Designing CBT Systems with Errors in Mind: Avoidance, Seeding, and Tolerance

Ruben Quiñonez, Terry Ryan, and Lorne Olfman
Claremont Graduate University, Claremont, CA, USA

rubenquinonez@hotmail.com; terry.ryan@cgu.edu; lorne.olfman@cgu.edu

Executive Summary

This study attempts to reconcile viewpoints on the role of errors in the design of computer-based training (CBT) systems. From one perspective, errors are detrimental to learning; from another, they can be beneficial. The results of an experiment are presented and discussed. When a CBT system is designed to allow learners to correct their own errors, they learn significantly more than when errors are prevented. When learners are allowed only to correct errors that others have made, they do not learn significantly more than when errors are prevented. Implications for the design of CBT systems and possibilities for future research are considered.

Keywords: Computer-based Training, CBT, Errors, Individual Differences, System Design

Introduction

Users make errors when working with computers. Errors consume anywhere between 25 and 50 percent of the typical user's time (Arnold & Roe, 1987; Carroll & Meij, 1998) and are considered detrimental because they are stressful and frustrating. Trainers typically try to prevent learners from making errors (Frese et al., 1991), but errors are important in the learning process. Errors provide opportunities for learners to recognize that their understanding is incorrect or incomplete. Errors focus a learner's attention on those aspects of learning she must improve.

In the design of computer-based training (CBT) systems, two opposing general strategies concerning errors are evident: 1) an *error avoidance* approach, in which training is designed so that errors do not occur (Carroll, 1998; Leplat, 1989), and 2) an *error acceptance* approach, in which training is designed so that errors are used to advance learning (Frese et al., 1991; Sein & Santhanam, 1999). Within the error acceptance approach, it is possible to think of errors in at least two different ways. The first of these, which can be labeled *error seeding*, is grounded in the idea that learners should experience errors in a planned way. An error seeding design strategy for CBT systems would incorporate a set of errors into training materials so that all trainees learn to recognize and correct them. The second variety of error acceptance, which can be labeled *error tolerance* (or error induction), is based on the notion that learners make errors naturally, based on their own levels of understanding, as they attempt to accomplish training tasks. An error tolerance design strategy for CBT systems lets learners make errors, but does not seed errors, giving trainees the chance to learn to diagnose their own mistakes and correct them.

Editor: Mike Hart

These views of how to handle errors (i.e., error avoidance, error seeding, and error tolerance) have consequences for the design of CBT systems. This paper attempts to reconcile the three perspectives through reporting the results of a laboratory study using CBT systems designed to support the learning of database queries. The results of the study are important for understanding how training systems should be designed.

Theoretical Background

Research Model

Designers of training systems may be influenced by many factors, ranging from insights gleaned from personal training experiences to formal theories of training. In this light, one source of inspiration for training system designers could be a theoretical framework, which is "a collection of interrelated concepts, like a theory but not necessarily so well worked-out" (Analytictech, 2006). Bostrom, Olfman, and Sein (1990) provide a framework for end-user training in which training involves the development of end users' mental models. Within this framework, novice users can form mental models of a system through use of it, by analogy, or via training. Figure 1 depicts the research model used for this study. The model reflects those portions of the Bostrom et al. (1990) framework relevant to this research.

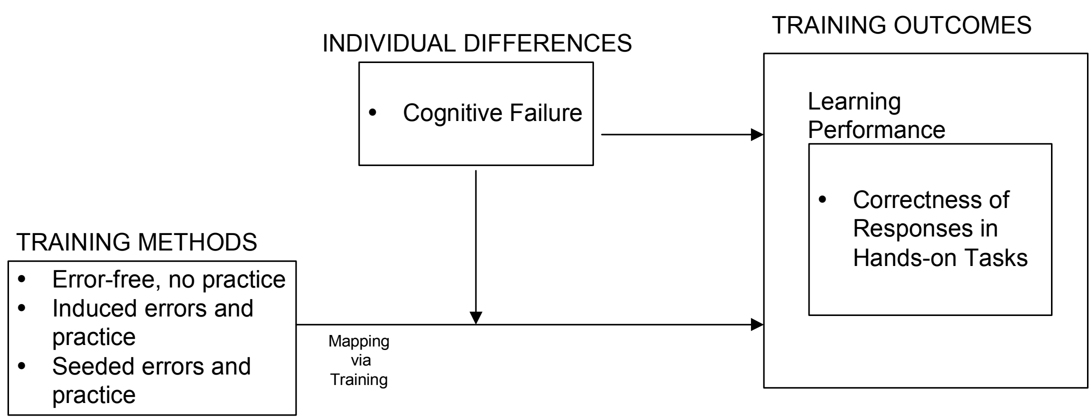


Figure 1: Research model for study

Errors and Training Methods

Error avoidance

One view of the role of errors in learning posits that they should be avoided because they are stressful and frustrating. One implementation of this view is minimalism (Carroll, 1984). Minimalism encourages giving less to the learner—less to read and less to get tangled in—leading the learner to achieve more. In a study of word processing training, Carroll and colleagues designed training situations to be as simple as possible and to make error states unreachable. They observe that learners in simple, error-free conditions grasp concepts faster, produce better work, and spend less time on errors than they would under conditions not so designed (Carroll, 1984). Minimalism has been applied to many different areas of information technology, such as user interfaces, programming, documentation, help systems, training, etc. (McCreary & Carroll, 1998).

