Analyzing Instructional Software Using a Computer-Tracking System

Robert J. Mills

The purpose of this research was to use computer-tracking system data to empirically determine if a relationship existed between effectiveness (learning) and efficiency (time) in using the program and the way the learner navigated the system's rules, examples, and practice areas. Four simple regressions were conducted. Three statistically significant relationships were found between success (learning) and the way the learner navigated the system's rules, example, and practice areas. No significant relationship between success (learning) and efficiency (time in the instructional program) was found. This study provides support for using a computer-tracking system to improve the process of evaluating instructional software programs.

The process of evaluating instructional software is a necessary component of the instructional design process and widely accepted as an important factor in the development of computer-based training. "Without evaluation, neither developers, nor the organization sponsoring training, know with any certainty or in any detail how well the training is functioning" (Gibbons, 1995, p. 13-1). Evaluation possesses special features (distinguished from traditional research) with the investigation focused on supporting and making sound value judgments (Scriven, 1980). A properly conducted evaluation can help to determine the effectiveness, efficiency, usability, and acceptability of an instructional program (Tessmer, 1993).

Advancements in technology have provided new and unique assessment opportunities for evaluating computer-based instruction (Barab, Fajen, Kulikowich, & Young, 1996). Unfortunately, there is little empirical research on design, development, and testing of unique assessment tools (such as a computer-tracking system) that analyze the data collected according to a specific list of instructional criteria. Therefore, it is difficult to fully understand the potential of using a unique assessment tool, such as a computer-tracking system, in a software evaluation. As the amount of technology used for instructional purposes increases, the need to discover more effective techniques for evaluating instructional software becomes increasingly important. Research on these unique assessment tools may not only lead to improved software evaluation but also serve as a foundation for dynamically monitoring and providing learning guidance to students.

The concept of using a computer-tracking system to help evaluate instructional software has been described as an unobtrusive tool for evaluating computer-based software (Gay & Mazur, 1993; Gibbs, 1995; Misanchuk & Schiwier, 1991; Misanchuk & Schiwier, 1992). According to Guba and Lincoln, unobtrusive measurement is the process of "observing, recording, and analyzing human behavior, or behavior patterns, without the knowledge or awareness of those being observed" (Guba and Lincoln, 1981, p. 263).

A computer-tracking system is a time-stamped record that gathers information such as navigational paths and selections made by the learner (Barab, Fagen, Kulikowich, & Young, 1996). The addition of a computer-tracking system allows evaluators who design applications to collect data regarding the navigational choices and decisions made by the learners. The evaluator can examine the computer-tracking file to determine exactly how a specific learner used the software program. The computer-tracking file is also referred to as a dribble file because of the systematic method by which data are collected and stored (Barab, Fagen, Kulikowich, & Young, 1996). Evaluators can then use dribble file data to improve the usability and effectiveness of the software program.

Robert J. Mills is Assistant Professor of Business Information Systems and Education, College of Business, Utah State University, Logan, Utah.

Review of Software Evaluation

Computers have provided the opportunity to create interactive, multimedia applications that can be used for teaching in both corporate and educational settings. However, evaluating these applications is often an arduous task. Time and budget constraints often prohibit corporate and educational organizations from conducting adequate software evaluations. As a result, computer-based training is often delivered without being properly tested and may not be meeting its instructional objectives.

Evaluation can be defined as judging the worth or merit of something (Scriven, 1967). Scriven is responsible for distinguishing between the formative and summative roles of evaluation. Formative evaluation is conducted to provide information useful in improving an instructional product or design. Summative evaluation is conducted to provide decision-makers or potential customers with judgments about the worth or merit of a program in relation to important criteria.

The evaluation questions asked are different between formative and summative evaluation. Formative evaluations tend to be diagnostic questions such as the following: what is working?; what needs to be improved?; and how can it be improved? On the other hand, summative evaluations tend to ask more judgmental questions: what results occur?; with whom?; with what training?; and at what cost? For instance, formative evaluation is generally looking to improve a particular piece of instruction while summative evaluation is looking to prove or determine the worth.

