster analysis multiplies the loadings by the scores. Both the positive and negative discrepancies are counted, with the positive discrepancy being the number of times the score is one and the estimated value zero. The negative discrepancy is the number of times

Table 2 Assessment of 25 Math Software Programs

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
Program Operation & Technical Quality:																									
1) No programming errors present:	y		y	y	y	y	y	y	y				y	y		y		y	y	y			y	y	y
2) User informed which keys to use:	y	y	y	y	y		y	y	y	y		y	y		y	y	y	y	y	y			y	y	y
3) Wrong key doesn't abort program:	y	y	y	y	y	y	y	y	y				y	y	y	y	y	y	y	y	y	y	y	y	y
4) Accepts abbrev. for common answers:				y				y			y	y		y	y	y	y	y	y	y	y				
5) Instructions given to quit:		y	y	y																					y
6) Instructions can be skipped:	y								y	y	y		y			y									
7) Able to modify program:																									
8) Uses appropriate math language:		y	y		y	y			y		y		y	y		y		y	y	y	y		y		y
9) Sound of voice is comprehensible:	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y		y	y	y	y	y	y		y
10) Uses computer capabilities well:	y			y	y	y	y	y	y				y	y		y			y	y			y		y
11) User can use program independently:	y	y			y	y	y	y	y	y	y	y	y	y	y	y		y		y			y	y	y
12) Use of repetition is acceptable:	y			y	y	y	y	y					y	y		y	y			y			y		y
Content & Instructional Quality:																									
1) Follows sound educational theory:	y	y	y	y	y	y	y	y	y	y	y		y	y		y	y	y	y	y	y	y	y	y	y
2) Follows sound educational methods:	y			y	y	y		y	y		y		y	y		y		y	y	y			y		y
3) Content is accurate & unbiased:	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
4) Amount of learning justifies amt. of time:	y			y		y	y	y					y	y		y				y			y		y
5) No use of graphics and animation:	y		y	y	y	y		y	y		y	y	y	y		y	y	y	y	y			y	y	y
6) Objectives are clear to user:	y	y	y	y	y		y	y	y	y	y	y	y	y		y		y		y			y	y	y
7) Questions are clear and concise:	y			y	y	y	y	y	y	y	y	y	y	y	y	y				y			y		y
8) User control level of difficulty:	y			y							y			y					y	y			y		y
9) Timing is paced and flexible:	y			y	y	y	y	y	y	y	y	y	y	y		y	y	y	y	y	y	y	y	y	y
Motivation:																									
1) Effective feedback for correct answer:	y			y	y	y	y		y		y		y	y		y			y	y			y	y	y
2) Effective feedback for wrong answer:	y			y	y	y	y		y		y		y	y		y						y	y		y
3) Pos. feedback more attract. neg.:	y			y	y	y		y	y		y		y	y		y			y	y			y		y
4) Graphics not used to reinforce:	y	y	y	y	y	y			y	y	y	y	y	y		y	y	y	y	y			y		y
5) Timing doesn't distract user:	y	y	y	y	y	y	y	y	y	y	y	y	y	y		y	y	y	y	y	y	y	y	y	y
6) Instruction is user-friendly:	y			y	y		y		y	y		y	y			y								y	
Total Number of Characteristics:	23	11	12	23	21	19	17	18	21	14	16	12	23	22	7	24	10	15	17	22	8	7	22	12	23

the score is zero and the estimated value one. Initially, the analysis considers each variable as a separate component. Then, the two most similar variables are joined to form a larger cluster. The analysis continues until separate components are established accounting for maximum variance among the program attributes.

For the area of program operation and technical quality, the output from the cluster analysis (Appendix II) revealed two independent families of attributes. They were: a) instructions given on how to quit the program; and b) ability to modify program needs. For the area of content and instructional quality, the analysis indicated three independent families of attributes: a) the program follows sound educational theory; b) content is accurate and unbiased; and c) timing of the program is self-paced and flexible. For the area of motivation, there were three independent families of constructs: a) effective feedback given for correct responses; b) graphics not used for reinforcement; and c) timing not distracting to user.

Each software program was then rated on the seven attributes found to represent the most variance in the three checklist areas (Table 3). The programs were rated on the basis of the total number of attributes, in terms of poor, fair or good. Programs which received the maximum score (7) were characterized as 'good' programs, since they contained

Table 3 Math Computer Programs' Ratings According To The 7 Primary Attributes

Program Operation & Technical Quality:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
5) Instructions given to quit:		Y	Y	Y																					Y
7) Able to modify program:																									
Content & Instructional Quality:																									
1) Follows sound educational theory:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
3) Content is accurate & unbias:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
9) Timing is paced & flexible:	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Motivation:																									
1) Effective feedback for correct ans:	Y			Y	Y	Y	Y		Y		Y		Y	Y		Y			Y	Y			Y	Y	Y
4) Graphics not used to reinforce:	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y			Y		Y
5) Timing doesn't detract user:	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Total # of Characteristics:	6	5	5	7	6	6	5	4	6	5	6	4	6	6	1	6	5	5	6	6	4	4	6	5	7

all of the qualities indicated by the analyses to be necessary for effective software. The program which yielded the minimum score (1) was considered 'poor', for it lacked most of the listed criteria items. Programs which received a mid-range score (4) were labelled as 'fair' programs.

The programs from each rank (1, 4 or 7) were then screened by the author, with respect to their suitability for the blind user. Three representative software programs were chosen, one from each main category. The next step was to field-test these three programs of varying quality and to find out whether the quality of the program would influence a) the blind users' improvement in addition and subtraction; b) their attitudes towards mathematics as a subject; c) their attitudes towards mathematics computer software in general; and d) their levels of aspiration with regard to bettering themselves in mathematics.

SELECTION OF 'GOOD' PROGRAM

Programs D and Y both received the highest score of 7. Program D is a drill and practice program which offers both large print and voice capabilities. It permits the users to choose the level they want to work at; the levels range from Grade 2 to Grade 6. It also provides the users with a choice of the operation they wish to practice: addition, subtraction, multiplication or division. The program also has certain negative characteristics. First of all, it program has a very long loading time (about 1 minute) which may cause the students to become frustrated or to experience anxiety levels detrimental to their performance (Schneidermann, 1980; Golas, 1983). Secondly, the program requires the use of a brailler, as well, which may interfere with students' attitudes towards mathematics or mathematics software.

Program Y is also a drill and practice program which focuses on the addition of one and two digits. The software is aimed at Grades 4 to 5. It has voice capabilities. Is easy to operate, and has clear and concise instructions. This program follows a multiple-choice format. It allows for only one try per problem which may cause some students to be hesitant to answer since they know that they have only one chance. However, in view of the technical problems observed with program D, program Y has been chosen as the representative for the 'good' program.

SELECTION OF 'FAIR' PROGRAM

Programs H, L, U, and V each received a mid-range score of 4. Each of these programs will be described in detail.

Program H is a drill and practice program whose sole content is subtraction with one-and two-digit numbers. This program is very basic at a Grade 3 to 6 level. It contains explicit instructions which enable a user to operate the program independently. The user only needs to supply the questions with numerical answers. However, this program does not utilize any creative, innovative or effective features of the computer. It presents problems simply as they would be written in a workbook. The feedback in this program is quite redundant and does not use appropriate language for mathematics as explained in chapter 4. This makes the questions difficult to understand. The user needs to learn that "hyphen" or silence means minus, and that the program reads the digits individually.

Program L is a drill and practice on the four operations. The level of difficulty cannot be adjusted. Although the program does indicate that it is set at Grade 4 to 6 level, its questions seem to be targeted towards a grade 3 or 4. This program allows the user to choose an operation (e.g. addition), and provides the user with one example. Once the user has typed in a response the program returns to the MENU, regardless of the accuracy of the answer! If a user wants to continue answering questions, the program must be reloaded. This program also reads out the mathematics symbols inappropriately. Its feedback responses use many slang comments. Examples of the comments are: "Right on!"; "Wow!"; "Too bad!" and "Oops!".

Program U is a drill and practice program concentrating on 'carrying over' in addition. The level is at Grades 5 to 6. The program's strategies appear to be designed for a sighted rather than a blind user. First of all, the problem is read (e.g. "two, two plus one, four"). Then it repeats three times, "What goes above?". An arrow points to a blank, which signifies a number which goes above the first digit when carrying over in addition. A blind user may have difficulty in answering the question due to the inability to perceive the location of the arrow. As well, the repetition of "What goes above" may interfere with the user's recall of the numbers. Secondly, every problem is framed by X's. Therefore, at the beginning of each question all the user can hear are the X's being sounded out by the voice synthesizer. The feedback of this program is minimal. If the answer is correct there is no response, if an error is made, it says "Try again".