Error acceptance

Another view of errors asserts that errors can be positive and should be included in the learning process. Schank (2002) claims, "For learning to take place, there has to be *expectation* failure" (p.62). He describes failure-driven CBT systems in which errors by learners play an essential role. These systems are based on computer simulations that allow learners to make mistakes that force them to improve the mental scripts they are employing.

Some researchers (Sein & Santhanam, 1999) advance a beneficial view of errors, finding that errors improve learning when learners recover from them by concentrating on the goal structure of their tasks. Other researchers (Frese et al., 1991) show that error training can lead to higher scores in performance. One group of learners in a word processing training experiment was not allowed to make errors; trainers intervened and reversed errors immediately. Another group of learners was allowed to make and correct errors without assistance from trainers. Learners who were allowed to make and correct errors learned more than did learners who were not allowed to make errors. This increase in learning was larger when learners faced more complex tasks.

Advocates of minimalism have also been known to recognize a positive effect of errors. Farkas (1998) acknowledged the benefits of errors in training, stating that reflecting on errors "increases understanding and promotes retention" (p.249) and suggested that designers should make errors positive opportunities for the trainee. By way of caution, he also pointed out that confusion, time pressure, or lack of interest on the part of trainees would diminish the value of errors in training.

Although there is substantial evidence to support the efficacy of the error avoidance approach, the error acceptance view has been under-researched. Furthermore, researchers have not directly compared what kind of errors ought to be included in training. An error tolerance strategy for CBT system design anticipates the induction of errors due to typical learner behavior. On the other hand, an error seeding strategy for CBT system design focuses on achieving a consistent exposure of all learners to important errors. The argument in favor of error tolerance is that CBT systems designed from this perspective allow each user to make errors appropriate to her current skill level and learn from them. The design benefit of error seeding is that it provides consistent error delivery to all learners, thereby assuring that no learners have gaps in the set of errors they have seen.

Individual Differences

Individual differences are trainee characteristics that impact training outcomes either directly or through interaction with the target system or training method. In short, the training literature indicates that individual characteristics must be assessed before trainees can be assigned to training groups (Borgman, 1986; Carroll & Carrithers, 1984; Compeau, Olfman, Sein & Webster, 1995; Shayo & Olfman, 1993). A review of the literature shows many individual difference variables (e.g., visualization ability and learning style) that affect learning (Bostrom et al., 1990; Sein & Bostrom, 1989). However, only one individual difference has been directly related to error training: failures in perception, memory, and motor function.

Cognitive Failure

Broadbent, Cooper, Fitzgerald & Parkes (1982) present the Cognitive Failures Questionnaire (CFQ), an instrument that measures failures in perception, memory, and motor function. They conclude that the CFQ instrument correlates with self-reported measures of memory deficits, absentmindedness, and slips of action. In addition, they report that people with high CFQ scores have increased vulnerability to stress.

Several studies have shown significant correlation between CFQ and measures of performance. Meiran, Israeli, Levi & Grafi (1994) report finding that cognitive failures are negatively correlated with speed of performance on focused attention tasks. In a study of 72 Australian undergraduates, Yates, Hannell & Lippett (1985) conclude that the CFQ correlates with cognitive interference that affects the ability to deal with stressful testing situations.

The CFQ has been used primarily in psychology and psychiatry (Broadbent, Broadbent & Jones, 1986; Houston, 1989; Kvavilashvili & Ellis, 1999; Matthews & Wells, 1988; Wagle, Berrios & Ho, 1999) and has been validated in several languages such as Japanese (Yamada, 1999), Spanish (Garcia-Martinez & Sanchez-Canovas, 1994), and German (Klumb, 1995). Only one study reporting its use was found in the training literature (Frese et al., 1991). This study concludes that error training seems to be positive for people with high scores on the CFQ. This is based on their hypothesis that subjects with high CFQ scores make more errors in general and should profit from error-management training. In the context of the current study, it is unclear that level of cognitive failure can be expected to interact with training method.

Hypotheses

In light of the preceding discussions, learners can be trained with three types of CBT system: error-avoidant, error-seeded, and error-tolerant. While, in general, the differences among the three types of CBT system can be expected to apply to a variety of training situations, in this research training involves novices learning to create database queries in Structured Query Language (SQL). Relative effectiveness of training in this situation is reflected, among other things, in post-training differences in trainees' performance in creating correct SQL queries. Expectations regarding the effects of type of CBT system on learning performance are reflected in the hypotheses below.

As stated earlier, one salient difference between the two error-training methods is that subjects who receive seeded errors will develop procedures to fix incorrect queries, while subjects who experience non-seeded (induced) errors will develop procedures to formulate new queries, *as well as* procedures to fix their own incorrect queries. Subjects who receive error-free training will not develop procedural knowledge of this kind at all. Based on the foregoing, the following hypotheses can be stated:

H1: Learners trained with an error-tolerant CBT system will perform better than those trained with an error-avoidant one.

H2: Learners trained with an error-tolerant CBT system will perform better than those trained with an error-seeded one.