Tessmer (1993) argues that over the last 30 years, many empirical studies have shown that formatively evaluating instructional materials has significantly increased overall student performance in a number of areas. By finding and eliminating deficiencies in the instruction, improving the ease of use of the end product and interface, and building on the strengths of the product, instruction will clearly be improved. As a result, stakeholders (i.e., designers, managers, and learners) can be more confident that the time and resources that are devoted to the creation of the instruction will be worth the investment and worth the time conducting the evaluation.

Ralph Tyler conceptualized the most popular evaluation approach used today in 1932. This evaluation approach is known as the objectives-based approach to evaluation. According to Worthen and Sanders (1987), Tyler's early approach to evaluation was "logical, scientifically acceptable, and readily usable by educational evaluators" (p. 63). This approach consists of seven basic steps that apply to both formative an summative evaluation approaches:

1. Establish broad goals or objectives.
2. Classify the goals or objectives.
3. Define the objectives in behavioral terms.
4. Find situations in which achievement of objectives can be shown.
5. Develop or select measurement techniques.
6. Collect student performance data.
7. Compare data with behaviorally stated objectives.

According to Guba and Lincoln (1981), there are a few problems associated with Tyler's objectives-oriented approach. For instance, one problem is the goals of the evaluation often become more important than the performance itself. Also, Worthen and Sanders (1987) caution that objective-oriented evaluation can limit the scope and perception of the evaluation similar to blinders, causing the evaluator to miss important outcomes not directly related to the goals of the evaluation.

An evaluation approach developed by Michael Scriven in 1972 to remove the blinders of the objectives-oriented approach is a practice called goal-free evaluation. In a goal-free evaluation, the evaluator enters the evaluation with the purpose of "finding out what the program actually is doing without detailed cueing as to what it is trying to do" (Scriven, 1980, p. 60). Scriven came to the conclusion that part of the evaluation process should be goal-free, and should examine actual effects against a profile of needs in education (Guba & Lincoln, 1981).

In order to conduct a goal-free evaluation, two items of information are necessary: an assessment of actual effects and a profile of needs against which the importance might be assessed (Guba & Lincoln, 1981). It is important to note that an objectives-oriented evaluation approach and goal-free evaluation approach are not mutually exclusive (Worthen, Sanders, & Fitzpatrick, 1997). In fact, goal-directed evaluation and goal-free evaluations can work very well together (Worthen & Sanders, 1987).

According to Martin Tessmer (1993), the obvious overall goal of evaluation is learning effectiveness. The goal of an evaluation is fourfold: identify deficiencies in learning effectiveness, locate ease of use problems, evaluate the efficiency of the instruction, and analyze instructional strengths. Tessmer suggests the most important goal of conducting a formative software evaluation is to improve the effectiveness of the instruction.

A number of common methods for collecting information for both formative and summative evaluation are available. Common evaluation techniques include testing, questionnaires, interviews, focus groups, observations, and existing documents.

One current trend in evaluation theory is the concept of using unobtrusive data collection techniques to obtain evaluation data. According to Guba and Lincoln (1981), unobtrusive measurement is the process of “observing, recording, and analyzing human behavior, or behavior patterns, without the knowledge or awareness of those being observed” (p 263). Using unobtrusive data collection methods helps to overcome the “unfortunate side effect of many information-collection methods where subjects alter their behavior when they know information is being collected from them” (Worthen, Sanders, & Fitzpatrick, 1997, p 385). Unobtrusive data collection measures also help the investigator to collect information without any unnecessary disruption to the learner or to the project.

Review of Computer-Tracking Systems

Although computer-tracking devices have a number of uses, there is little documentation on using computer-tracking systems for software evaluation. As a result, the requirements for an article to be included in this review are broad. Many of the articles reviewed were theoretical papers examining the possibility of using a computer-tracking system as an unobtrusive data-collection technique (Gay & Mazur, 1993; Gibbs, 1995; Misanchuk & Schwier, 1991; Misanchuk & Schwier, 1992; Schwier & Misanchuk, 1990). Misanchuk and Schwier have published three papers examining various methods for collecting data in non-linear hypertext environments. These methods varied in complexity and structure for recording the navigation of a learner in a computerized environment. Other authors have investigated using video and audio capture as dribble file tools (Gay & Mazur, 1993), as well as using a computer-tracking system to analyze and improve hypermedia programs (Barab, Fajen, Kulikowich, & Young, 1996; Lawless & Kulikowich, 1996).