Program V is a drill and practice focusing on subtraction with borrowing. The level is at Grades 5 to 6. This program follows the same format as program U. Once again, the strategies appear to be oriented towards sighted users. An arrow is placed on the first digit of a set of numbers, with the intention to borrow when subtracting.

Instead of repeating the numbers, the question "What goes above?" is needlessly repeated three times. Also the letter X is used as a decoration for each question and therefore, needlessly repeated. As with program U, there is only one feedback response which says "Try again" if a mistake is made.

Program H has been chosen to represent the 'fair' program for field-testing, since it contains none of the major flaws observed in programs L, U and V.

SELECTION OF 'POOR' PROGRAM

Program O was the only one to receive the minimum score of 1. This program is a drill and practice program which concentrates on addition and subtraction with one and two digit numbers. Its level ranges from Grades 4 to 6. This program is difficult to operate since it offers no instructions. It also has the folowing distracting feature. It first presents a question, and then a series of random numbers repeated very quickly. The word 'Go' appears, and the program begins counting backwards from fifteen. This all occurs while the user is thinking of a response to the question and educationally speaking has neither rhyme nor reason! If a user types in an answer, the program does not respond immediately, but continues to count for a few more seconds. This results in the user having no idea if his/her answer has been accepted, until the computer offers one of two responses: "Good!" or "No, the answer is ..". Since there are no other programs to choose from in this category, program O has been chosen to represent the 'poor' software level. In the next chapter, the programs representing the three software levels are formally field-tested.

CHAPTER 6

Study B: Field-testing of Selected Programs

SAMPLE OF THE TARGET POPULATION

A limited number of anglophone blind students attends schools in Montreal and the surrounding areas. As indicated in Chapter 4, the software programs ranged from a Grade 4 to 6 level, and for the following field-tests, the author had access to only seven blind students enrolled in these grades.

All blind or partially sighted students integrated into a regular school in Montreal must fulfil similar criteria. These criteria consist of: average or above average IQ; ability to use devices (i.e. typewriter) or techniques (i.e. braille or Nemth code for mathematics) to do schoolwork; and appropriate social behaviors. The fact that the students are all blind or partially sighted implies that they all share similar traits for accessing information, for instance, the need to rely on a voice synthesizer to be able to read a computer screen.

On the above basis, the field-test sample is assumed to represent a relatively homogenous population. However, the mathematics ability of the individuals in the sample may differ. Both psychological and curricular testing is conducted on most blind and partially sighted students who are integrated into regular schools. The author was able to gain access to the students' mathematics results on norm-referenced tests such as Key Math and part of the RAT. Based on this information, the students were classified according to their mathematical performance (Table 4).

TESTING DESIGN

A three-group pretest-posttest design (Campbell and Stanley, 1966) was employed in order to study the possible effects of the three mathematics computer programs ('good', 'fair', and 'poor') on subjects' learning; their general attitudes towards mathematics and mathematics software were also surveyed. The first stage in the design was the administration of a pretest containing mathematical problems and attitudinal questions (Appendix III). The second stage was the experimental treatment (i.e. the computer program); and the final step involved the administration of the posttest (Appendix III). The differences in mathematical ability due to the application of the treatment were then determined by comparing the pretest and posttest scores (Campbell & Stanley, 1966). Borg and Gall (1983) stress that a key problem with the pretest-posttest design "is establishing suitable control so that any change in the posttest can be attributed only to the experimental

Table 4 Description of Sample Population

N	Sex	Age	Grade Level	Acuity (Left/Right Eyes)	Math Level	CAI Experience
1	M	10	4	L.E. 20/200 R.E. 20/240	Average	Yes
2	M	11	5	L.E. 20/420 R.E. 20/400	Good	Yes
3	F	11	5	L.E. Light Perception R.E. No Light Perception	Good	Yes
4	M	13	6	L.E. Light Perception R.E. Light Perception	Average	Yes
5	F	12	6	L.E. 20/400 R.E. 20/320	Poor	Yes
6	M	13	6	L.E. Light Perception R.E. Light Perception	Poor	Yes
*7	M	11	5	L.E. 20/400 R.E. 20/320	Good	Yes

*7: Subject used in pilot study

treatment...Many extraneous variables need to be controlled in order to allow an unequivocal interpretation of experimental data" (p.663). Campbell and Stanley (1966) have identified eight types of extraneous variable, some of which may be pertinent to the design of the current study.

The subjects in this study each worked alone when using the computer. This was done to ensure that their learning or attitudes would not be influenced by other students.

'Instrumentation effects' refer to effects which occur because one of the measuring instruments has been changed; either the pretest or posttest contains easier items. To avoid this contamination of the results, the pretest and posttest computation items were identical for this study.

The entire duration of the study was less than within one hour. It is therefore very unlikely that additional learning took place from an external source, or that the subjects developed physically, socially or intellectually during the testing period. In using the before-an-after design within a short time period, there is a chance that the subjects may have become test-wise. To reduce this testing effect as far as possible, the order of the questions was changed on the posttest.

Another factor which may affect the results is the allocation of the 'good', 'fair' and 'poor' programs to the sample population. This was counter-balanced as well as possible. The subjects who were ranked as good in mathematics received either a 'fair' or 'poor' program. The subjects who were ranked to be average in mathematics, received either a 'poor' or 'good' program. Lastly, those subjects whose test results indicated that they were poor in mathematics were assigned either the 'fair' or 'good' program. This was the best means available of ensuring that extreme scores would not occur in the results. Students who received high mathematics scores might have performed too well if assigned the 'good' software program. Conversely, the students who scored poorly might have produced very low results using the 'poor' program.

The author was able to gather information from the schools concerning the subjects' previous computer experience. It was found that only one subject in the sample had never used a mathematics program before this study. This information was also sought on the posttest to reaffirm the author's findings.

The subjects' familiarity with computers reduced the possibility that they guess a response to an item. Borg and Gall (1983) state that subjects who lack information about a particular topic will express an opinion in order to conceal their ignorance or because they feel pressured into expressing an opinion.

INSTRUMENTATION

The purpose of the pretest was to assess the subjects' entry level in mathematics. Initially, it included eleven subtraction and nine addition question statements using one-digit numbers. (These were later modified to two-digit statements: see next section.) The posttest served several purposes: first, to verify if there was any learning gain after using a mathematics computer program, secondly, to measure the students' attitudes toward the mathematics computer program's features, mathematics as a subject, and their students' levels of aspiration with regard to mathematics; thirdly, to gather demographic information regarding sex, age, grade, degree of vision, and etiology. Factual questions were also asked regarding computers and mathematics. The addition and subtraction questions were identical to those presented on the pretest but were in reverse order.

With regard to the questions dealing with attitudes, the author created a set of 3 point scales. Conventional Likert scales require respondents to indicate their agreement with each statement on a 5-point scale ranging from: strongly agree to strongly disagree (Henerson, Morris, Fitz-Gibbon, 1978). However, the author decided to use 3-point scales for the following reasons. Firstly, the questions were to be given orally and it might have been difficult for the blind subjects to process five choices. Secondly, the subjects were all at the elementary level. Consequently their vocabulary and comprehension level may not have enabled them to make the distinction between strongly agree and agree, or strongly disagree and disagree. To facilitate the subjects' comprehension, simple 'Yes-No' and 'Like-Dislike' formats were stressed.

For the first question on the posttest, a four-point scale was utilized. It asked the subjects to choose a statement which best described how they felt about the computer program they had just used. This was an important question and was therefore placed at the beginning of the questionnaire, as suggested by Borg and Gall (1983). The author was careful to make the question sound interesting and non-threatening, in order to motivate the subjects to continue responding to the remaining items; also to alternate between questions which reflected a positive attitude and those reflecting a negative attitude.

Demographic information was also sought on the post-test. In one of the questions, the respondent was able to comment on a mathematics computer program they had used before this study. Another question related to the use of the computer for other topics than mathematics. For the final three questions, a 4-point scale was utilized to assess the respondents' level of aspiration in mathematics.

CONSENT TO PARTICIPATE

The consent of each school's principal was needed in order to perform the testing. The principals were contacted by the author and the study was described in detail to them. They were asked not to discuss the study with any member of their staff. The author assured each principal that the subjects' identities would remain anonymous. It was not considered necessary to contact the parents since the testing was to be administered during the itinerant teacher's regular duties.

PILOT STUDY

A pilot study was first conducted. The actual testing was not done by the investigator but by another person, the experimenter. The pilot study served the purposes of checking a) whether the procedures to be implemented were well understood, b) the operation of the computer program, and c) if the wording of the question items and response statements was easily understood by the subject. Borg and Gall (1983) maintain that pilot studies can provide feedback from the subjects and other persons involved which may lead to important improvements in the study. Borg and Gall also recommend that for a pilot study to be valid, the subjects should be similar to those which will be utilized in the actual study. Due to the small sample size, only one subject was used in the pilot study.

