1993

Computer Training: The Role of Computer Attitudes and Behavior Modeling in the Acquisition of Declarative and Procedural Knowledge.

Drew Benjamin Brock
Louisiana State University and Agricultural & Mechanical College

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_disstheses

Recommended Citation
https://digitalcommons.lsu.edu/gradschool_disstheses/5615

This Dissertation is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Historical Dissertations and Theses by an authorized administrator of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.
INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
Computer training: The role of computer attitudes and behavior modeling in the acquisition of declarative and procedural knowledge

Brock, Drew Benjamin, Ph.D.
The Louisiana State University and Agricultural and Mechanical Col., 1993
COMPUTER TRAINING:
THE ROLE OF COMPUTER ATTITUDES AND BEHAVIOR MODELING
IN THE ACQUISITION OF DECLARATIVE AND PROCEDURAL KNOWLEDGE

A Dissertation
Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Psychology

by
Drew Benjamin Brock
B.A. University of Texas at Austin, 1988
M.A. Louisiana State University, 1991
December, 1993
Table of Contents

List of Tables ........................................ iii
List of Figures .......................................... v
Abstract .................................................. vi
Introduction .............................................. 1
Cognitive Learning Theory .............................. 6
Social Learning Theory ................................... 18
Integrating Cognitive and Social Learning Theory ... 28
Relations Between Computer Knowledge and Performance . 34
Knowledge Acquisition and Computer Training .......... 38
Computer Attitudes and Computer Training .............. 45
Method .................................................. 59
Results ................................................... 77
Discussion ................................................ 110
References ............................................... 164
Appendix A - Outline for Key Learning Points Lecture 171
Appendix B - Task List for Practice Session .......... 177
Appendix C - Sample Modeling Intervention Script .. 182
Appendix D - Declarative Knowledge Test .............. 184
Appendix E - Procedural Knowledge Test .............. 186
Appendix F - Computer Performance Test .............. 189
Appendix G - Beneficial Tool Attitude Scale .......... 192
Appendix H - Autonomous Entity Attitude Scale ....... 194
Appendix I - Factor Loadings for Knowledge Items .. 196
Vita ...................................................... 197
List of Tables

Table 1. Breakdown of the Procedure Used to Assess the Hypotheses in the Present Study ........ 60

Table 2. Summary of Task Practice Attempts and Times by Task Group ..................... 65

Table 3. Sample Sizes, Means, Standard Deviations, Correlations, and Internal Consistency Measures for All Study Variables .......... 78

Table 4. Blocked Regressions for Declarative and Procedural Knowledge on Computer Performance Measures .................. 85

Table 5. Modeling by Lecture by Time ANOVA for Declarative Knowledge .................. 89

Table 6. Modeling by Lecture by Time ANOVA for Procedural Knowledge Using the Procedural Knowledge Test. 90

Table 7. Modeling by Lecture ANOVA for Procedural Knowledge Using Accuracy, Speed, and Efficiency Performance Measures ........ 91

Table 8. Pre-test and Post-test Declarative Knowledge Means and Standard Deviations by Training Condition .................. 92

Table 9. Pre-test and Post-test Procedural Knowledge Means and Standard Deviations by Training Condition .................. 93

Table 10. Accuracy, Speed, and Efficiency Means and Standard Deviations by Training Condition .................. 95

Table 11. Modeling by Lecture by Time ANOVA for Beneficial Tool Attitudes. .................. 98

Table 12. Modeling by Lecture by Time ANOVA for Autonomous Entity Attitudes. .................. 99

Table 13. Pre-test and Post-test Beneficial Tool Attitudes by Training Condition .................. 100

Table 14. Pre-test and Post-test Autonomous Entity Attitudes by Training Condition .................. 101

Table 15. Multiple Regressions for Pre-test Beneficial Tool and Autonomous Entity Attitudes on Knowledge Acquisition .................. 104
Table 16. Multiple Regressions for Pre-test Beneficial Tool and Autonomous Entity Attitudes on Performance Measures ........................ 105

Table 17. Modeling by Lecture ANOVAs for Practice Errors and Practice Time ........................ 125
List of Figures

Figure 1. Computer Attitude Interaction on Computer Use in Brock (1993) ........................................... 55

Figure 2. Computer Attitude Interaction on Performance Accuracy .................................................. 109

Figure 3. Time by Training Condition Cell Means for Autonomous Entity Beliefs ......................... 134
Abstract

This research examined technology training (using computers as a specific instance of technology) in a framework that integrated concepts drawn from cognitive and social learning theory. Borrowing from cognitive learning theory, two types of knowledge were assessed: declarative knowledge (knowing what) and procedural knowledge (knowing how). Behavior modeling training, drawn from social learning theory, was used as the primary training methodology. It was proposed that the factors contributing to learning as a function of behavior modeling are the same factors that lead to procedural knowledge acquisition. I also examined the effects of two major attitudes toward computers in predicting computer knowledge and the role of training methodology in altering attitudes. Using 255 undergraduate subjects divided into four training conditions, the study explored the effects of modeling training and lecture training (in a crossed design) on declarative and procedural knowledge (measured using both a paper and pencil test and a performance test). Knowledge and computer attitude assessment occurred before and after training.