Very few documented studies in which authors have incorporated the use of a computer-tracking system during the evaluation process were located. For example, Trumbull, Gay, and Mazur (1992) built a computer-tracking system to evaluate a program called “Bughouse.” Similarly, the Cities in Schools Hypermedia Training Program also used a computer-tracking system to help evaluate the effectiveness of the training program (Harvey & Nelson, 1995).

This research indicated there is a large range of complexity with a dribble file. A computer-tracking system can be created in an authoring language such as Authorware Professional, or HyperCard (Schwier & Misanchuk, 1990). Simple dribble files can be added to the existing program by only requiring a few lines of additional code. More complicated computer-tracking systems have also been created in advanced programming languages such as C (Trumbull, Gay, & Mazur, 1992). If programming is not possible, Gibbs (1995) suggested the possible use of video and audio recording in place of a computer-tracking system.

Regardless of how the computer-tracking system is created, Harvey and Nelson (1995) recommended the computer-tracking system be designed and programmed into the hypermedia application as the instructional program is being developed. This strategy of developing both the software and dribble file together helps to ease the development and effectiveness of the evaluation.

The type and amount of information collected by a dribble file is basically unlimited. Much of the research indicated the possibility of tracking every decision made by the user (Misanchuk & Schwier, 1991; Schwier & Misanchuk, 1990). Tracking allows the evaluator to collect valuable information such as time, interactions, key presses, mouse-clicks, and a summary of the usage (Harvey & Nelson, 1995).

Other dribble files allow for the collection of additional information provided by the learner. For example, Gay and Mazur (1993) provided the learner with the opportunity to make both audio and text comments throughout the evaluation. This information was then placed in the dribble field at the appropriate location for the evaluator to review.

The use of a computer-tracking system provides a number of unique data-collection possibilities that are not available through other collection techniques. A variety of advantages for using a computer-tracking system have been described in the literature.

Dribble files provide an excellent description of actual usage made by the learner (Misanchuk & Schwier, 1991). Examining this actual usage allows for the detection of errors in the program and parts of the program, which receive less traffic than other areas. These problem areas can then be investigated in further detail and corrected.

Another advantage of using a computer-tracking system is the unobtrusive nature of the data collection. Selecting a data collection technique that will not cause significant disruption to the project is an important part of conducting an evaluation (Worthen & Sanders, 1987; Worthen, Sanders, & Fitzpatrick, 1997). This method does not distort the delivery of training and emulates a real-world experience rather than a controlled lab situation (Harvey & Nelson, 1995). As a result, evaluators are able to track the experiences of a user in a non-linear environment without disrupting the learner's navigation through the program.

Some serious limitations to the design, development, and use of a computer-tracking system in evaluating instructional software were found in the literature. Computer-tracking systems often require considerable programming experience. The complexity of creating a computer-tracking system can be time consuming and require valuable resources (Gay & Mazur, 1993). When these valuable resources are consumed by the development of a computer-tracking system, the actual program is often left short of the necessary funds to create the product (Harvey & Nelson, 1995).

Finally, a computer-tracking system often collects too much information to be useful. This raw data can be overwhelming in amount and difficult to analyze (Schwier & Misanchuk, 1990). It is often necessary to incorporate other evaluation techniques to help create a complete picture with the large amounts of information collected by the computer-tracking system (Gibbs, 1995).

Problem Statement

The literature suggests a computer-tracking system has a great deal of potential as an unobtrusive data-collection technique for evaluating instructional software. This approach appears to help provide a more comprehensive picture about what the learner is experiencing than interviews and questionnaires alone. Scriven (1980) suggested that triangulation is a technique to get a fix on a phenomenon by approaching it from more than one independently based route. Computer-tracking systems allow the evaluator to triangulate data creating a more comprehensive evaluation. The computer-tracking system provides an excellent mechanism for describing the actual use of the student maneuvering through the instructional software.