The experimenter was an itinerant teacher whose job included being a tutor and resource person for blind and partially sighted students integrated into the regular school system. The itinerant teacher is assigned a certain number of students, and visits them on a regular basis in their schools.

The subjects in this study were all accustomed to seeing this teacher and all of them appeared to feel at ease with her. The test took place in a natural one-on-one classroom setting.

To avoid an experimenter bias effect, the experimenter was not given any indication of the purpose of the study, nor had any idea that the software programs were ranked according to quality. All she knew was that the institution for the blind had purchased three mathematics programs and the author was interested to determine their effectiveness. It was to the author's advantage that the itinerant teacher was not familiar with computers nor with computer programs.

On-site requirements were as follows:

1. One Apple IIe or II Plus microcomputer with attached floppy disk drive and monitor.

2. An Echo II or Echo Plus speech synthesizer with an interface card that fits into slot #3 of the Apple.

Important Note: These two voice synthesizers cannot be used on an Apple IIc.

All the schools which collaborated in this study had access to this equipment. The itinerant was given the testing materials on the day of the testing, and explicit instructions for loading the program and conducting the test (see Appendix IV).

The pilot test proceeded as follows. The pretest was administered orally, consisting of twenty addition and subtraction questions. The subject answered the questions orally, while the experimenter recorded the responses. The experimenter then loaded the computer mathematics program. Since this subject had received high scores on the mathematics norm-referenced tests, he was given the fair program. The subject then worked with the mathematics program until its completion; all instructions were provided by the program. The experimenter then proceeded to administer the posttest, reading the items orally and recording the subject's responses.

The pilot study indicated that the entry level of the computation questions was too simple. As a result, the addition and subtraction questions were modified from one-digit to two-digit numbers. Apart from this modification, the measures remained the same. The experimenter reported that the question items were otherwise clear and did not present any problems to the subject. The experimenter also reported that no problems arose in the operation of the computer program.

THE FINAL FIELD-TEST

The final field-test replicated the pilot study in all respects except the modifications to the questionnaires (see previous section). It took six days to conduct the fieldtesting on each subject. Only one subject tested at a time. This secured the independence of the students' responses. All subjects were given ample notice before the testing would take place. The day and time of each appointment was negotiated with the classroom teacher, itinerant and school principal, to ensure that the time was convenient. The testing took place in a naturalistic setting, in the subjects' own school with the itinerant teacher acting as the tester. The approximate length of time for the testing was a little under one hour.

RESULTS

Once the field-testing had been completed, the data were collated, and the subjects' performance assessed on the basis of the pretest and posttest. Table 5 displays the pretest and posttest scores, and examines the difference scores in relation to the assigned mathematics program and the mathematics level of each subject.

Table 5 Pretest & Posttest Scores

Subject	Computer Program	Pretest Score	Posttest Score	Difference Score	Mathematics Level
1	poor	4	5	+1	average
2	fair	16	15	-1	good
3	poor	17	18	+1	good
4	good	18	19	+1	average
5	good	18	19	-1	poor
6	fair	13	12	-1	poor

None of the programs appears to have had an effect upon the subjects' mathematics computation skills. The difference scores between the pretest and posttest show a small range of shifts only. The range of shifts in question renders it most unlikely that learning took place. Two subjects (#2 and #6) scored lower in the posttest than on the pretest, even though the first was ranked as a high mathematics achiever on the basis of recent school results.

The attitudinal measures did, however, provide useful information concerning subjects' attitudes. Table 6 illustrates a) the subjects' overall evaluation of the computer program which they had used during the field study and b) their levels of aspiration in mathematics.
The frequencies of responses to the posttest questions as a whole are given in Appendix III.

Table 6 Subjects' Responses to Math Questions

	Subjects: 1	2	3	4	5	6
#1. Please give me your overall opinion of the math computer program you have just used.	Very good/ very useful	No good/ ot useful	Good useful	Good useful	Very good/ very useful	Good useful
#32. Do you think computer programs are able to help you in math ?	Yes	About right	No	Yes	Yes	About right
#34. Do you find math computer programs interesting to use?	Yes	About right	Yes	Yes	Yes	Yes
#45. How much do you enjoy math?	Not very much	A lot	Not very much	Not very much	A lot	Not very much
#46. Is getting better at math important to you?	Quite imp.	Very imp.	Very imp.	Very imp.	Very imp.	Quite imp.
#47. How good do you think you are at math?	Good	Good	Not very good	Good	Not very good	Not very good

CHAPTER 7

Discussion of Results and Conclusion

In this final chapter, the field-test results will be discussed, guidelines will be offered to software manufacturers and purchasers regarding the problems most commonly found in adapted software for the blind and partially sighted.

DISCUSSION OF RESULTS

The field-test results revealed no obvious differences between pretest and posttest scores. This may certainly have been due to the small sample used in the testing; it may also have been due to the use of 'weak' treatments (Borg & Gall, 1983). It is difficult to produce a treatment which is strong enough to have an effect upon the dependent variable. In the present study, the treatments (computer programs were administered only once to the subjects. Perhaps they would have yielded greater gains if used over a longer period. Unfortunately, this would have been impossible with the highly specialized sample of subjects which was currently available. It should be noted that a few of the blind subjects experienced difficulty in using the software programs, particularly the 'fair' and 'poor' ones. (See the descriptions of the 'poor' and 'fair' programs in Chapter 5 for a discussion of their technical flaws.) However, they seem to have hesitated to express these problems as criticisms. Even though technical difficulties did arise during the testing, the subjects were not critical of these problems in their responses to the posttest items concerning the program's attributes. This lack of criticism may have been for two reasons: a) the students' inexperience with good software, or b) the possibility that they do not have the confidence to criticize educational materials generally. All but one of the subjects judged the program in question as 'good and useful'; but it is distinctly possible that <u>none of the software programs was adequate to teach blind students at this educational level.</u> It can be assumed that most of the subjects liked the program they used because they had never worked with a better one. They may be ignorant of the fact that better software is available, since they have not had access to the vast world of standard software (Vanderheiden, 1982; Lauer & Mowinski, 1984; and Young, 1984).

On these bases, it is also possible that <u>these subjects were not good judges of the software</u>.

Only one subject (#2) in the sample reported the program which he used as being 'no good, not useful at all'. This subject had been assigned the 'fair' program and was ranked as having a high average in mathematics. He may therefore have had higher standards for evaluating software than the other subjects. Indeed, this subject claimed a relatively high degree of previous computer experience - more than four hours a week - and demonstrated his critical judgement in his judgments of the other programs he had used. The response statements for this question should certainly have been more specific, in order to tease apart the different effects of program quality and user experience.

It would have been more informative, for example, to have asked the subjects precisely how many hours a week they use computers. By doing this, it could have been determined whether the amount of time spent working with the computer was related to the stringency of subjects' evaluations of computer software.

The other possibility concerns the possibility that subjects were <u>unwilling</u> to criticize the software. Students are not often in a position to judge educational materials, and are unaccustomed to having a staff member value their opinion. The subjects may therefore have attempted to please the experimenter by answering in accordance with the responses they perceived as being the experimenter's preference (i.e. an experimenter bias, or 'halo effect').

It is also possible that the subjects may have felt obliged to judge the software favorably since it was presented to them by their itinerant teacher. Thus, an experimenter bias effect may have occurred without the experimenter being aware of it. "The method of transmission of the experimenter's expectancy to the subjects is not clearly understood,... yet the experimenter bias effect does appear to be a real threat to the internal validity of experiments" (Borg & Gall, 1983, p.646). One way of finding out in future if this was a contributing factor would be to have the testing conducted by several experimenters. It could then be determined whether it makes a difference who administers the test. Borg and Gall suggest that an effective way to avoiding this effect is to train naive experimenters to work with the students. The present researcher had considered this when designing the study and did not work directly with the subjects herself for this reason. The investigator was careful not to discuss the study's hypothesis with the actual experimenter; and the experimenter had no training in computers.

The training of the experimenter to be completely objective throughout the testing session was another method employed to circumvent this effect. The instructions to the experimenter had included emphases on not discussing the tests' items with the subjects; on not revealing one's opinion about the program's attributes; and on refraining from making comments concerning the items on the pretest and posttest. Lastly, the experimenter was to be conscious of the tonality and emotional impact of the voice when reading the items and possible responses. However, the possibility that the experimenter bias effect contributed to the results must certainly be taken into account if the present results are in any way to be generalized to the target population.