I found declarative and procedural knowledge to be highly related to performance with the latter being significantly more so than the former. Analysis of the training methods indicated that modeling was an effective
means of instructing individuals on computer use. Lecture training was not superior to practice alone in increasing trainee knowledge, but was effective in improving performance. Pertaining to attitudes, beliefs in the computer as a beneficial tool were increased in training conditions that received a lecture whereas beliefs in the computer as an autonomous entity were decreased in conditions that did not receive a lecture. Pre-test autonomous entity beliefs significantly predicted computer knowledge acquisition. This research suggests that modeling training does not require a lecture component to be effective as a means of training individuals on computers. Future research should replicate these findings and further explore the proposed relationship between cognitive and social learning theory in other domains aside from computers. The study also suggests implications for research on how computer attitudes affect learning, and on how training programs can alter trainee attitudes toward the machines.
Introduction

A consideration of the major factors influencing the workforce of the 20th century would fall far short of being complete if it failed to include the impact of computer technology. The advent of computer-related technologies in our society represents nothing less than a revolution in the means by which tasks are carried out in the workplace. This revolution is particularly noteworthy when one considers it in the context of other innovations. For example, Gantz (1986) notes that the personal computer has, in a few years, acquired the equivalent societal penetration of the telephone in its seventy-five years of use. Further examples of this phenomenon are abundant in everyday life. Virtually all aspects of white-collar, and a growing number of blue-collar, workplaces absolutely require some form of computer technology to remain effective. Two maxims seem appropriate in this context: first, that computer-related technology has significantly altered the landscape of the workplace (e.g., computers replacing typewriters, databases replacing filing cabinets), and second, that this alteration has been swift and unrelenting.

Considering these two factors, it is not surprising to note a concomitant increase in the needs of those individuals who use the new technologies. Naisbitt (1982) perceptively noted this widening gap between technology
and the human element through his introduction of the term "high touch." Naisbitt argues that increases in "high tech" must be matched with increases in accessibility to the human element (i.e., "high touch"). This contrast seems to polarize the demands of technology with the needs of the worker; a view which has a parallel in the growing computer technology literature.

Technology researchers tend to possess either a positive or negative outlook on computer-related technology prior to conducting their research (Long, 1987; Turnage, 1990). Idealists take a positive outlook on the impact of technology on society and see it as an opportunity for workers (e.g., Martocchio, 1992), or as a beneficial tool for humans (e.g., Lee, 1970). In contrast, nay-sayers, who view computers in a negative light, tend to characterize the technology in terms of deskilling (i.e., decreasing the skills required to do a job thereby decreasing the need for highly skilled labor), or in terms of the physical and physiological problems associated with the implementation of office automation (e.g., Smith, Cohen, & Stammerjohn, 1981).

Recently, Turnage (1990) has reiterated this dichotomy (i.e., idealists versus nay-sayers) in her review of the impact of computer-based technology on the field of psychology. Rather than contrasting these views, Turnage argues for merging the polarized perspectives in
order to increase the productivity of the American workforce. She points to the importance of matching the needs of the American worker with the explosive demands for new skills related to computer technology. In essence, it is essential that the field of Industrial/Organizational Psychology begin to look at the means by which we train our workers for the technology dominated workplaces of the future.

One major purpose of the present research directs itself at the comments of Turnage (1990) as well as other reviewers of work-related technological research (e.g., Parsons, 1988). We are hearing the increasingly common cry that the impact of psychology on technology falls far short of keeping pace with the expansive penetration of technology in the workplace. Thus the first purpose of this research directs itself at narrowing the gap between psychological research on technology and the effects of that technology in the workplace. There are two avenues for further investigation in the research literature: attitudes toward computer-based technology and technology training. To examine the interplay between attitudes and training, I will endeavor to bring together the research which: 1) examines relations between attitudes and computer use, and 2) suggests that attitudes have some relation to training outcomes.
In initiating my analysis of attitude and ability research on technology, I chose to narrow the examination to computer based technology specifically. It can be argued that computers themselves are the most pervasive and representative form of technology in the workplace. Further, other forms of new technology (e.g., fax machines, automated tellers, robotics) lack the empirical base from which a sound theoretical examination may begin.

In order to examine the effects of computer training on trainee attitudes toward and ability with computers, I conducted a review of the psychological literature relevant to computers. One general observation from this review brings out a major distinction in our literature: research on computer training is not synonymous with training that uses computers. That is, we have a proclivity toward using computers to train individuals without fully regarding the impact of the computer itself on the training results. More research is needed which directly examines the computer's impact on the user.