H3: Learners trained with an error-seeded CBT system will perform better than those trained with an error-avoidant one.

It was expected that subjects with lower levels of cognitive failure would outperform those with higher levels of cognitive failure, however no interaction was expected.

H4: Subjects with low cognitive failure scores will perform better than subjects with high cognitive failure scores.

Research Method

A laboratory experiment was conducted to test the four hypotheses stated above. Participants were randomly assigned to experimental conditions.

Independent Variables

CBT systems

Three Web-based CBT systems were designed and implemented to support this study. The first system (error-avoidant) allowed no errors to occur, instead requiring learners to study and execute correct queries provided to them by the system. The second (error-seeded) included errors introduced by the system, having learners attempt to fix given incorrect queries. The third (error-tolerant) allowed user-generated errors, having learners attempt to create queries similar to given correct queries. Except for the design aspects related to treatment of errors, the three CBT systems were identical.

Each system included a tutorial on database queries in SQL. The systems assumed no prior knowledge of databases or SQL. In addition to the tutorial section, each system included two pre-test questionnaires, one performance testing section, and one post-test questionnaire. The tutorial sections varied from system to system; the other components did not.

Cognitive failures

The Cognitive Failures Questionnaire (Broadbent et al., 1982) was used to measure cognitive failures. The CFQ has 25 items, and scores range from 25 to 125. As in prior research (e.g., Frese et al., 1991), CFQ scores above the sample median were deemed to be high; those below, low.

Dependent Variable

Performance was expressed in terms of the creation of 10 database queries, one for each topic covered in training (see Table 1). Performance scores resulted from adding one point for each correct query created and subtracting ¼ point for each incorrect query attempt. For example, if a research subject created 8 correct queries without making errors, she would receive a score of 8. If another subject also created 8 correct queries, but in order to do so committed four errors, then her score would be 7, or [8 – (4 * ¼)]. As such, negative scores were possible. If a third subject answered eight questions correctly but in order to do so unsuccessfully tried 33 times, then her score would be -.25, or [8 – (33 * ¼)]. Although it is possible for someone who answered one or more questions correctly to receive an adjusted score lower than someone who did not answer any questions correctly, this situation was not observed.

Table 1. Performance Tasks

Task #	Topic
1	Simple SELECT
2	Conditional selection
3	WHERE clause
4	AND clause
5	OR clause
6	IN compound condition
7	BETWEEN compound condition
8	Aggregate functions
9	DISTINCT clause
10	ORDER BY clause

This scoring methodology attempts to adjust for the effects of additional practice and subsequent learning that some subjects may experience during the quiz. By taking incorrect attempts into account, it was intended that adjusted scores would be a more accurate reflection of SQL mastery at the beginning of the quiz (i.e., before quiz practicing might occur). The choice of ¼ point as the amount by which to adjust for the practice effects due to incorrect query efforts was somewhat arbitrary. (Substitution of other reasonable values [e.g., 1/3 or 1/5 point] did not significantly affect results.) In this study, raw scores ranged from 0 to 10 (out of a maximum of 10), and adjusted scores ranged from –26 to 10.

Participants and Random Assignment

The 113 participants in the experiment were undergraduate students—primarily business majors enrolled in an introductory information systems course—at a state university in the western US. Participants were recruited from course sections and were offered $25 gas cards as incentives to participate in the study. Because experimental sessions took place early in the college term, participants had not yet been exposed to database query concepts. Each experimental session was two hours in length and included an average of twelve participants. Random assignment was achieved through assignment of participants to sessions, each of which provided one method of training.

Tutorial Sections

The tutorial sections presented learners with SQL concepts and exercises, starting with the SELECT statement. The concepts and exercise formats used in the tutorials were similar to previous studies (e.g., Olfman & Mandviwalla, 1994).

For each concept, the tutorials provided the user with an explanation of it, a first exercise, and then a second exercise. The first exercise was always to study and run a sample query for the concept, while the nature of the second exercise varied by CBT system. With error-avoidant CBT systems, participants had as a second exercise to study and run another given query. With error-seeded CBT systems, participants had as a second exercise to fix a given incorrect query and then run it. With error-tolerant CBT systems, participants had as a second exercise to create a query, similar to the one in the first exercise, and then run it. Each query to be run appeared—or was typed by participants, depending on the type of CBT system—in an area on the screen. This area included a button that could be used to run the query. Upon execution, the results of the query—or an error message if appropriate—appeared in another area below the query area.

For the first exercise, all participants, regardless of CBT system type, were allowed to run the given query up to three times before proceeding to the second exercise. Participants experiencing the error-avoidant condition (i.e., using the error-avoidant CBT system) were allowed to run the second given query up to three times. Participants using the error-seeded CBT system were allowed up to three tries to fix and run the given incorrect query, and then were shown a correct version if necessary. Participants using the error-tolerant CBT system were allowed three attempts to formulate and run an appropriate query, being shown a correct version of the query if they were unsuccessful. Participants in the error-seeded and error-tolerant conditions could not continue from one concept to the next in their tutorials until their second exercise query ran successfully or they had made three tries.