The subjects were all asked whether they enjoyed mathematics as a subject. More than half of them admitted to disliking mathematics. However, all of the subjects expressed a desire to improve in the subject. The subjects who ranked themselves as not being very good in mathematics did not appear to be helped by the software program. Rather, it seems that they judged themselves more harshly than the mathematics programs. When asked whether they thought that mathematics computer programs could help them in mathematics, only Subject #3 answered no. However, this same subject responded that he was not very good at math. Apparently, the software programs did not help the subjects to feel more competent in math. Instead, it appears that the software programs may have reinforced the already negative perception of those subjects who felt they were no good in math. This assumption is contrary to the purpose of drill and practice programs which are designed for low achievers who require additional practice (Lathrop & Goodson, 1983 & Taber, 1983).

Ultimately, the researcher speculates that motivation for using computers as well as wanting to improve in math may very well have affected these subjects' evaluations of the math programs presented to them. The earlier findings of the study (Chapter 5) support the notion that math software currently designed for the blind or partially sighted tends to be neither effective nor appropriate. The study thus supports the conclusions of other authors, mentioned earlier, on this matter (Laurer & Mowinski, 1986; Young, 1984; Moyer & Scadden, 1987; Goodrich, 1984; Lewis, Harrison, Lynch & Saba, 1988; and Vanderheiden, 1982).

CONCLUSION

Many problems were discovered in the courseware evaluated by the author (see Chapter 5). These included: program instructions being too complex and lengthy, the need for more refined student error analysis and corrective feedback to the student by the computer; technical flaws and programming errors. As part of an ongoing evaluation process, the thesis thus offers guidelines to publishers and manufacturers to follow in remedying the present shortcomings of software designed for the blind and partially sighted. Ideally, the problems should be identified by formative evaluation procedures, before the software is marketed, rather than afterwards as in the current study. Via formative evaluation, developers have the opportunity to assess if a software program is capable of teaching, and to make revisions if necessary (Smith & Boyce, 1984). Even though it can be an expensive process, to implement, the advantages of formative evaluation outweigh its costs. If evaluation procedures are used as an integral part of the instructional development process in this manner, the effectiveness of the product will be substantially improved (Baker & Alkin, 1973). This increases the likelihood of creating a product which will ultimately be easier to market.

Weston (1987) proposes the inclusion of experts and learners in the formative evaluation of instructional products. Implementing a team approach including a subject matter expert, an instructional designer, and a programmer can all contribute to the design of the program. The feedback received from the intended learners can help to identify any problems with the program. If time is taken to research the specific needs of the target audience, a high-quality product which is appropriate to the intended learners' needs will undoubtedly emerge (Smith & Boyce, 1984).

Another important way of improving software designed for the blind is to increase the sensitivity among manufacturers and developers regarding the importance of effective delivery of computer hardware for the blind. Guidance as to the practical aspects of using special technical modifications can provide manufacturers with a better understanding of the problems they are trying to solve, and can lead to the design of more effective special facilities.

Developers should realize that blind individuals use considerably different strategies for organizing and presenting information on the computer screen. Even though voice synthesizers are currently providing the blind and partially sighted with access to software, they are providing neither equivalent nor optimal access as compared to that of sighted individuals. According to The Trace Research and Development Center for the Severely Communicatively Handicapped (1982), the software problem is further complicated when modified hardware places demands on the computer's operation system. For example, the program which tells a computer how to operate a voice synthesizer cannot necessarily cope with input instructing the computer on how to operate a printer.

Commercial software producers have been accused many times of being in no hurry to fill the gaps in available special education CAI materials (Vanderhelden, 1982; Laurer & Mowinski, 1986; and Young, 1984) In fact, this is such a common problem that a journal dedicated to improving the educational opportunities for children and youth with disabilities has been established, entitled "Closing The Gap". The lack of abundant good special education software is the main reason that a full-scale application of CAI to the instructional needs of disabled students has not yet being achieved (Trace Center, 1982). It is probably safe to presume that the relatively small special education population, the complex nature of special education CAI software, and the costs involved in producing such software, have all contributed to this situation.

Teachers of computer aids for the blind are in a double bind. They are overwhelmed by the constant flow of new adaptive aids and updates, which are largely underdeveloped and lack sufficient documentation. At the same time they need to keep up with the ever growing computer field. Yet budget resources for their training are often lacking. The researcher therefore suggests that manufacturers include adequate documentation along with the software program. This should include suggestions on how to integrate the program into a curriculum. Instructional activities which are not inherent in the software should be suggested.

Lesson plans should be presented which integrate these missing events into an instructional classroom setting.

Much remains to be done to improve the sad situation of educational software for the visually handicapped. In Laurer and Mowinski's opinion, (1986) CAI manufacturers are currently taking full advantage of the public's ignorance by introducing systems which keep the blind users' unnecessarily dependent upon experts. Another source of concern is the rapidly changing nature of the computer industry. The standardization of codes for use with all machines would greatly relieve the blind user from depending upon experts, because each change has the potential of denying visually handicapped users access to computer technology. Too few resources exist for training and equipping users who could benefit from computer aids. Very often blind users need to be more knowledgeable about computers than their sighted counterparts. A cohesive and sustained effort made by braille professionals, equipment manufacturers and visually impaired consumers alike is needed to ensure that the blind have full access to the computer.

In summary, it is crucial that greater stability within the hardware should be promoted, and exploitation of the blind users' dependency on experts lessened. Communication through networking with others, and sharing information about computer techniques, appears to be the key to improving computer access. Institutions, agencies, manufacturers, developers of software and the visually impaired individuals themselves must be willing to make the effort to increase interpersonal contact and to exchange information. A greater awareness is needed of each organization's financial and staff limitations including those of the manufacturers and developers of software, in order for solutions to be found to these troublesome problems. Only then, when communication is generated among evaluators, trainers, information disseminators, and manufacturers of special aids will the amount of good, useful software for the blind be increased.

REFERENCES

Allan, R. (1984). Software in Evaluation: Scarborough Board of Education. Computers in Education, 11(1), 14-16.

Ashcroft, S. C., & Bourgeois, M. (1980). Recent technological developments for the visually impaired: State of the art. Journal of Special Technology, 3(2), 5-10.

Baggaley, J. P. (1987). Continual Response Measurement: Design and Validation. Canadian Journal of Educational Communication, 16(3), 217-238.

Baker, E. L., & Alkin, M. C. (1973). Formative Evaluation of Insructional Development. Audio Visual Communication Review, 21(4), 389-418.

Bitter, G. G., & Wighton, D. (1987). The Most Important Criteria Used By The Educational Software Evaluation Consortium. The Computing Teacher, 14(8), 7-9.

Bloom, B. S. (1965). Taxonomy of Educational Objectives, Cognitive and Affective Domains. NY: Longman Inc.

Bloom, B., Hastings, J., & Maddus, G. (1971). Handbook on Formative and Summative Evaluations of Student Learning. NY: McGraw-Hill.

Borg, W. R., & Gall, M. D. (1983). Educational Research: An Introduction. (4th ed.). NY: Longman Inc.

Borich, G. D. (Ed.). (1974). Evaluating Educational Programs & Products. Englewood Cliffs, NJ: Educational Technology Publications.

Bower, G. H. & Hilgard, E. R. (1981). Theories Of Learning. (5th ed.). Englewood Cliffs, NJ: Prentice-Hall, Inc.

Brandon, P. R. (1988). Recent Developments in Instructional Hardware and Software. Educational Technology, 28(10), 7-12.

Briggs, L. G., & Wager, W. W. (1981). Handbook of Procedures for the Design of Instruction. (2nd ed.). Englewood Cliffs, NJ: Educational Technology Publications.

Brunken, P. (1984). Independence For The Visually Handicapped Through Technology. Education Of The Visually Handicapped, 15(4), 127-133.

California Library Media Consortium For Classroom Evaluation of Microcomputer Courseware, (1983). In Lathrop, A., & Goodson, B. (1983) Courseware in the Classroom, CA: Addison-Wesley Publications.

Campbell, D. T., & Stanley, J. C. (1966). Experimental and quasi-experimental designs for research. Chicago, IL: Rand McNally.

Carey, J. O., & Carey, L. M. (1980). Using Formative Evaluation for the Selection of Instructional Materials. Journal of Instructional Development, 3(3), 12-18.

Cartwright, G. P. (1984). Computer Applications in Special Education, in Walker, D. F., & Hess, R. D. (1984). Instructional Software: Principles and perspectives For Design and Use. Belmont, CA: Wadsworth Publishing Company.

Coopersmith, S. (1967). The Antecedents of Self-Esteem. San Francisco, CA: Freeman.