Another observation I gained from the extant literature that specifically addresses computer training concerns the co-existence of two apparently independent theories of learning. At the macro-focal level, researchers tend to use social learning theory as a theoretical base (e.g., Goldstein, 1993). In examining more micro-level issues in learning, researchers adopt the
cognitive learning framework (e.g., Glaser & Bassock, 1989). The second major purpose of the present study is to integrate concepts borrowed from the two theories within the context of computer training.

Cognitive learning theory is discussed first to provide a theoretical framework for the knowledge-based aspects of computer training. Once the general cognitive learning principles have been put forth, a discussion of social learning theory and its methodological counterpart, behavioral modeling training, is presented. In serving the second purpose of this effort, concepts from the two theories are subsequently linked together to form an integrated theoretical framework for the present study. Within this context, research on computer attitudes is brought in to complete the design of the study. Thus my approach is two-pronged: first, to examine how readily the components of the two learning theories may be combined, and second, to provide a testable theoretical basis for training in computer technology and the role of computer attitudes in training success.
Cognitive Learning Theory

At its most basic level, training is the process of facilitating the acquisition of a skill or learned behavior (Hinrichs, 1976). In the context of organizations, McGehee and Thayer (1961) define training as formal organizational procedures used to facilitate learning that contributes to the goals of the organization. These definitions point to the centrality of learning in the context of training. Consequently, a discussion of the learning process in general can serve to form a theoretical basis for the present research and a framework for the discussion of further topics.

Learning has been defined as a "relatively permanent change in knowledge or skill produced by experience." (Weiss, 1990, p. 172). From this definition, Weiss (1990) points out several aspects of learning theory that are particularly instructive. First, we can distinguish learning from performance on the basis of behavior. Performance is the execution of learned behavior, therefore, it is possible to include the concept of learning without performance (a lack of transfer of training would explain a phenomenon such as this).

A second aspect of Weiss' (1990) definition concerns the issue of experience. Experience separates the learning construct from other influences on behavior such as traits, dispositions, and maturational factors. Under
this definition, learning through experience is assumed to be ubiquitously accessible across individuals regardless of their particular characteristics. A further point concerning learning and experience is that the knowledge acquired results from changes that possess a certain degree of permanence. Thus, more transient changes (e.g., fatigue) are not considered learning.

A final issue in this definition of learning that speaks directly to the present study concerns the emphasis in the definition on a change in knowledge or skill. This change has two components which are relevant here: the acquisition of skill (i.e., moving from no skill to some quantity of skill), and the transformation of existent skill (i.e., moving from novice to expert within a skill). The next section more fully explicates these two components.

The Acquisition of Skill

As has been noted above, the acquisition of skill is central to learning, which, in turn, is central to the success of any training program. The process of skill acquisition has received extensive attention in the cognitive learning literature. Fitts (1964; Fitts & Posner, 1967) divided the process of skill acquisition into three distinct stages. First, the cognitive stage represents the period in which the necessary skills are encoded such that a desired behavior may be attained.
These initial encodings are considered to be crude approximations of the desired behavior. Following this stage is a process of refinement which Fitts (1964) characterizes as the associative stage. It is in this stage that the initial approximations of the new skill become refined. Errors in initial encoding are identified and removed resulting in more efficient performance of the skill. Finally, the autonomous stage represents an extended, if not indefinite, period of continued improvement in the acquired skill. The distinguishing mark between the second and third stages is qualitative in nature. Individuals in the associative stage are "trouble-shooting" while individuals in the autonomous stage are "fine-tuning" their skills.

While these three stages may appear logical, it was the research of Anderson (1982) that first incorporated the framework into a systematic cognitive theory. Anderson (1982) introduced two concepts that are central to his theory as well as to the research presented here: declarative and procedural knowledge. The first step in the skill acquisition process is the encoding of facts about the skill domain. These facts are termed declarative knowledge and are typically characterized as static (i.e., unchanging in content), flexible in their organization, and describable (Best, 1989). In sum, declarative knowledge consists of "knowing what"
statements. Suppose for example that my knowledge of bicycle riding consists of the following: performing the skill requires a bicycle; there are two pedals which are used to propel the rider; handle bars are used to steer the bicycle; gears are used to vary the ratio of the turning pedals with the wheel. All of these pieces of information may be termed declarative knowledge; they are static (i.e., these characteristics of bicycle riding are unlikely to change) and clearly describable. Further, I can use this information in a variety of ways. For example, if I were interested in categorizing forms of locomotion, my knowledge could serve to place "bicycle riding" in the category of "non-motorized locomotion." Alternatively, I may be interested in identifying all objects which have gears. Here my knowledge would place bicycles alongside other objects such as clocks, mills, and cars. My declarative knowledge of bicycle riding is therefore quite flexible in its application.