All systems were designed with four additional characteristics in mind: 1) topic exercises, 2) a feedback mechanism, 3) a correction mechanism, and 4) error independence. All tutorials provided learners with two exercises per concept to prevent them from making poor generalizations, as suggested by (Reed, 1993). For tutorials in which errors could occur, a feedback mechanism provided an appropriate error message whenever a learner's response was incorrect. Whenever the learner's response was correct, the feedback mechanism displayed appropriate results and allowed her to proceed to the next topic. The correction mechanism consisted of a user interface that displayed the correct answer after three consecutive invalid tries. This provided feedback to the user and allowed her to continue to the next topic. Error independence in the tutorials implemented the concept of static simulation (Leplat, 1989), which prevented invalid responses for one topic from adversely impacting the learner's progress in her tutorial.

Performance Testing Section

Similar to the tutorial sections of the application, the performance testing section displayed one exercise at a time—one for each of the ten queries to be created—with a query area, an execution button, and an answer area below. Participants could type queries into (and correct queries in) the query area. Results from execution of queries, or error messages, appeared in the answer area. The CBT systems kept logs of learner behaviors, including the queries they submitted and the time they took for the various exercises in both the tutorial and performance testing sections of the experiment.

Schemas

Equivalent database schemas were used during the tutorial and performance testing sections of the experiment. Schemas were made available to participants in separate windows, if they choose to access them. Participants might choose to view schemas to see what tables and fields existed in the database they were using.

Procedure

The experiment was conducted in a laboratory designed for computer training, containing twenty-four workstations in rows facing the front of the room. At the front of the room, the researcher had one workstation equipped with an overhead projector.

Upon arrival at the experimental facility, participants were greeted by the researcher and immediately seated at a workstation. Table 2 provides details of the phases of the experimental procedure. Most of the procedure was controlled by the CBT systems. All data collection was done by the systems.

Table 2. Summary of Experimental Procedures

Phase 1	Researcher provides general instructions and introduces purpose of experiment. Participant signs letter of consent.
Phase 2	Participant logs-on to application and reads specific instructions.
Phase 3	Participant completes background questionnaire and “cognitive failures” questionnaire.
Phase 4	Participant accesses tutorial, which covers basic elements of SELECT, WHERE, conditional operators, and aggregate functions. Participant has 45 minutes to complete tutorial.
Phase 5	Participant continues with performance testing, which presents near-transfer and far-transfer tasks related to topics presented in tutorial. Participant has 45 minutes to complete performance testing.
Phase 6	Participant completes feedback questionnaire.
Phase 7	Participant has option to “surf the Web” or leave facilities after completing tasks.
Phase 8	Participant receives $25 gas card.

Results

Participant Characteristics

The sample size consisted of 38 participants in the error-avoidant condition, 37 in the error-seeded condition, and 38 in the error-tolerant condition, for a total of 113 participants. Of these 61 were female, 107 were full-time students, 111 were undergraduates, and 109 had no prior exposure to SQL. All participants completed all the sections of the experiment.

Cognitive Failure Questionnaire Scores

Cognitive Failures Questionnaire scores ranged from 33 to 109, with a mean of 64.9, a standard deviation of 11.2, and a median of 62.7. Thus, CFQ scores below 63 were classified as low (57 subjects) and above 63 as high (56 subjects).

Performance Testing Results

The dependent variable in this study was calculated as described in the Research Method section. The researcher and a database professional manually scored all queries; differences in scoring were discussed and resolved. Scores ranged from –26 to 10. Means and standard deviations by treatment group for performance scores are shown in Table 3.

Table 3. Performance Scores by Type of CBT System and CFQ Level

	Error-Avoidant	Error-Seeded	Error-Tolerant	Overall
Participants	38	37	38	113
Mean	-2.36	-.06	1.39	-.34
Low CFQ	-3.26	.16	1.25	-.68
High CFQ	-1.45	-.38	1.50	.00
Std. Dev	7.08	3.48	4.66	5.48

Performance scores were examined visually to determine if they followed an approximately normal distribution. As shown in Figure 2, scores are somewhat negatively skewed, but the degree of skewness was not viewed as problematic for the analysis performed. ANOVA is very robust to skewed distributions (Lindman, 1992, p. 109).

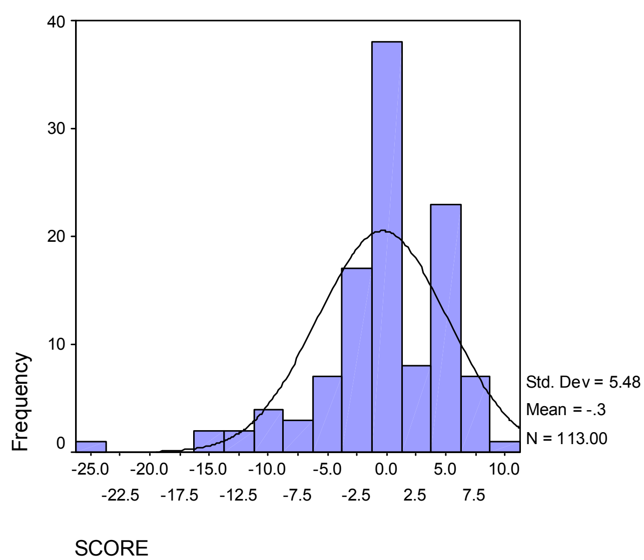


Figure 2. Performance score distribution

Hypothesis Testing

Results from the analysis of variance that was run to test the study's hypotheses are presented in Table 4. The criterion level of alpha = 0.05 was chosen for rejection of null hypotheses.