Cronbach, L. J. (1982). Designing Evaluations of Educational and Social Programs. San Francisco, CA: Jossey-Bass.

Council of Ministers of Education, (1985). Software Evaluation: Criteria For Educational Computer Software Evaluation. Canada: Ministers of Education, Canada.

Dempsey, J. V., & Wager, S. V. (1988). Taxonomy for the Timing of Feedback in computer-Based Instruction. Educational Technology, 28(10), 20-24.

Dick, W. (1977). Formative Evaluation. In Briggs, L. J. (Ed.)., (1977) Instructional Design: Principles and Applications. Englewoods Cliffs, NJ: Educational Technology Publications.

Dick, W. (1980). Formative Evaluation in Instructional Development. Journal of Instructional Development, 3(3), 3-6.

Dick, W., & Carey, L. (1985). The Systematic Design of Instruction. (2nd ed.). Glenview, IL: Scott, Foresman & Co.

Dickson, G. W., Senn, J. A., & Chervany, N. L. (1977). Research in Management Information Systems: The Minnesota Experiments. Management Science, 23(9), 913-923.

Edwards, K. J. (1974). Summative Evaluation: Some Basic Consideration. In Borich, G. D. (Ed.). (1974). Evaluating Educational Programs & Products. Englewood Cliffs, NJ: Educational Technology Publications.

Faust, G. W. (1977). Selecting Instructional Strategies: Or Once You've Got An Objective, What Do You Do With It? Journal of Instructional Development, 1(1), 8-22.

Fisher, G. (1984). Identifying Equitable Software. The Computing Teacher, 11(8), 51.

Fitz-Gibbon, C. T., & Morris, L. L. (1975). Theory-based evaluation. Journal of Educational Evaluation, 5(1), 1-4.

Fitz-Gibbon, C. T., & Morris, L. L. (1978). How to Design a Program Evaluation. Beverley Hills, CA: Sage.

Foulds, R. A. (1982). Applications of Microcomputing in the Education of the Physically Disabled Child. Exceptional Children, 49(2), 155-162.

Frantz, G. A., & Wiggins, R. H. (1982). Design case History: Speak and Spell learns to talk. Spectrum, 19(2), 45-49.

Gagne, R. M., & Briggs, L. J. (1977). Principles of Instructional Design. NY: McGraw-Hill.

Gerlach, V. S., & Ely, D. P. (1971). Teaching and Media: A Systematic Approach. Englewood Cliffs, NJ: Prentice-Hall.

Golas, K. C. (21983). The Formative Evaluation of Computer-Assisted Instruction. Educational Technology, 23(1), 26-28.

Goodrich, G. L., Bennett, R. R., Paul, H. S. & Asley, J. K. (1980). Preliminary report on evaluation of synthetic-speech reading machines. Journal of Visual Impairment & Blindness, 78(9), 408-413.

Gropper, G. L. (1974). Instructional Strategies. Englewood Cliffs, NJ: Educational Technology Publications.

Hannaford, A. E., & Taber, F. M. (1982). Microcomputer Software for the Handicapped: Development & Evaluation. Exceptional Children, 49(2), 137-142.

Hatfield, R. C. (1975). Defining an Instructional Strategy. In S. D. Zalatimo and P. J. Sleeman (Ed.). (1975) A Systems Approach to Learning Environments. Roselle, NJ: Meded Projects, Inc.

Higginbottom, P. (1980). Grid Analysis Package. University of Leeds, U.K.

Holladay, D., Navey, C., & Kaysen, J., (1984, October). Street's Voice Synthesizers: The Truth Behind The Rumors, Raised Dot Computing p. 2-3.

Jaegar, M. (1988). Zaps, Booms & Whistles in Educational Software. The Computing Teacher, 14(3), 20-27.

Jay, T. B. (1983). The Cognitive Approach to Computer Courseware design & Evaluation. Educational Technology, 23(1), 22-26.

Just, M. A., & Carpenter, P. A. (1980). A theory of reading: from eye fixations to comprehension. Psychological Review, 87(40), 329-354.

Katz, D. S. & Morgan, R. L. (1974). A Holistic Strategy For The Formative Evaluation of Educational Programs, In Borich, G. D. (Ed.). (1974). Evaluating Educational Programs & Products. Englewood Cliffs, NJ: Educational Technology Publications.

Keller, A. (1987). When Machines Teach. NY: Harper & Row.

Kelly, G. (1955). Psychology of Personal Constructs. NY: Norton.

Kirtley, D. D. (1975). The Psychology of Blindness. Chicago, IL: Nelson-Hall.

Krolick, B. (1984). Computer Access For the Visually Impaired. The Computing Teacher, 11(8), 48-50.

Laurer, H. (1984, October). An Evaluation of the Microbrailler, Raised Dot Computing, p.6.

Lathrop, A., & Goodson, B. (1983). Courseware in the Classroom. CA: Addison-Wesley Publishing Company.

Laurer, H., & Molinski, L. (1984, November). The Quandry: Selcting Computer Aids for the Visually Impaired. Raised Dot Computing, p.1-4.

Laurer, H., & Mowinski, L. (1986). Computer Access by Visually Impaired Persons: Is It Still A Dream? Journal of Visual Impairment & Blindness, 80(8), 910-913.

Leiker, C. A. (1982). Human Factors Considerations Relevant To The Performance of Computer Users. Paper presented to the 20th International Congress on Applied Psychology, University of Edinburgh.

Lewis, R. Harrison, P. Lynch, E. & Saba, F. (1988). Effective Software for Special Education: Results of a statewide survey. Closing The Gap, 6(6), 19-20.

Marrapodi, M. R. (1984). Females and Computers? Absolutely! The Computing Teacher, 11(8), 57-58.

Maryland Computer Services, Incorporated. (1983, July/August). Product Spotlight. p.1.

Maslow, A. H. (1970) Motivation and Personality. NY: Harper & Row.

Merrill, M. D. (1975). Learner Control: Beyond Aptitude Treatment Interactions. Audio-Visual Communication Review, 23, 217-226.

Merrill, M. D. (1985, March). Where Is The Authoring in Authoring System? Ninth annual Peter Dean lecture given to the Association for the Development of Computer-Based Instructional Systems, Washington, D.C.

Michaelis, P. R. & Wiggins, R. H. (1982). A Human Factors Engineer's Introduction To Speech Synthesizers. In Badre, A. & Schneiderman, B. (Eds.) Directions in human computer interaction, (pp. 149-178). Norwood, NJ: Ablex Publishing Corporation.

Mizokawa, D. T. & Levin, J. (1988). Standards for Error Message in Educational Software: An Initial Proposal. Educational Technology, 28(1), 19-23.

Moyer, J. J., & Scadden, L. A. (1987). The Computer Access Field Considers Itself. Journal of Visual Impairment & Blindness, 81(4), 168-172.

Naiman, A. (1987). A Hard Look at Educational Software. Byte, 12, 193-200.

National Council of Teachers of Mathematics. (1984). Guidelines for evaluating computerized instructional materials. (rev. ed.). Reston, VA: author.

The International Council for Computers in Education, (1984). Evaluator's Guide for microcomputer-Based Instructional Packages. Portland, Oregon: Northwest Regional Educational Laboratory's Computer Technology Program.

Ohtsuki Communication Products, Incorporated. (undated). Express Yourself in print and Braille with the Ohtsuki printer. (Adv.) Walnut Creek, CA: author.

Owston, R. D. (1987). Software Evaluation: A Criterion-Based Approach. Ontario, Canada: Prentice-Hall Canada Inc.

Owston, R. D., & Wideman, H. H. (1987). The Value of Supplementary Panel Software Reviews with Field Observations. Canadian Journal of Educational Communication, 16(4), 295-308.

Patterson, A. C., & Blcch, B. (1987). Formative Evaluation: A Process Required in Computer-Assisted Instruction. Educational Technology, 23, 26-30.

Ploch, M. (1986). Computers in Schools: Can They Make The Grade? High Technology, 15(9), 44-69.

Poynor, H. (1972). Detection of curriculum treatment differences: Excluding teacher variance in hierarchical analysis of variance design. Unpublished doctoral dissertation, The University of Texas at Austin.

Preece, J. & Jones, A. (1985). Training teachers to select educational software: Results of a formative evaluation of an Open University Package. British Journal of Educational Technology, 16(1), 9-20.

Rogers, C. (1961). On becoming a person. Boston, MA: Houghton Mifflin.

Romiszowski, A. J. (1981). Designing Instructional Systems: Decision-making in course planning and curriculum design. London: Kogan page.

Rose, R. (1984). Identifying Equitable Software. The Computing Teacher, 11(8), 24-27.

Ruconich, S. (1984). Evaluating Microcomputer Access Technology for Use by Visually Impaired Students. Education of the Visually Handicapped, 15(4), 119-125.