Sequential to declarative knowledge is the knowledge of methods, strategies, and approaches to the skill domain. Anderson (1982) terms these processes procedural knowledge or knowledge which is characterized by "knowing how" statements. In contrast to declarative knowledge, procedural knowledge is dynamic, skill specific (i.e., its organization is bound to one skill), and difficult to describe (Best, 1989). To continue with the previous
example, my procedural knowledge of bicycle riding might consist of the following: to propel the bicycle you have to sit on the seat, place your feet on the pedals, and churn your legs; to steer the bicycle you have to hold the handlebars and pull the arm associated with the direction you want to go towards you; to go faster you have to increase the gear ratio by lifting the gear lever (you will also have to exert more force in pedaling). My procedural knowledge of bicycle riding may change if, for example, I practice enough and determine that sitting on the seat is not a requirement for bicycle riding. In this sense procedural knowledge is considered to be dynamic. The task specificity of my knowledge of bicycle riding is extensive; there are very few other situations in which my knowledge would be useful. Finally, as many would attest, it is much easier to show someone how to ride a bicycle than it is to describe how a bicycle is ridden.

As can be seen, declarative knowledge precedes procedural knowledge in that one must first possess facts about a skill before one can readily understand how to implement (or proceduralize) the skill. In keeping with the framework proposed by Fitts (1964), Anderson treats declarative knowledge as the first stage and procedural knowledge as the third stage. Intermittently, Anderson (1982) uses the term knowledge compilation to describe the processes involved in transmuting declarative knowledge to
procedural knowledge. Thus the declarative and procedural stages are characterized by the accumulation of declarative and procedural knowledge, respectively, while the knowledge compilation stage involves the process of practice with declarative knowledge to the point where it becomes procedural. In the following sections, I provide a discussion of each of the three major stages involved in knowledge acquisition.

The Declarative Knowledge Stage

As I mentioned above, the key feature of the declarative knowledge phase of skill acquisition involves the encoding of facts about the task. Anderson (1982) points out that information processing at this stage is not guided by procedures for encoding. As such, learners must apply their own general and fallible procedures to the information in the hopes of correctly interpreting the information. The method of interpretation at this stage involves a series of weak-method (or general purpose) problem solving strategies which may be utilized for all learning situations (Anderson, 1987; Glaser & Bassok, 1989; Gray & Orasanu, 1987). These strategies represent the methods individuals use to work out puzzling new situations. Since the situation is novel, the strategies are termed "weak-methods" because they have not been refined to fit the specific problem. Weak-method problem solving methods act on declarative knowledge in a method
analogous to a computer's interpreter program. Information presented to the learner (or interpreter) is assembled and deciphered in light of the given situation or problem state. Just as the process of mastering a new skill involves the mastery of procedures for conducting that skill, individuals may also possess skills (i.e., weak-method strategies) for learning skills.

The use of weak-method problem solving strategies by individuals in the declarative knowledge stage suggests two other characteristics of this stage. First, Kanfer and Ackerman (1989) point out that a hallmark of the declarative knowledge stage is a heightening of the attentional resource demands placed on the learner. Examples of the types of activities that place these demands on the learner include observation of others performing the task, encoding of task rules, and compilation of production rules (i.e., condition-action pairings) for task performance (Anderson, 1982; 1985; Kanfer & Ackerman, 1989).

A second by-product of these weak-method strategies is their slow, error-prone nature. Individuals engaged in the process of encoding new information are likely to make numerous errors in comprehension of the information, appropriate application of rules, and correct utilization of strategies for information encoding (Anderson, 1982). All of these factors are due to the "newness" of the task
for the learner. Further, encoding by definition places extensive demands on short-term memory. It is not until the skill has been proceduralized and stored in long-term memory that these demands will be eased.

To summarize, the flexibility and applicability of weak-method solutions offers a distinct advantage to the initial declarative knowledge stage. However, along with this flexibility come disadvantages associated with attentional demands and the application of error prone strategies. Once the information has become adequately encoded, there is a need to increase processing efficiency; this mechanism is knowledge compilation.

The Knowledge Compilation Stage

The dominant feature of the knowledge compilation stage is practice. Information stored in the first phase of skill acquisition is utilized in a more rapid and efficient manner (Anderson, 1982). According to Anderson (1982), there are three key mechanisms involved in this stage: an increase in performance speed, a decrease in verbal rehearsal of stored declarative knowledge, and a loss of point-to-point (or piecemeal) application of encoded information. Fisk and his colleagues (Fisk & Schneider, 1983; Fisk, Ackerman, and Schneider, 1987) have also noted increases in performance speed, as well as increased accuracy and transference of procedures from short- to long-term memory as key behavioral examples of
the knowledge compilation process. Another feature of this stage is a noted decrease in the attentional demands of the learner (Kanfer & Ackerman, 1989).