An analysis of the main effects shows that training method had a significant effect (F2,112=4.830 p=.010), although CFQ did not (F1,112=.250, p=.618). The latter result means that H4 must be rejected.

Table 4. ANOVA of Performance Scores by Type of CBT System and CFQ

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	306.091	5	61.218	2.142	.066
Intercept	14.672	1	14.672	.513	.475
Training Method	264.944	2	132.472	4.635	.012
CFQ	7.133	1	7.133	.250	.618
Training Method X CFQ	26.690	2	13.345	.467	.628
Error	3058.058	107	28.580		
Total	3377.438	113			
Corrected Total	3364.149	112			

A further analysis of H1, H2, and H3 was then conducted. Tukey post hoc results are shown in Table 5. Participants in the error-tolerant condition scored higher on performance testing than participants in the error-avoidant condition (p=.007); H1 may be accepted. Participants in the error-tolerant condition did not score higher on performance testing than participants in the error-seeded condition (p=.462); H2 must be rejected. Participants in the error-seeded condition did not score higher on performance testing than participants in the error-avoidant condition (p=.151); H3 must be rejected. A power analysis based on Cohen's (1977) power tables indicates that there was sufficient power to detect a no-difference conclusion for a medium effect size (f=.25-.30) for alpha = .05.

Table 5. Tukey Post Hoc Test

(I) Condition	(J) Condition	Mean Difference (I-J)	Std. Error	Sig.
Error-Avoidant	Error-Tolerant	-3.75*	1.22	.007
Error-Avoidant	Error-Seeded	-2.29	1.22	.151
Error-Tolerant	Error-Seeded	1.46	1.22	.462

* Difference significant at 0.05

Discussion

Results Supporting Hypotheses

There was support for the first hypothesis: participants who receive training with an error-tolerant system will perform better than those who use an error-avoidant one. This finding is consistent with predictions from theories in cognitive science, such as Anderson's (1983) theory of adaptive control of thought and Schank's (1982) theory of dynamic memory. The primary reason for this outcome is that participants who use the error-tolerant CBT system develop the productions (in Anderson's terms) or scripts (in Schank's) needed to formulate and correct their own queries. Participants who experience error-free training do not develop the required procedural knowledge to be able to create correct queries.

Results Rejecting Hypotheses

One expectation of this research study was that participants who used an error-seeded CBT system would perform better than learners who used an error-avoidant one. Participants who experienced seeded errors scored slightly higher than participants who received error-free training. Even though the actual scores were in the expected direction, they were not statistically significant. Why was the difference in performance between the error-avoidant and the error-seeded conditions less than expected?

A partial answer to this question is based on the manner in which participants appropriated the CBT systems with which they were provided. It had been expected that participants using the error-avoidant CBT system would run fewer queries during the entire experiment (i.e., the tutorial and performance testing sections combined) than participants using the other two systems, since the error-avoidant CBT system provides nothing but correct queries during tutorials. It was also expected that participants using the error-seeded CBT system would run slightly fewer queries during the experiment than participants using the error-tolerant one, because it is easier to fix queries than it is to create them. It was also expected that patterns of time spent during the experiment would be similar to patterns of query execution. Figures 3 and 4 indicate that these expectations were accurate.

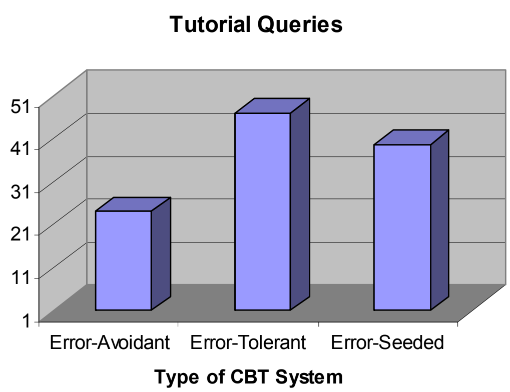


Figure 3: Tutorial queries per treatment (mean)

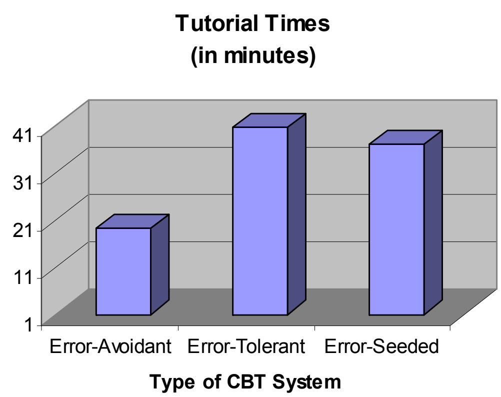


Figure 4: Session times per treatment (mean)

Patterns for queries run and time spent during performance testing (as opposed to during the entire experiment) show some interesting and unexpected results, though. Figure 5 shows that participants using the error-avoidant CBT system ran considerably more queries during performance