Russell, J. D. & Blake, B. L. (1988). Formative and Summative Evaluation of Instructional Products and Learners. Educational Technology, 28(9), 22-28.

Salomon, G. & Gardner, H. (1986). The Computer as Educator: Lessons From Television Research. Educational Researcher, 15(1), 13-18.

Sanford, L. (1984). A Formative Evaluation of an Instructional Program Designed To Teach Visually Impaired Students To Use Microcomputers. Education of the Visually Handicapped, 15(4), 135-144.

Scheur, B. (1984, Novemeber). Ripping Off The Blind, Raised Dot Computing, p.7.

Schneidermann, B. (1979). Human Factors Experiments in Designing Interactive Systems. Computers, 12, 9-19.

Schneidermann, B. (1980). Software Psychology: Human Factors in Computer and Information Systems. Cambridge, MA: Winthrop Publishers, Inc.

Schreier, E. M., DeWitt, J.C., Goldberg, A.M., & Leventhal, J.D. (1987). An evaluation of synthetic speech software programs. Journal of Visual Impairment, 81(2), 70-75.

Scriven, M. (1967). The Methodology of Evaluation. In Tyler, R. W., Gagne, R. M., & Scriven, M. Perspectives on Curriculum Evaluation. Chicago, IL: Rand McNally.

Signer, B. (1983). How do teacher and student evaluations of CAI compare? The Computing Teacher, 11(2), 34-36.

Smith, P. L., & Boyce, B. A. (1984). Instructional Design Considerations in the Development of Computer-Assisted Instruction. Educational Technology, 25(7), 5-11.

Steinberg, E. R. (1983). Reviewing the Instructional Effectiveness of Computer Courseware. Educational Technology, 23(1), 34-36.

Steffin, S. A. (1983). A Suggested Model For Establishing the Validity of Computer-Assisted Instructional Materials. Educational Technology, 23(1), 17-19.

Sturdivant, P. (1984). Access to Technology: The Equality Paradox. The Computing Teacher, 11(8), 65-67.

Taber, F. M. (1983). Microcomputers in Special Education: Selection and Decision Making Process. Reston, VA: The Council For Exceptional Children.

Telesensory Systems Incorporated. (1986). VERSAPOINT: The Next Generation Personal Braille Embosser. (Adv. No VP17600-9/86). Mountain View, CA: author.

The Trace Research & Development Center for the Severly Communicatively Handicapped. (1982). Computers for the Handicapped: Panacea or Pie in the Sky. Programs for the Handicapped, 3, 4-7.

Truett, C. (1984). Field-Testing Educational Software: Are Publishers Making The Effort? Educational Technology, 25(5), 7-12.

Truett, C., & Ho, C. (1986). Is Educational Software Field-Tested? The Computing Teacher, 14(2), 24-25.

Tuttle, D. W. (1987). The Role of the Special Education Teacher-Counselor in Meeting Students' Self-Esteem needs. Journal of Visual Impairment & Blindness, 81(4), 156-161.

Ullmer, E. J. (1988). On the Form of Human Enterprise: Examining the Design of Design. Educational Technology, 25(9), 14-21.

Vanderheiden, G. (1982). Computers Can Play a Dual Role for Disabled Individuals: Besides providing assistance microcomputers should give disabled individuals access to standard software. Byte, 7 136-146.

VTEK. (1985). Exciting new products for people who are totalluy blind. (Adv. No M-2723/85). Santa Monica, CA: author.

Walker, D. F., & Hess, R. D. (1984). Instructional Software: Principles & Perspectives for Design & Use. CA: Wadsworth Publishing Company.

Weston, C. B. (1986). Formative Evaluation of Instructional Materials: An Overview of Approachers. Canadian Journal of Educational Communication, 15(1), 5-17.

Weston, C. B. (1987). The importance of Involving Experts & Learners in Formative Evaluation. Canadian Journal of Educational Communication, 16(1), 45-58.

Williams, F. & Williams, V. (1985). Success with Educational Software. NY: Praeger.

Young, M. E. (1984). Constraints on Microcomputer Access for Visually Impaired Persons. Journal of Visual Impairment& Blindness, 78(9), 426-427.

APPENDIX I

IDENTIFICATION OF SOFTWARE PROGRAMS

a) used in the evaluation; and
b) excluded from the evaluation

Appendix I

IDENTIFICATION OF SOFTWARE PROGRAMS USED IN THE EVALUATION

A) The American Printing House For The Blind. (1986). Math Quiz. [computer program] KY: Street Electronic Corporation (Echo Speech Generator, Version 3.1.2).

B) The American Printing House For The Blind. (1985). Clock Time. [computer program] KY: Street Electronics Corporation (Echo Speech Generator, Version 3.1).

C) The American Printing House For The Blind. (1985). Clock Time. [computer program] KY: Street Electronics Corporation (Echo Speech Generator, Version 3.1).

D) The American Printing House For The BLind. (1986). Math Tester. [computer program] KY: Street Electronics Corporation (Echo Speech Generator, Version 3.1.2.).

E) Street, Levieux, & Kory. (1985). Math Level 15 [computer program] CA: Street Electronics Corporations (Echo Speech Generator, Version 3.0).

F) Street, Levieux & Kory. (1985). Math Level 12 [computer program] CA: Street Electronics Corporation (Echo Speech Generator, Version 3.0).

G) Mallow, M. & Guitierrez, M. (1986). Math Tutor Program [computer program] NE: The American Printing House For The Blind (Echo Speech Generator, Versaion 3.1.2).

H) Street, Levieux & Kory (1985). Math Drill [computer program] CA: Street Electronics Corporation (Echo Speech Generator, Version 3.1.2).

I) Street, Levieux & Kory. (1985). Word Problems [computer program] CA: Street Electronics Corporation (Echo Speech Generator, Version 3.1, A 009).

J) Schumacher, S. P. & Hearn, C. L. (1986). Musical Math Teacher [computer program] KY: The American Printing House For The Blind (Echo Speech Generator, Version 3.1.2).

K) The American Printing House For The Blind. (1986). Change [computer program] KY: Stree Electronics Corporation (Echo Speech Generator, Version 3.1.2).

L) Street, Levieux & Kory. (1985). Math [computer program] CA: Street Electronics Corporation (Echo Speech Generator, Version 3.1, A 013).

M) Street, Levieux & Kory. (1985). Math Level 13: Subtraction [computer program] CA: Street Electronics Corporation (Echo Speech Generator, Version 3.0).

N) Street, Levieux & Kory. (1985). Math Level 10: Review [computer program] CA: Street Electronics Corporation (Echo Speech Generator, Version 3.0).

O) Street, Levieux & Kory. (1985). Tom's Math Drill [computer program] CA: Street Electronics Corporation (Echo Speech Generator, Version 3.1, A 029).

P) The American Printing House For The Blind. (1986). Math Decathlon [computer program] KY: Street Electronics Corporation (Echo Speech Generator, Version 3.1.2).

Q) The American printing House For The Blind. (1986). Math 1: Addition & Subtraction [computer program] KY: Street Electronics Corporation (Echo Speech Generator, Version 3.1.2).

R) Street, Levieux & Kory. (1985), Math Level 5: Grouping. [computer program] CA: Street Electronics Corporation (Echo Speech Generator, Version 3.0).

S) Street, Levieux & Kory. (1985). Math Level 6: Tens [computer program] CA: Street Electronics Corporation (Echo Speech Generator, Version 3.0).

T) Street, Levieux & Jory. (1985). Math Level 12: Subtraction [computer program] CA: Street Electronics Corporation (Echo Speech Generator, Version 3.0).

U) Street, Levieux & Kory. (1985). Math Level 8: Carrying In Addition [computer program] CA: Street Electronics Corporation (Echo Speech Generator, Version 3.0).

V) Street, Levieux & Kory. (1985). Math Level 9: Subtraction With Borrowing. [computer program] CA: Street Electronics Corporation (Echo Speech Generator, Version 3.0).

W) Street, Levieux & Kory. (1985). Math Level 14: Computation 12C. [computer program] CA: Street Electronics Corporation (Echo Speech generator, Version 3.0).

X) The American Printing House For The Blind. (1986). Math Drill Hard. [computer program] KY: Street Electronics Corporation (Echo Speech Generator, Version 3.2).

Y) Street, Levieux & Kory. (1985). Math Level 11: Computation 11c. [computer program] CA: Street Electronics Corporation (Echo Speech Generator, Version 3.0).

PROGRAMS WHICH WERE EXCLUDED

1. Street, Levieux & Kory. (1985). ZAP. [computer program] CA: Street Electronics Corporation (Echo Speech Generator, Version 3.1, a 017).
* Program displays the word "ZAP" on the screen and fails if user makes an error.