Anderson (1982) describes two key processes involved in this stage. Knowledge taken from the declarative stage may be processed either by composition or proceduralization. In composition, declarative information that is temporally related (i.e., information that occurs serially in time) is pieced together into sequences. Composition allows for increased processing speed since information is stored sequentially rather than as a series of distinct pieces of information. Proceduralization involves drawing on and storing the essential features of a piece of information as a representation of that knowledge (Anderson, 1982). Thus, while the full detail of the declarative knowledge may not be readily accessible, proceduralization allows the essence of the information to be captured thus permitting more rapid recall. As can be seen, the characteristic advantages of these two key processes differ. Composition speeds sequenced processing by decreasing the number of pieces of information that must be processed whereas proceduralization decreases memory workload (Anderson, 1982). In sum, these two processes, through extensive practice, form the beginnings of procedural knowledge.
The Procedural Knowledge Stage

The transition from knowledge compilation to procedural knowledge is characterized not so much in terms of quantity of information processing but more in terms of quality. Continuing with the computer analogy presented above, computers have programs which compile interpreted (or object) code into faster, accurate, domain-specific procedures. Similarly, the declarative knowledge stage serves to interpret (using weak method problem solving strategies) code (declarative knowledge) which is compiled (proceduralized) in the procedural knowledge phase (Gray & Orasanu, 1987). Coming out of the knowledge compilation phase, the learner has accrued information sufficient to perform the task rapidly. However, the hallmark of procedural knowledge lies in the accuracy of the information. Shiffrin and Schneider's (1977) landmark article on automatic and controlled information processing refers to the automaticity with which experts perform tasks. This feature of information processing is congruent with Anderson's (1982; 1985) conception of procedural knowledge. Experts not only retrieve information via automatic processes, but they also show a great deal of accuracy in the application of the retrieved information. Anderson (1982) and Rumelhart and Norman (1978) use the term tuning to describe the processes which portray the increases in accuracy found in this phase.
The tuning mechanism discussed by Anderson (1982) and Rumelhart and Norman (1978) is characterized by three key procedures. First, learners with procedural knowledge show a widening generalizability in the application of procedures. Second, these learners increase the power of the discrimination rules that guide the appropriate application of the procedures. That is, individuals with procedural knowledge simultaneously increase the span of applicability of their rules to different instances requiring their use, and make the criterion for when the rule is to be applied more rigorous. Consequent to the first two procedures, the process of adjusting the strengths of rules is a third mechanism involved in tuning. Through experience in applying procedural knowledge, learners at this stage adjust the strength of the procedural rules such that weaker rules are removed from memory thereby increasing processing efficiency.

As an example, let us consider bicycle riding again. My intent is to determine the necessity of the bicycle seat toward the skill of bicycle riding. Let us further assume that I have acquired the knowledge that I do not need a bicycle seat to ride my bicycle. As I practice with other people's bicycles, I develop a rule that states that bicycle seats are not required for bicycle riding regardless of the owner of the bicycle. Thus I have widened the applicability of my rule. To illustrate the
second component of the tuning mechanism, let us further suppose that in applying my rule I discover that seats are helpful to the extent that they provide some rest for my legs on long rides. Rather than weakening the strength of my rule or limiting the applicability of the rule, I would in all likelihood make the criterion for application of my rule more rigorous. I might say then that bicycles seats are not necessary for bicycle riding on short distances, thereby making the application criterion more rigorous without affecting the generalizability (i.e., the rule still applies to all bicycles) or strength of the rule itself. Finally, the strength component of the tuning mechanism may be increased through continued practice using different bicycles and different riding distances. The new rule will increase in strength with every successful application.

Summary

The review of cognitive learning theory provided above has been tailored toward the literature focusing on knowledge acquisition. For a more comprehensive review of other elements in cognitive learning theory, I direct you to the reviews of Weiss (1990) or Glaser and Bassock (1989). The intent of this section has been to provide the reader with a basis in the key learning variables for the present study: declarative and procedural knowledge.
Social Learning Theory

The area of training in psychology is often characterized as paradoxical in nature (Muchinsky, 1990). The field is seen as critically important for success in organizations (Baldwin & Ford, 1988) yet lacking in adequate theoretical and empirical research (Campbell, 1971). Recent works have contributed greatly toward improving the organization and structure of the training field (e.g., Goldstein, 1989; 1993), yet there remains a marked lack of definitive research from which practitioners may draw upon in implementing training programs.

The sub-area within training that is of interest in the present study concerns the methodology used to train employees. Goldstein (1993) discusses numerous methods ranging from the traditional lecture and on-the-job methods to more modern forms such as programmed instruction, computer-aided instruction, simulations, and behavior modeling training. According to Goldstein (1993), it is the latter method, behavior modeling training, that has received the most research attention in recent years. Latham (1989) offers the interactionist perspective inherent in social learning theory (and hence in behavior modeling training) as the primary reason for the popularity of this approach. By incorporating cognition and behavior as well as the moderating influence
of the environment, Bandura's (1986) social learning theory is able to provide an approach that is accessible to a broad spectrum of researchers. Below I provide an overview of social learning theory, the basis for behavior modeling training, followed by a discussion of the particular aspects of the methodology itself. The concluding portion of this review highlights those aspects of behavior modeling training that are particularly relevant to the present study.