From an initial examination of computer-tracking system data from a program designed to teach relational database concepts, it was observed that many of the top posttest-scoring learners took time to review the relational database concept rules just before exiting the instructional program. Second, they appeared to examine all three primary presentation forms—rule, example, and practice—while navigating the software. Third, they also correctly answered practice items. Fourth, they did not take any more time than other learners.

Therefore, the purpose of this study was to use the computer-tracking system data to empirically determine if a relationship existed between effectiveness (learning) and efficiency (time) in using the program and the way the learner navigated the system's rules, examples, and practice areas. Such data may be more descriptive and may provide more concrete evidence than interviews with learners in evaluating software or used with other data collection methods to better evaluate instructional software. Specifically, four hypotheses were tested:

1. No statistically significant relationship exists between a student's use of rule reviews just prior to exiting the instructional program and his or her posttest score.
2. No statistically significant relationship exists between a student's use of all primary presentation forms (rule, example, practice) and posttest score.
3. No statistically significant relationship exists between the number of practice items correctly answered and the student's posttest score.
4. No statistically significant relationship exists between a student's time in the program and his or her posttest score.

Method

Participants

The participants of this study consisted of 94 freshman and sophomore level college students at a large Western university. The participants selected were enrolled in an introductory computer class and had a basic understanding of databases. This study took place late in the semester after students were taught basic computer skills and basic database concepts. However, none of the participants claimed prior experience with relational database functions. Participants were informed their participation was voluntary and they could withdraw at any time. Participants were also informed the study would not have any affect on grades. Each participant signed an informed-consent form.

Materials

The instruments of this study included the following:

1. Sixteen-item multiple-choice assessment measuring students' knowledge of relational database concepts taught in the instructional software program. This item served as the posttest. The test was pilot tested with a small sample of students and instructional designers before the actual data collection began. This exam was designed to be difficult in order to avoid a possible ceiling effect. An expert in database systems examined the test to help ensure content validity by certifying the exam questions were accurate and relevant to the instruction. A KR-21 was also conducted to check the reliability of scores obtained from the instrument. Responses to the achievement test were assigned a score of 1 for a correct answer and a 0 for an incorrect answer. According to Pedhazur and Schmelkin (1991), "multiple-choice items on measures of achievement are prime examples of dichotomously scored items" (p. 97). The posttest reliability, calculated using KR-21 on the entire sample in this study, was .7007.
2. A computer software program teaching relational database functions was designed using the Component Display Theory of Instruction (Merrill, 1994). Specifically, the instruction was a concept lesson at the Use level according to the Component Display Theory Matrix (Merrill, 1994). The software was pilot tested with a small sample of students before actual data collection began. An expert in Component Display also examined the software to ensure the software followed CDT prescriptions. An expert in information systems also analyzed the software for content accuracy in the area of relational database functions. Finally, an expert in interface design examined the software program to ensure the interface was properly designed.
3. A computer-tracking system was developed to store information about participants' primary presentation form usage (time in each presentation area, number of examples, and number of correct practice items).

Instructional Task

The course was designed to allow the learner to navigate through each lesson in a way to best meet his or her learning needs. The learner was not forced to navigate to, or participate in, any specific area of the program. The purpose of the software was to teach four relational database function concepts. Four concept lessons were developed to teach the union, intersection, projection, and difference relations. Each lesson included three major areas: presentation (rule review), examples/non-examples, and practice.

The presentation area of instruction provided the learner with a definition of the relational database function being examined (union, intersection, projection, and difference). The definition included the concept name and a list of attributes that distinguished from the other relational database relation concepts. The learner had the opportunity to learn about each of the attributes in greater detail. Also, a "reference example" was included to visually illustrate the definition being described.