2. Street, Levieux & Kory. (1985). Math Level 4. [computer program] CA: Street Electronics Corporation (Echo Speech Generator, Version 3.0, A 015).
* Program reads "Unidentifiable Error in 300"at the beginning of the program.

3. Unknown. (1981). Lines [computer program] MINN: MECC Distribution (Elementary Volume 9, Geometry).
* Program is based on graphic illustrations only.

4. Unknown. (1981), Triangles. [computer program] MINN: MECC Distribution (Elementary Volume 9, Geometry).
* Program is based on graphic illustrations only.

5. The American Printing House For The Blind. (1985). Math Level XVIII: Math Facts. [computer program] KY: Street Electronics Corporation (Echo Speech Generator, Version 3.1).
* Program does not read the questions it only reads the instructions.

6. Street, Levieux & Kory. (1986). Getting Good At Subtraction. [computer program] CA: Street Electronics Corporation (Echo Speech Generator, Version 3.0).
* Program's instructions are based on visual illustrations only.

7. Street, Levieux & Kory. (1985). Math Level 2: Find the Missing Number. [computer program] CA: Street Electronics Corporation (Echo Speech Generator, Version 3.0).
* Questions are based on visual illustrations only.

APPENDIX II

Apple II - Plus output of
the three cluster analyses
(as per Higginbottom, 1980)

N.B. Program attributes are referred to
in the computer output as 'constructs'

a.) Cluster Analysis: Program Operation & Technical Quality

TOP-RIGHT IS CORRELATION MATRIX
DIAGONAL IS SUMMED RELATIONSHIPS
BOTTOM-LEFT IS INDIV. RELATIONSHIPS
2-TAIL P: *=5%, **=1%. NN=DF+1

CONSTRUCT

	R1	R2	R3	R4	R5	R6	R7	R8
R1	613.71	0	-.66	-.88	-.88	-.71	-1**	.19
R2	0	676.64	-.92	-.97*	-.85	-.85	-1**	-.93*
R3	43.19	85.17	968.17	-1**	-1**	-1**	-1**	-.96*
R4	77.88	93.34	99.77	776.94	-.99**	-.97*	-1**	-.17
R5	77.08	71.73	99.5	98.81	956.2	-1**	-.87	-.76
R6	50.92	71.73	99.99	93.34	95.12	711.77	-.96*	-.5
R7	99.96	100	100	99.7	75.06	91.92	1066.13	-1**
R8	3.47	96.79	91.53	2.85	57.97	24.77	99.9	472.43
R9	78.53	91.92	75.06	96.79	98.89	97.24	100	37.64
R10	99.56	29.89	68.1	21.13	93.34	24.77	99.9	24.77
R11	4.44	6.16	97.35	93.34	98.89	0	99.99	29.89
R12	77.88	29.89	88.1	0	84.62	57.97	99.7	2.85

CONSTRUCT

	R9	R10	R11	R12
R1	-.89	1**	-.21	.88
R2	-.96*	-.55	-.25	-.55
R3	-.87	-.94	-.99*	-.94
R4	-.98*	-.46	-.97*	0
R5	-.99**	-.97*	-.99**	-.92
R6	-.99*	-.5	0	-.76
R7	-1**	-1**	-1**	-1**
R8	-.61	.5	-.55	-.17
R9	801.08	-.61	-.71	-.98*
R10	37.64	619.04	0	1**
R11	50.57	0	484.1	-.19
R12	96.79	99.94	3.47	641.22

CONSTRUCTS IN ORDER OF
CONTRIBUTION TO VARIANCE:
7 3 5 9 4 6 2 12 10 1 11 8

COMPONENT 1 - PRINCIPAL CONSTRUCT IS NO.7
INCLUDED IN ORDER OF IMPORTANCE
ARE CONSTRUCTS:3 9 4 6 2 12 10 1 11 8

COMPONENT 2 - PRINCIPAL CONSTRUCT IS NO.5
INCLUDED IN ORDER OF IMPORTANCE
ARE CONSTRUCTS:3 9 4 6 10 11

b) Cluster Analysis: Content & Instructional Quality

TOP-RIGHT IS CORRELATION MATRIX
DIAGONAL IS SUMMED RELATIONSHIPS
BOTTOM-LEFT IS INDIV. RELATIONSHIPS
2-TAIL P: *=5%. **=1%. NN=DF 1

CONSTRUCT

	R1	R2	R3	R4	R5	R6	R7	R8
R1	442.95	-.61	-.52	-.92	-.71	-.71	-1**	-.97*
R2	37.64	425.38	-1**	.98*	.92	0	.76	.71
R3	27.09	99.12	634.36	-1**	-.96*	-.96*	-.99*	-1**
R4	84.62	95.11	99.9	411.88	0	0	.99**	.19
R5	50.57	84.62	91.92	0	234.41	0	-.21	-.17
R6	50.57	0	91.92	0	0	200.36	.21	-.17
R7	99.12	57.97	97.35	98.89	4.44	4.44	399.85	0
R8	93.34	50.92	99.96	3.47	2.85	2.85	0	331.27
R9	0	0	27.09	29.89	0	50.57	37.64	77.88

CONSTRUCT

	R9
R1	0
R2	0
R3	-.52
R4	-.55
R5	0
R6	-.71
R7	-.61
R8	-.88
R9	223.08

CONSTRUCTS IN ORDER OF
CONTRIBUTION TO VARIANCE:
3 1 2 4 7 8 5 9 6

COMPONENT 1 - PRINCIPAL CONSTRUCT IS NO.3
INCLUDED IN ORDER OF IMPORTANCE
ARE CONSTRUCTS:2 4 7 8 5 6

COMPONENT 2 - PRINCIPAL CONSTRUCT IS NO.1
INCLUDED IN ORDER OF IMPORTANCE
ARE CONSTRUCTS:7 8

COMPONENT 3 - PRINCIPAL CONSTRUCT IS NO.9

c) **Cluster Analysis: Motivation**

TOP-RIGHT IS CORRELATION MATRIX
DIAGONAL IS SUMMED RELATIONSHIPS
BOTTOM-LEFT IS INDIV. RELATIONSHIPS
2-TAIL P: *=5%, **=1%. NN=DF<1

CONSTRUCT

	R1	R2	R3	R4	R5	R6
R1	349.09	.99**	1**	0	-.89	.85
R2	98.89	344.37	1**	-.19	-.96*	.71
R3	99.94	99.56	351.85	.76	-.96*	.17
R4	0	3.47	57.97	96.26	-.32	-.5
R5	78.53	91.53	91.53	10.04	370.43	-.99**
R6	71.73	50.92	2.85	24.77	98.81	249.08

CONSTRUCTS IN ORDER OF
CONTRIBUTION TO VARIANCE:
5 3 1 2 6 4

COMPONENT 1 - PRINCIPAL CONSTRUCT IS NO.5
INCLUDED IN ORDER OF IMPORTANCE
ARE CONSTRUCTS:3 2 6

COMPONENT 2 - PRINCIPAL CONSTRUCT IS NO.1
INCLUDED IN ORDER OF IMPORTANCE
ARE CONSTRUCTS:3 2

COMPONENT 3 - PRINCIPAL CONSTRUCT IS NO.4
NO RELATED CONSTRUCTS

APPENDIX III

SAMPLE OF PRETEST AND POSTTEST

(The number inserted into the boxes following each item is the number of subjects who responded with the given rating.)

NUMBER: ________________

SESSION: ________________

PRETEST (1988)

YOUR ANSWERS TO THESE QUESTIONS WILL GREATLY HELP IN MY STUDY OF THE COMPUTER PROGRAM YOU WILL BE WORKING WITH.

ALL THE INFORMATION YOU GIVE WILL BE ANONYMOUS.

ANY QUESTIONS BEFORE WE BEGIN?

SECTION A: PLEASE TELL ME THE ANSWER TO THE FOLLOWING QUESTIONS

1. 28 - 3 =

2. 55 - 41 =

3. 47 + 35 =

4. 13 + 26 + 49 =

5. 21 + 44 =

6. 19 - 10 =

7. 23 + 65 =

8. 41 - 30 =

9. 28 - 6 =

10. 28 - 17 =

11. 46 + 10 =

12. 19 + 6 =

13. 35 - 21 =

14. 38 + 12 =

15. 26 + 70 =

16. 29 - 15 =

17. 19 + 40 =

18. 28 - 13 =

19. 37 - 16 =

20. 27 - 22 =

THANK YOU FOR YOUR HELP!