Social Learning Theory

Social learning theory emerged out of attempts to incorporate imitative behavior within a conditioning paradigm (Miller & Dollard, 1941). Following these attempts, the landmark research of Bandura and Walters (1963) shifted the theoretical focus of such research toward the phenomenon of observational learning (or modeling). However, as Weiss (1990) points out, the characteristics of observational learning in the Bandurian approach are markedly more cognitive in focus than other, more traditional, reinforcement-oriented views (e.g., Mowrer, 1950). This cognitive focus weighs heavily in the development of the present study.

Bandura (1986) makes two key delineations of the modeling construct which are instructive. First, the result of observational learning must be the development of a new knowledge structure. Instances of instinctive
reaction, behavioral contagion (e.g., yawning), social facilitation, and social loafing are not considered to be within the scope of observational learning (Weiss, 1990). Second, and most important, modeling represents a distinctively different mode of learning from that of direct experience. Although direct experience (also called enactive learning) is a viable form of learning (to the extent that the outcomes attained are roughly comparable to outcomes attained via other methods of learning), Bandura (1986) believes that enactive learning suffers greatly in terms of the efficiency of information processing it offers. Bandura asserts that such trial-and-error learning is inferior to the process of observing others' behavior (and the resultant outcomes of that behavior) which forms the impetus for modeling.

With these distinctions in mind, Bandura's theoretical approach may be outlined as the combination of four interrelated sub-processes: attention, retention, production (or reproduction), and motivation (Bandura, 1986). The first of these processes, attention, is prerequisite to the onset of modeling. Bandura (1986) outlines several variables that influence the degree of attention the observer will expend toward the observation of a behavior. These variables can be divided into four categories. First, properties of the behavior that is to be modeled such as discriminating ability, salience, and
behavioral complexity influence attention. Second, valences and expectations for the occurrence of future behavior that are attached by the observer can influence attention. Similarly, individual differences of the observer can play a role in attention. For example, Weiss (1978; 1990) notes that the extent to which one is an active or passive observer is largely influenced by the motivating effect of low self-esteem. I note that attitudes toward computers can be construed as an example of an individual differences variable that may influence attention in computer training. The hypothesized role of these attitudes in the context of behavior modeling training will be discussed below. Characteristics of the situation are a final variable that influences attention. Situations may act to constrain the amount of attention the observer may allocate to the observation of behavior.

Given that the variables discussed above enable the individual to sufficiently attend to a behavior, the second key process in social learning theory involves the retention of observed behavior in memory. This stage borrows heavily from the cognitive realm. Bandura (1986) invokes the constructs of symbolic coding, categorization, and rehearsal as the operative processes in retaining modeled behavior. Further, in keeping with the interactionist's perspective, retention is influenced by the cognitive skills and structures of the individual.
Following retention, the observer of modeling must make a transition from attended to and symbolically encoded behavioral observation to the production of behavior. In what is perhaps the most critical process involved in social learning theory, the production phase forms the critical cross-over from cognition to behavior (Weiss, 1990). Four key sub-processes drive the transition from cognition (encoded observation) to behavior (enactment). First, the individual must have an accurate cognitive representation of the methods of combination and sequencing of behavior which forms the task. Subsequently, the individual must observe behavioral enactments of this representation to adequately classify the desired behavior. The degree of fit between these behavioral enactments and the individual's cognitive representation is consequently gauged by feedback information from the environment. The result of feedback-moderated comparisons between cognitive depictions and behavioral enactments forms the final critical process: conception matching. The process of conception matching is in turn moderated by individual characteristics such as physical abilities and skills (Bandura, 1986).

The successful completion of the first three stages in Bandura's theory insures that learning has taken place. However, the distinction between learning a behavior and the actual performance of that behavior invokes the need
for the fourth process in social learning theory: motivation. Bandura (1986) uses the synonymous term "incentive" to characterize those aspects which motivate the modeling observer to enact learned behavior. The most widely recognized of these incentives is the phenomenon of vicarious experience which has repeatedly been shown to increase modeled behavior (Weiss, 1990). Other incentives, Bandura argues, may emerge from external sources (e.g., social incentives, sensory input incentives) as well as internal sources (e.g., self-evaluation). As is the case with all of Bandura's modeling processes, the motivation phase is moderated by individual differences. The most notable of these differences is the self-efficacy construct which dictates the amount of effort an individual will exert toward a given task, and the amount of persistence in the face of adversity the individual will endure (Bandura, 1977). Bandura (1986) suggests that individual differences such as self-efficacy may dictate the learner's preferences for particular types of incentives, internal standards, and social comparison biases.