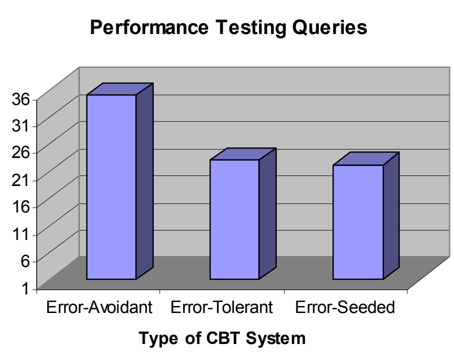


Figure 5: Queries during performance testing by treatment (mean)

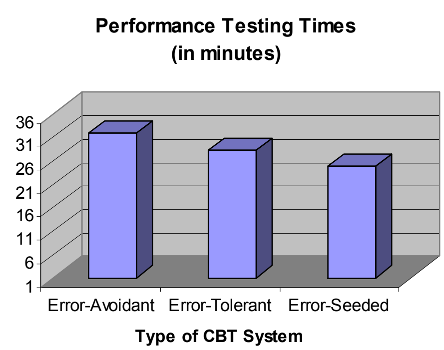


Figure 6: Session times for performance testing per treatment (mean)

testing than the other two groups did. Figure 6 confirms their increased activity in that they also spent more time in performance testing.

A protocol analysis revealed that participants who used the error-avoidant CBT system were using the performance testing section of the experiment to figure out how to create and debug queries, in much the same way that participants in the error-tolerant condition had used the tutorial section to do this. (Of course, the concept descriptions and correct examples were no longer available to participants during performance testing.) Since they had not been able to practice during the tutorial, but could only run the correct query examples, error-avoidant CBT system participants practiced creating and debugging queries during performance testing. Such practice was made easier because no limit existed on the number of queries that could be run during performance testing. Participants had no compiled productions or scripts due to training, but they were able to produce practice queries, in an effort to compile them, as needed. The effect of such efforts was that error-avoidant CBT system participants scored higher than expected.

One explanation for why error-seeded CBT system participants did not perform better than they did could be based on cognitive theory. As discussed above, these participants went through a series of tasks designed to teach them how to fix incorrect queries, not how to formulate new queries. During performance testing, these same participants were challenged to create queries, but they would have lacked the appropriate productions or scripts to do so. Notably, error-seeded CBT system participants did not behave as error-avoidant CBT system participants did, by using performance testing for practice. Perhaps this is due to having developed an inappropriate sense of confidence from their experiences in fixing incorrect queries, leading to frustration and disengagement during performance testing when they found, against recent expectations, that they lacked the understanding needed to succeed.

Another explanation could be that error-seeded CBT system participants did not concentrate on understanding the entire statement, but rather only its sub-components, when they were fixing problems with queries during the tutorial. This behavior would be consistent with the Winograd and Flores (1986) notion of "breaking down" cognitive objects. Error-seeded system participants might have assumed that other parts of the queries were correct and therefore not attended to them. Error-tolerant CBT system participants, on the other hand, could not have done this, since no query was supplied by the system. As support for this conjecture, it should be noted that 20 error-seeded CBT system participants made simple syntax errors, compared with only 11 error-tolerant CBT system participants. (Simple syntax errors would include such mistakes as running "Select*from" without the needed spacing between terms.)

The study included the CFQ instrument, but it showed no effects. The fact that the cognitive failures instrument failed to meet expected outcomes suggests that more research is needed to assess its applicability to the training domain.

In summary, two points are important. First, use of an error-tolerant CBT system led to the development of the productions or scripts needed to formulate queries. This provided significantly better results than use of a system designed to be error-avoidant. Second, participants who used an error-avoidant CBT system tried to practice during performance testing to compensate for the lack of practice during the tutorial, leading to higher than expected performance testing scores. This may have been enough to prevent a difference between outcomes from use of error-avoidant and error-seeded CBT systems from being significant.

Conclusion

Implications for Human-Computer Interaction

The findings in this study show the importance of error-based training. CBT systems that are tolerant of mistakes allow learners to develop procedural knowledge by forcing them to recognize and correct what they have done wrong. While not all errors are likely to be beneficial to learners, those that focus the learner's attention on the procedure to be learned are likely to be. Designers of CBT systems should be hesitant about avoiding errors completely, but should focus instead on building systems that are good at helping learners to identify and correct the mistakes that they make.

As demonstrated in this research, letting learners commit, identify, and correct their own errors is beneficial for learning, but this approach, in isolation, would be limited. Not all learners would make the same errors. Whether a particular user happened to make an error of a given kind would be haphazard. Such inconsistency among learners may not always be what is most desirable in training. Perhaps, seeded error training could also prove to be effective, with further refinement. The potential benefit of seeded errors is that it could force trainees to deal with *typical* errors, assuring that they are prepared to handle them after training is completed. CBT designers (and researchers) should consider the prospect of hybrid CBT systems, in which seeded errors and user-generated errors are viewed as complementary.

Study Limitations

One limitation of this study is that, even though they were well defined, the treatments failed to be operationalized as accurately as might be desired. The most important example of this is that error-avoidant CBT system participants were not supposed to be able to practice creating queries, but they did so anyway, during testing.