The example/non-example area of instruction provided the learner with a set of matched example/non-examples pairs of the relational database function concept being examined. Each of the relational database function concepts contained between six and eight examples (depending on the number of critical attributes) that ranged in difficulty from easy to hard. In addition, attribute isolation was achieved through the use of attention-focusing devices such as color, arrows, and other graphical symbols (Merrill, 1994).

Finally, the practice area of instruction provided practice items of newly encountered examples and non-examples arranged in random sequence. The participants engaged in this activity by answering "true" or "false" as to whether the item being examined was an example of the relational database function being studied. Descriptive correct-answer feedback was provided for incorrectly answered practice items.

Procedure

Participants completed all tasks within a one-hour block of time. The participants in this study were assigned the task of using the software program to learn more about relational database functions. They were notified that a test on the database concepts taught in the software program would be given upon completion of the instructional software program. Safeguards were in place to ensure that students worked independently. Two exam monitors were continually observing the participants as they worked through the instructional software and posttest examination. A basic introduction to the study was presented in a consistent manner to each of the participants. This introduction was programmed into the software program.

Upon completion of the instructional software program, participants were given the 16-item posttest to measure their understanding of the relational database function concepts taught in the instructional software program.

Results

Tests of Assumptions for Analyses

To check assumptions for analysis, tests for homoscedasticity and normality were conducted for each of the regression analyses. Results indicated that data contained equal variance across the range of residuals; there was not a vast departure of normality.

Regression Analyses Results

Three tables summarize statistics produced in the simple-regression analysis. Table 1 summarizes descriptive statistics including minimums, maximums, means, and standard deviations used in the regression equation.

Table 1: Minimums, Maximums, Means, and Standard Deviations of Posttest, Use of Rule Reviews, Use of All Primary Presentation Forms, Number of Practice Items Correctly Answered, and Time in Program

N=94	Minimum	Maximum	Mean	Std. Deviation
Posttest	19.00	100.00	67.994	18.832
Use of rule reviews	.00	1.0	.2872	.4549
Use of all primary presentation forms	.00	1.0	.489	.503
Number of practice items correctly answered	.00	52	18.128	8.686
Time in the program	5 min 23 s	33 min 9 s	17 min 57s	4 min 12 s

Table 2 describes the variables combined in the model selected for presentation (R^2, adjusted R^2, F value, and statistical significance). In Table 3, unstandardized beta weights and standardized beta weights are presented.

Results for each hypothesis addressed in this study are presented below:

Hypothesis #1 - No statistically significant relationship exists between a student's use of rule reviews just prior to exiting the instructional program and his or her posttest score.

Table 2: Summary of Regression with Use of Rule Reviews, Use of All Primary Presentation Forms, Number of Practice Items Correctly Answered, and Time in Program as the Predictor Variable and Posttest as the Dependent Variable

Variable	R Square	Adjusted R Square	F	P
Use of rule reviews	.116	.106	12.066	.001*
Use of all primary presentation forms	.042	.031	3.992	.049*
Number of practice items correctly answered	.074	.064	7.336	.008*
Time in program	.007	-.004	.6540	.421

* p=.05

The use of rule reviews measure was a dichotomous variable with possible scores of zero or one. A zero was scored if the student did not review any rules before exiting the instructional program. A one was scored if the student reviewed rules before exiting the instructional program.

The use of rule reviews ($F=12.066$, $p=.001$) predictive variable was a statistically significant predictor of posttest performance. The predictor use of rule reviews accounted for 11.6% of the variance in posttest performance. Use of rule reviews scores related positively to posttest scores ($\beta=.341$).

Hypothesis #2 - No statistically significant relationship exists between a student's use of all primary presentation forms (rule, example, practice) and posttest score.

The use of all primary presentation forms (rule, example, practice) measure was a dichotomous variable with possible scores of zero or one. Zero was scored if the student did not examine all three primary presentation forms, and a one was scored if they did.

The use of all primary presentation forms measure ($F=3.992$, $p=.049$) predictive variable was a statistically significant predictor of posttest performance. The predictor rule review accounted for 4.4% of the variance in posttest performance. All PPF scores related positively to posttest scores ($\beta=.204$).