To summarize Bandura's social learning theory, individuals engage in four temporally ordered processes in learning and performance. An observer must first attend to the presented stimulus, then properly encode in memory the essential elements of the observed behavior.
Subsequently, the individual must transfer the encoded information from memory to action. Finally, the likelihood of enacted learned behavior is dependent upon the degree of motivation the observer has to enact the behavior.

Behavior Modeling Training

The learning method associated with and developed from social learning theory has been termed behavior modeling training. Original research using behavior modeling techniques emerged most frequently in the treatment of phobic individuals (e.g., Bandura, 1977). Latham (1989) points out that widespread application of behavior modeling training in organizations has been limited primarily to the area of leadership skills. Goldstein and Sorcher's (1974) text on leadership is an example of this focus which also represents one of the first applications of Bandura's social learning theory to training in organizations.

Research on behavior modeling training as a technique typically uses the four major steps of social learning theory (attention, retention, production, and motivation) discussed above as a framework for the methodology. Decker and Nathan (1985) provide an excellent text for the implementation of behavior modeling training programs. As an example of a typical behavior modeling methodology, I offer the supervisor training research of Latham and Saari
Individuals in the Latham and Saari study were first introduced to the topic by the trainers. Here the attentional processes of the trainees are activated in concurrence with the first step in Bandura's theory. The second phase in social learning theory, retention processes, was activated by having individuals observe a model demonstrating key learning points. In addition to the behavioral observation, the learning points were presented to the trainees both before and after the model presentation. Following the observation of the model, trainees met as a group to discuss the effectiveness of the model thereby further encouraging retention. Following this phase, trainees engaged in role-playing activities which served the dual purpose of further retention and motor production of the modeled behavior. The final step in the training program involved providing trainee feedback on performance of the trained behaviors. This step served as the motivation phase in Bandura's theory. Results from the Latham and Saari (1979) training program were evaluated using Kirkpatrick's (1976) levels of training criteria which evaluate studies based on: trainee reaction to the training, degree to which the trained material was learned, transfer of learned material to performance on the job, and value of the training program to the organization as a whole. Latham and Saari
(1979) found evidence in favor of behavior modeling training at all criterion levels.

In general, support for the effectiveness of behavior modeling training in organizations has been widespread. The research of Meyer and Raich (1983) found full support for the effectiveness of behavior modeling training. Russel, Wexley, and Hunter (1984) found that behavior modeling training elicited positive reactions from trainees and proved to be an effective method in terms of learning criteria. However, results failed to find support for the transfer of learned behavior to performance on the job. Encouragement for the training methodology is evidenced in the meta-analytic research of Burke and Day (1986) who found behavior modeling training to be an effective means of training employees across a wide array of organizations and situations.

Other researchers have examined the various components of behavior modeling training in an attempt to identify what aspects make the methodology effective. The research of Decker (1980; 1982; 1983; 1984) identified behavioral rehearsal and social reinforcement as components that could enhance the already effective process of modeling. Further, Hogan, Hakel, and Decker (1986) found that trainee rule code generation (as opposed to trainer provided rules) enhanced subject’s learning of modeled behavior. Finally, Gist and her colleagues have
amassed several empirical and theoretical examinations of the role of self-efficacy in behavior modeling training (Gist, 1987; 1989; Gist, Stevens, & Bavetta, 1991; Gist and Mitchell, 1992). Results from this research have indicated that an individual's perceived self-efficacy is enhanced through the use of modeling interventions.

To summarize the research and theory on social learning, one can say with a certain degree of confidence that the principles put forth by Bandura have received extensive support. Within the field of industrial psychology, research examining the components of social learning theory continues unabated. The subsequent section follows this vein in an attempt to integrate social learning phenomena with cognitive learning theory.
Integrating Cognitive and Social Learning Theory

Based on the reviews of cognitive and social learning theory presented above, it would seem apparent that the two theories have a considerable amount in common. Not only are both theories focused on learning, but they also have cognitive factors as the central aspects of their respective approaches. Therefore, one might expect that distinctions such as procedural and declarative knowledge would play a part in social learning theory. Conversely, one might also expect that researchers in knowledge acquisition would include the observation of modeled behavior as important to their approach. However, this theoretical integration has not occurred in the literature thus far.

The cognitive learning area appears to center its focus around micro-level issues in learning (e.g., expert-novice problem solving strategies). Further, Tversky (1982) noted that cognitive investigators moved the focus away from holistic approaches of learning toward more memory based examinations. Consequently, cognitive learning researchers tend to downplay issues related to which methods yield the greatest gains in learning. Although Tversky (1982) makes the observation that a re-emergent interest in more traditional learning areas is occurring, an examination of recent reviews in learning (e.g., Weiss, 1990) indicates that segregation of
cognitive and social learning theories still occurs. As a whole then, cognitive learning research may be seen as predominantly centered on micro-focal issues in learning.