Given a possible concern for better operationalization of treatments, it is possible that the magnitude of overall scores could have been affected. This magnitude might also be viewed as having been affected by the level of the testing, the level of the students, or some other factor. Given that the focus of this study was only on relative differences in amount of learning across treatments, magnitude of scores was not explicitly treated in this work and little can be said about it.

The study faces another limitation in terms of the generalizability of learning SQL syntax from a simple database schema. The primary focus here was on how different error design strategies for CBT systems might affect learning performance. Other generalizations regarding learning of database concepts should be made with caution.

As with most experimental designs, the measurements were taken in one session. A longitudinal experimental design would be required to reveal other aspects of the learning process (e.g., retention) that might be of interest.

Future Research

In future research, a number of design changes could be applied to the experimental study done here. Refinement of performance testing could be achieved in many ways. More tasks per concept could be included. This might strengthen both the reliability and validity of scores obtained. A limit could be placed on the number of attempts that could be made in testing. This would mitigate the effects of practice during testing. Different forms of task could be included in testing, since learning can be viewed in terms of declarative knowledge and query correction, not simply query creation.

Additionally, assessment could put more emphasis on far-transfer tasks, in which learners must adapt what they have learned to solve novel tasks (Schuh, Gerjets, & Scheiter, 2005).The salient difference between near- and far-transfer tasks is that the solution of the latter cannot be found directly in the training materials, but has to be inferred, or constructed, from the provided information (Sein, 1988). The challenge for the trainee is to determine which combination of near-transfer tasks is required to complete the task. Assessment with far-transfer tasks can be viewed as getting at an understanding that can be used outside of the immediate context of the training. As such, it may reflect better the outcomes that are desired in SQL- and similar kinds of training.

Improved experimental sensitivity might be achieved by increasing the time allowed for both tutorial and testing sections. Some participants reported that they felt pressured for time while going through the training application. Increasing the number of times queries could be submitted during tutorials, beyond the three attempts allowed, might also have benefit, although this could lead to increased frustration of participants having difficulty with a concept.

A modified research design where the participants are given a “cheat sheet” during performance testing might bring better results. Some participants indicated that, even though they understood the material, it was difficult for them to remember all the syntax details during performance testing. Since the objective of the training is for the trainee to *understand* the material, not necessarily to *memorize* the material, this seems justified.

Finally, a modified session format where the participants are asked to “think out loud” during the tutorial and performance testing portions of the experiment might bring additional information. This modification should allow better capture of attitudes toward the training.

This study, however, contributes to a better understanding of the role of using errors in software training. The findings show that the beneficial view of error training has merit. The study provides support for using induced error training, and also hints that there is some support for the use of seeded errors in training materials. Additional research on how CBT systems should incorporate error strategies is needed.

References

Analytictech. (2006). Elements of research. Retrieved August 2, 2006, from http://www.analytictech.com/mb313/elements.htm

Anderson, J. R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Arnold, B., & Roe, R. (1987).User errors in human-computer interaction. In M. Frese, E. Ulich, & W. Dzida (Eds.), *Psychological issues of human-computer interaction in the work place.* Amsterdam: Elsevier.

Borgman, C. L. (1986) The user's mental model of an information retrieval system: An experiment on a prototype online catalog. *International Journal of Man-Machine Studies, 24*, 47-64.

Bostrom, R. P., Olfman, L., & Sein, M. K. (1990). The importance of learning style in end-user training. *MIS Quarterly,* 101-119.

Broadbent, D. E., Broadbent, M. H., & Jones, J. L. (1986). Performance correlates of self-reported cognitive failure and of obsessionality. *British Journal of Clinical Psychology, 25*(4), 285-299.

Broadbent, D. E., Cooper, P. F., Fitzgerald, P., & Parkes, K. R. (1982). The cognitive failures questionnaire (CFQ) and its correlates. *British Journal of Clinical Psychology, 21,* 1-16.

Carroll, J. M. (1984). Minimalist training. *Datamation, 30*(18), 125-136.

Carroll, J. M. (1998). Reconstructing minimalism. In J. M. Carroll (Ed.), *Minimalism beyond the Nurnberg funnel* (pp. 1-17). London: MIT Press.

Carroll, J. M., & Carrithers, C. (1984). Training wheels in a user interface. *Communications of the ACM, 27*(8), 800-806.

Carroll, J. M., & Meij, H. v. d. (1998). Ten misconceptions about minimalism. In J. M. Carroll (Ed.), *Minimalism beyond the Nurnberg funnel.* London: MIT Press.

Cohen, J. (1977). *Statistical power analysis for the behavioral sciences*. New York, NY: Academic Press.

Compeau, D., Olfman, L., Sein, M., & Webster, J. (1995). End-user training and learning. *Communications of the ACM, 38*(7), 24-26.

Farkas, D.K. (1998). Layering as a safety net for minimalist documentation. In J. M. Carroll (Ed.), *Minimalism beyond the Nurnberg funnel.* London: MIT Press.

Frese, M., Brodbeck, F., Heinbokel, T., Mooser, C., Schleiffenbaum, E., & Thiemann, P. (1991). Errors in training computer skills: On the positive function of errors. *Human-Computer Interaction, 6*, 77-93.