Hypothesis #3 - No statistically significant relationship exists between the number of practice items correctly answered and the student's posttest score.

The correct practice predictive variable was a continuous variable that ranged from 0 to 52. This number indicates the number of practice items correctly answered by a student while examining the practice portion of the instructional software program.

The correct practice ($F=7.336$, $p=.008$) predictive variable was a statistically significant predictor of posttest performance. The predictor rule review accounted for 7.4% of the variance in posttest performance. Correct practice scores related positively to posttest scores ($\beta=.272$).

Hypothesis #4 - No statistically significant relationship exists between a student's time in the program and his or her posttest score.

The time in program predictive variable had a wide range among students. Students spent an

Table 3: Coefficients with Use of Rule Reviews, Use of All Primary Presentation Forms, Number of Practice Items Correctly Answered, and Time in Program as the Predictor Variable and Posttest as the Dependent Variable

Predictor Variable	B	Standardized Beta (β)
Use of rule reviews	14.097	.341
Use of all primary presentation forms	7.641	.204
Number of practice items correctly answered	.589	.272
Time in program	.006	.084

average of 17 minutes 56 seconds in the Relational Database Program. The minimum amount of time spent in the Relational Database program was 5 minutes 22 seconds, and the maximum time spent in the program was 33 minutes 9 seconds.

The total time in program ($F = .654$, $p = .421$) predictive variable was not a statistically significant predictor of posttest performance.

Conclusions

Based upon the findings of the research questions addressed, the following conclusions were made:

1. As evidenced by a statistically significant relationship found between students who reviewed the rules before exiting the instructional program (finding #1), and posttest performance, the conclusion is made that students using the Relational Database Program would likely benefit by having a review of the rules just prior to exiting the instructional program. Students who reviewed the rules before exiting the instructional program scored 14% higher on average on the posttest than those who did not review the rules ($B = 14.097$).
2. As evidenced by a statistically significant relationship found between students who examined all primary presentation forms (finding #2) and posttest performance, the conclusion is made that students using the Relational Database Program would likely benefit by being required to examine all three primary presentation forms. This finding is consistent with prior research on primary presentation forms. Students who examined all primary presentation forms scored 7.5% higher on average on the posttest than those who did not review primary presentation forms ($B = 7.641$).
3. As evidenced by a statistically significant relationship found between the number of correct practice items answered by a student (finding #3) and posttest performance, the conclusion is made that the Relational Database program may benefit by expanding and enhancing practice opportunities in the program. For example, program control should require students choosing not to work on practice items to complete some of the practice items. In addition, students may also benefit if practice items answered incorrectly were placed back into the practice item database to be used again. Students scored .6% higher on average for each additional practice item correctly answered ($B = .589$).
4. As evidenced by not finding a statistically significant relationship between time in the instructional program (finding #4) and posttest performance, the conclusion is made that time did not appear to be a factor in this particular software program. It appears that the predictive variable, time, should be removed from the evaluation overlay as a predictor of posttest performance.

Limitations and Recommendations for Further Research

The evaluation overlay that was designed and developed for this study was limited in scope. However, the research provides a foundation for monitoring and analyzing student usage of instructional software programs.

There appears to be a great deal of potential using an computer-tracking system for evaluating instructional software. The tracking system was relatively easy to develop and provided an excellent source of data to evaluate. The computer-tracking system also proved to be an excellent use of an unobtrusive data collection technique.

Limitations to this study should be discussed. One limitation is the students did not use the instructional software for a very long time. For instance, the average length of time in the instructional program was only about twenty minutes. To maximize the instructional effect of the software program, it could be argued that students should have spent more time in the instructional program. However, as previously discussed, time did not appear to be a factor in student posttest performance.

Another limitation of this research is the ability to generalize this study to a larger population. “Population validity is the extent to which the results of an experiment can be generalized from the sample that participated in it to a larger group of individuals, that is, a population" (Gall, Borg, & Gall, 1996, p. 217). Because participants used in this study were not randomly selected, population validity for this study would be considered low.