NUMBER: ____________________

SESSION: ____________________

POST-TEST (1988)

TO COMPLETE THIS STUDY, I WOULD BE GRATEFUL FOR YOUR ANONYMOUS RESPONSES TO THESE QUESTIONS. ANY QUESTIONS BEFORE WE BEGIN?

SECTION A: PLEASE TELL ME WHICH STATEMENT BEST DESCRIBES YOUR ANSWER

1. PLEASE GIVE ME YOUR OVERALL OPINION OF THE MATH COMPUTER PROGRAM YOU HAVE JUST USED.

2 ☐ IT WAS VERY GOOD; VERY USEFUL

3 ☐ IT WAS GOOD; IT WAS USEFUL

☐ IT WAS NOT VERY GOOD; I DIDN'T FIND IT VERY USEFUL

1 ☐ IT WAS NO GOOD; NOT USEFUL AT ALL.

SECTION B: PLEASE ANSWER EACH QUESTION.

2. 27 - 22 =

3. 37 - 16 =

4. 28 - 13 =

5. 19 + 40 =

6. $29 - 15 =$

7. $26 + 70 =$

8. $38 + 12 =$

9. $35 - 21 =$

10. $19 + 6 =$

11. $46 + 10 =$

12. $28 - 17 =$

13. $28 - 6 =$

14. $41 - 30 =$

15. $23 + 65 =$

16. $19 - 10 =$

17. $21 + 44 =$

18. $13 + 26 + 49 =$

19. $47 + 35 =$

20. $55 - 41 =$

21. $28 - 3 =$

SECTION C: PLEASE TELL ME WHICH STATEMENT BEST DESCRIBES YOUR ANSWER.

22. Do you like to be informed of what you will be learning in a computer lesson?

5☐ like it   1☐dislike it   ☐ don't care

23. Do you dislike working on the computer by yourself?

4☐ like it   1☐dislike it   1☐don't care

24. Do you like it when math computer programs provide you with instructions?

5☐ like it   1☐dislike it   ☐ don't care

25. Do you dislike it when math computer programs call you by name?

5☐ like it   ☐ dislike it   1☐don't care

26. Do you like to type in an abbreviation (such as y for yes) instead of typing in the whole word?

5☐ like it   ☐ dislike it   1☐don't care

27. Do you dislike it when a math computer program repeats a question?

5☐ like it   ☐ dislike it   1☐don't care

28. Do you like math computer programs to praise you when you get the right answer?

5☐ like it   ☐ dislike it   1☐don't care

29. Do you dislike it if a math computer program tells you if you make a mistake?

4☐ like it 1☐ dislike it 1☐ don't care

SECTION D: PLEASE TELL ME WHICH STATEMENT BEST DESCRIBES YOUR ANSWER.

30. Do you find math computer programs ask difficult questions?

2☐ yes 1☐ no 3☐ about right

31. Do you find math computer programs give you enough time to answer the questions?

1☐ yes 5☐ no ☐ about right

32. Do you think computer programs are able to help you in math?

3☐ yes 1☐ no 2☐ about right

33. Do math computer programs let you have enough tries per answer?

1☐ yes 3☐ no 2☐ about right

34. Do you find math computer programs interesting to use?

5☐ yes ☐ no 1☐ about right

SECTION E: PLEASE TELL ME THE ANSWER TO EACH QUESTION.

35. What sex are you?

4☐ male 2☐ female

36. How old are you?

3☐ 7-9yrs. 3☐ 10-12 yrs. ☐ 13-15yrs.

37. What grade are you in?

1☐ 4th 2☐ 5th 3☐ 6th

38. What is your visual acuity?

____

39. If you have a visual impairment, what caused your loss of vision?

2☐ birth loss 4☐ natural loss ☐ accidental loss

40. Do you have a computer at home?

1☐ yes 5☐ no

If yes, brand name?

____

41. Do you use a computer at school?

6☐ yes ☐ no

If yes, about how many hours per week do you spend using a computer?

1☐ <1 hr. 2☐ 1-2 hrs. ☐ 2-3 hrs. 1☐ 3-4 hrs. 3☐ >4 hrs.

42. Do you take math as a subject?

6☐ yes ☐ no

43. Have you ever used a math computer program?

6☐ yes ☐ no

If yes, what did you think of the program?

____

____

SECTION F:PLEASE TELL ME WHICH STATEMENT BEST DESCRIBES YOUR ANSWER.

44. Would you be interested in using computer programs for all your subjects?

4☐ yes 2☐ no ☐ maybe

45. How much do you enjoy math?

☐ quite alot 2☐ alot 4☐ not very much ☐ not at all

46. Is getting better at math important to you?

4☐ very important 2☐ quite important ☐ not very important ☐ unimportant

47. How good do you think you are at math?

☐ very good 3☐ good 3☐ not very good ☐ poor

APPENDIX IV

INSTRUCTIONS TO THE EXPERIMENTER

APPENDIX IV
INSTRUCTIONS TO THE EXPERIMENTER

'Treatment fidelity' refers to the extent to which the testing procedures are implemented according to the investigator's specifications (Borg & Gall, 1983). In order to prevent experimenter bias in this respect the investigator wrote precise instructions for the experimenter to follow. As a further precaution, the investigator placed reminders within the directions for the experimenter to follow each step precisely as indicated and not to change any procedure. The full instructions to the experiments are given

1. The subject should be alone with you in the room, all materials and equipment should be readily available.

2. Write down the subjects assigned number on the pretest in order for the author to be able to identify the subject. Write down the date of the session.

3. Read the instructions written on the pretest to the subject: "Your answers to these questions will greatly help me in my study of the computer program you will work with. All the information you give will be anonymous. Any questions before we begin?"

4. Please do not provide any details about the test items to the subject. As well, please try to keep your tone of voice constant when reading the items so that the subject can not infer any preference for an answer.

5. Read the following instruction written on the pretest before asking the test items: "Please tell me the answers to the following questions."

   Please note: The subjects are allowed to use any method they want to compute the questions, except use of a calculator or abacus. Please do not provide them with any assistance.

6. Begin reading the test items, permitting the subject as much time as possible to answer each question. Once the subject responds, record the answer directly on the pretest beside the question. You can repeat a question as often as necessary. It is very important that you do not change the order of the questions. Please follow this format until the last question of the pretest.

7. Load up the assigned math software program. Begin by inserting the diskette into the diskdrive. Turn on the computer by pressing the switch in the back of the computer. To turn on the monitor, press the switch either found at the bottom of the screen or on the side. The red light on the diskdrive indicates that the program is being loaded. At this point, have the subject sit in front of the computer where he/she can have access to the keyboard. Inform the subject that it will be the software program which will provide the rest of the instruction. Please do not sit close to the subject while the program is in operation. Even though the instructions for each program are different, you do not need to supply the students with any additional instructions since all three programs can be operated independently by the subjects. If the students are experiencing difficulty when using a program, please do not interfere. Rather tell the subjects that you are unable to help and that they must figure out the problem on their own.

8. Once the subject has followed the instruction until the duration of the program, turn off the computer by pressing the switch at the back of the computer. You may also turn off the monitor by pressing the switch at the bottom or side.

9. Write the subject's assigned number and date of the session on the posttest. Read the following instructions written on the posttest to the subject: "To complete this study, I would be grateful for your anonymous responses to the following questions."

10. If there are questions, it must be emphasized again that you must not provide any information about the posttest's items. Begin by reading the first question at a very slow rate. You also need to read the four possible responses, please pause slowly after each one to avoid confusion. Once a subject has chosen an answer, place a checkmark in the appropriate box next to the answer on the posttest.

11. For Section B of the posttest, please follow the exact steps as administered in the pretest. Please ensure that you do not change the order of the questions.

12. For Section C, the instructions written need to be repeated: "Please tell me which statement best describes your answer." Read the question and proceed to read the three possible responses. Wait for the subject to respond and record the answer by placing a checkmark in the appropriate box.

13. For Section D, the same procedures are followed.

14. Section E is gathering factual information. Please read the instructions written to the subject: "Please tell me the answer to each question." It is not necessary to read the possible answers to the subject. Rather record the answer the subject provides you with by placing a checkmark in the appropriate box. Question 42 comprises of two questions. The first question's response is either a "yes" or "no", in which you record the subject's answer be checking the appropriate box. The second question requires that the subject provide a statement. Please write down exactly what the subjects says on the lines provided.

15. In Section F, you need to repeat the instructions which are written: "Please tell me which statement best describes your answer." Apart from reading each item, you also need to read the possible answers each item contains. Record the subject's response by placing a checkmark in the appropriate box.

16. Please thank the subject on my behalf for his/her time and assistance. Gather the testing materials and escort the subject to his/her regular classroom.

I wish to take this opportunity to thank you, the experimenter, for all your valuable assistance.