Garcia-Martinez, J., & Sanchez-Canovas, J. (1994). Adaptacion del cuestionario de fallos cognitivos de Broadbent, Cooper, Fitzgerald y Parkes. *Analisis y modificacion de conducta, 20*(73), 727-752.

Houston, D.M. (1989) The relationship between cognitive failure and self-focused attention. *British Journal of Clinical Psychology, 28*, 85-86.

Klumb, P.L. (1995). Cognitive failures and performance differences: Validation studies of a German version of the cognitive failures questionnaire. *Ergonomics*, *38*(7), 1456-1467.

Kvavilashvili, L., & Ellis, J.A. (1999). The effects of positive and negative placebos on human memory performances. *Memory, 4,* 421-437.

Leplat, J. (1989). Simulation and simulators in training: Some comments. In L. Bainbridge & S. A. R. Quintanilla (Eds.), *Developing skills with information technology* (pp. 277-292)*,* West Sussex, England: John Wiley.

Lindman, H. R. (1992). *Analysis of variance in experimental design.* New York, NY: Springer-Verlag.

Matthews, G., & Wells, A. (1988). Relationships between anxiety, self-consciousness, and cognitive failure. *Cognition & Emotion, 2*(2), 123-132.

McCreary, F. A., & Carroll, J. M. (1998). Appendix: Reviews, general discussions, and applications of minimalism since 1990. In J. M. Carroll (Ed.), *Minimalism beyond the Nurnberg funnel.* London: MIT Press.

Meiran, N., Israeli, A., Levi, H., & Grafi, R. (1994). Individual differences in self reported cognitive failures: The attention hypothesis revisited. *Personality & Individual Differences, 17*(6), 727-739.

Olfman, L., & Mandviwalla, M. (1994). An experimental analysis of end-user software training manuals. *Information Systems Journal, 5*, 19-36.

Reed, S. K. (1993) A schema-based theory of transfer. In D. K. Detterman & R. J. Sternberg, (Eds.), *Transfer on trial: Intelligence, cognition, and instruction.* Norwood, NJ, Ablex.

Schank, R. C. (1982). *Dynamic memory.* New York: Cambridge University Press.

Schank, R. C. (2002). *Designing world-class e-learning.* New York: McGraw-Hill.

Schuh, J., Gerjets, P., & Scheiter, K. (2005). Fostering the acquisition of transferable problem-solving knowledge with an interactive comparison tool and dynamic visualizations of solution procedures. *CogSci 2005 - XXVII Annual Conference of the Cognitive Science Society*, July 21-23, 2005, Stressa, Italy, pp. 1973-1978.

Sein, M. K. (1988). *Conceptual models in training novice users of computer systems: Effectiveness of abstract vs. analogical models and influence of individual differences.* Unpublished doctoral dissertation, Indiana University, Bloomington.

Sein, M. K., & Bostrom, R. P. (1989). Individual differences and conceptual models in training novice users. *Human-Computer Interaction, 4,* 197-229.

Sein, M. K., & Santhanam, R. (1999) Research report: Learning from goal-directed error recovery strategy. *Information Systems Research, 10*(3), 276-285.

Shayo, C., & Olfman, L. (1993) Is the effectiveness of formal end-user software training a mirage? *Proceedings of the 1993 conference on computer personnel research*, ACM Press, 88-99.

Wagle, A. C., Berrios, G. E., & Ho, L. (1999) The cognitive failures questionnaire in psychiatry. *Comprehensive Psychiatry, 40*(6), 478-484.

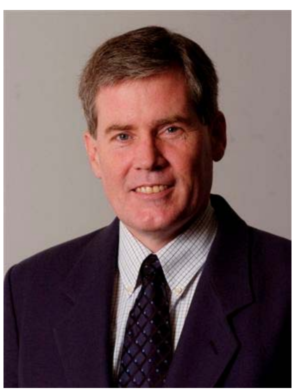
Winograd, T., & Flores, F. (1986). *Understanding computers and cognition.* Norwood, N.J.: Ablex.

Yamada, N. (1999) Error proneness questionnaire: Construction, reliability and validity. *Japanese Journal of Educational Psychology, 47*(4), 501-510.

Yates, G. C., Hannell, G., & Lippett, R. M. (1985) Cognitive slippage, test anxiety, and responses in a group testing situation. *British Journal of Educational Psychology, 55*(1), 28-33.

Biography

Ruben Quinonez is the Vice President of Technology at PayPoint, a subsidiary of First Data Corporation. In his 20 years of IT experience in private and government sectors, Dr. Quinonez has concentrated on application development, database design and tuning, and information security. He has published many papers in the field and has taught at several universities. He holds graduate degrees in Business Administration and Information Systems and a Ph.D. in Information Systems



Terry Ryan is an associate professor in the School of Information Systems and Technology at Claremont Graduate University and co-director (with Lorne Olfman) of the Social Learning Software Lab (SL2). His research interests are in the design, development, and evaluation of information systems to support social learning

Lorne Olfman is dean of the School of Information Systems and Technology at Claremont Graduate University, Fletcher Jones chair in technology management, and co-director (with Terry Ryan) of the Social Learning Software Lab (SL2). His research interests are in designing effective collaboration, learning and knowledge management technologies