Participants for this study were selected for two reasons. First, they were available and readily accessible for the study. Second, they were selected because they were likely to be information-rich with respect to the purposes of the study. Gall, Borg, and Gall (1996) refer to this as purposeful sampling. The major focus of this study was not necessarily to find data that could be generalized to a large population, but rather to create an evaluation-overlay process to be used by instructional designers and software evaluators.

Further research is recommended to design an evaluation process that incorporates a computer-tracking system and evaluates usage according to specific instructional criteria. It seems possible that an effective evaluation overlay could be designed for a variety of instructional design theories. In addition, conducting a multiple regression analysis would also help to determine how variables found by analyzing the computer-tracking system relate when analyzed together.

Additional research is also recommended to examine more advanced designs of the computer-tracking system. For instance, the design could allow for students to comment when they run into a problem during the instruction. This research on using computer-tracking system to improve the software evaluation process will help instructional designers and software evaluators to develop more efficient, effective, and appealing instruction.

References

Barab, S. A., Fajen, B. R., Kulikowich, J. W., & Young, M. F. (1996). Assessing hypermedia navigation through pathfinder: Prospects and limitations. *Journal of Educational Computing Research, 15*(3), 185-205.

Gall, M. D., Borg, W. R., & Gall, J. P. (1996). *Educational research* (6th ed.). New York: Longman.

Gay, G., & Mazur, J. (1993). The utility of computer tracking tools for user centered design. *Educational Technology, 33*(4), 45-59.

Gibbons, A. S. (1995). *IT 635 custom course reader*. Logan, UT: Utah State University Instructional Technology Department.

Gibbs, W. J. (1995). Multimedia and computer-based instructional software: Evaluation methods. *Proceedings of the 28th Association of Small Computer Users in Education Summer Conference*, 18-22. (ERIC Document Reproduction Service No. ED 387 096).

Guba, E. G., & Lincoln, Y. S. (1981). *Effective evaluation*. San Francisco: Jossey-Bass Publishers.

Harvey, F. A., & Nelson, A. (1995). A hybrid investigation of hypermedia training. *Proceedings of the Association for Educational Communications and Technology, USA.* (ERIC Document Reproduction Service No. ED 383 299).

Lawless, K. A., & Kulikowich, J. M. (1996). Understanding hypertext navigation through cluster analysis. *Educational Computing Research, 14*(4), 385-399.

Merrill, M. D. (1994). *Instructional design theory*. Englewood Cliffs, NJ: Educational Technology Publications.

Misanchuk, E. R., & Schwier, R. (1991). Interactive media audit trails: Approaches and issues. *Proceedings of the Association for Educational Communications and Technology, USA*, 499-520.

Misanchuk, E. R., & Schwier, R. (1992). Representing interactive and hypermedia audit trails. *Journal of Educational Multimedia and Hypermedia, 1*(3), 355-372.

Pedhazur, E. J., & Schmelkin, L. P. (1991). *Measurement, design, and analysis: An integrated approach*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Schwier, R., & Misanchuk, E. R. (1990). Analytical tools and research issues for interactive media. (Report No. IR014519). Saskatchewan, Canada. (ERIC Document Reproduction Service No. ED 321 751).

Scriven, M. (1967). The methodology of evaluation. In R. E. Stake (Ed.), *Curriculum evaluation* (pp. 39-89). Chicago: Rand McNally.

Scriven, M. (1980). *Educational thesaurus* (2nd ed.). Pt. Reys, CA: Edgepress.

Tessmer, M. (1993). *Planning and conducting formative evaluations*. London: Kogan Page.

Trumbull, D., Gay, G., & Mazur, J. (1992). Students' actual and perceived use of navigational and guidance tools in a hypermedia program. *Journal of Research on Computing in Education, 24*(3), 315-328.

Worthen, B. R., & Sanders, J. R. (1987). *Educational evaluation.* New York: Longman.

Worthen, B. R., Sanders, J. R., & Fitzpatrick, J. L. (1997). *Program evaluation.* New York: Longman.