Kanfer and Ackerman's (1989) work on knowledge acquisition in industrial training marks the rare exception of research which examines cognitive learning variables with varied training methodologies. Interestingly, Kanfer and Ackerman (1989) note that skill acquisition in the initial declarative knowledge phase is typically aided by techniques such as the specification of task objectives, instruction on the task, observation of others, encoding of task rules, and development of strategies for task performance. Thus, although research in cognitive learning theory does not explicitly examine training techniques, there is a noted parallel between the techniques used to encourage knowledge acquisition and the procedures underlying behavior modeling training. It is this parallel that marks the focus of my integration.

In my review of behavior modeling training, I was surprised to note Bandura's position on cognitive learning theory. According to Bandura (1986), "(c)onstruing learning in terms of factual and procedural knowledge is well suited for cognitive problem solving. But there are many domains of activity that require additional mechanisms to get from knowledge structures to proficient
action" (p. 107). While Bandura does capture the micro-
focal nature of much of cognitive learning theory, his
position belies this statement to a considerable extent.
Below I provide several examples which demonstrate how
Bandura’s (1986) social-cognitive approach overlaps with
key aspects of Anderson’s (1982) skill acquisition
research.

The first parallel that may be drawn between the two
learning theories concerns how knowledge and skill are
related. Bandura argues that knowledge and cognitive
skills are necessary but insufficient prerequisites for
performance skill. That is, knowing what something is
(declarative knowledge) and knowing how to do something
(procedural knowledge) do not guarantee that one can in
fact perform. However, Bandura (1986) notes that the
acquisition of performance skill requires a "conception-
matching mechanism" which, using information attained from
"physical enactment" (i.e., task practice), guides the
transfer from knowledge to performance. The term
"physical enactment" has clear parallels in cognitive
learning theory’s second phase of skill acquisition.
Recall that once individuals acquire declarative knowledge
about a task, there is a period of intensive practice or
rehearsal known as the knowledge compilation phase.
Therefore, Anderson (1982) would agree with Bandura (1986)
in stating that the acquisition of performance skill
requires extensive practice with the task itself. To call on an adage: "practice makes perfect."

A second parallel between the two theories has already been alluded to: Bandura's (1986) conception-matching mechanism is in many ways similar to Anderson's (1982) production system-based tuning mechanism. In both approaches, the key element involves using acquired knowledge coupled with practice at a skill to formulate rules of action. Bandura (1986) claims that the essence of learning lies in obtaining generalizable rules which can be applied to novel instances. This method of learning is markedly similar to the processes underlying procedural knowledge. Recall that Anderson's (1982) tuning mechanism, which guides the acquisition of procedural knowledge, serves to aid in the creation of generalizable or widely applicable rules. Both theories suggest that a singular mechanism drives the formation of generalizable rules which are the quintessence of learning. Further, Bandura states that judgments require reasoning from propositional knowledge. Cast in light of cognitive learning theory, judgments require the integration of condition-action rules (i.e., propositions). Bandura (1986) also claims that the conceptions involved in conception-matching (the process of transferring knowledge to skill) are rules of action which provide guidelines for skilled performance. The
author further notes that procedural rules are required for skilled performance. In sum, it is the opinion of both Bandura and Anderson that the path to skilled performance lies in the acquisition of procedural knowledge.

A third and final tie between cognitive and social learning theory concerns a variety of modes of learning and common mediating cognitive processes. These modes of learning include modeling, conditioning, verbal instruction, and enactive exploration. Bandura (1986) asserts that each of the modes are characterized by how they convey information about rules of action. Further, the author notes that, whereas the various forms of learning differ in the information they present to the perceiver, they are all mediated by the process of conception-matching.

In a similar vein, Anderson (1982) uses the term "weak-method problem solving" to refer to the common mediating processes by which novices approach problems. Novices encounter information from a variety of sources all of which receive the same strategic application of problem solving methods. Kanfer and Ackerman (1989) indicate that individuals in the process of acquiring declarative knowledge frequently encounter modes of learning such as task instruction, observation of demonstrations, strategy development, and task rule
encoding and storage. The acquisition of knowledge from these modes of learning is mediated by weak-method problem solving strategies. Clearly, these modes of learning, borrowed from cognitive learning theory, have strong parallels with social learning theory's learning modes. While weak-method problem solving strategies do not directly correspond to conception-matching mechanisms, the gist of the argument lies in proposing that both theories allow for the input of several modes of learning which are treated by the same central processor.

The three parallels presented above suggest a possible integration of the typically micro-focal cognitive learning theory with the more applicable social learning theory. In essence, the processes which depict how learning occurs (i.e., via the generation of rules) appear congruous, the mechanisms which guide skilled performance appear to be similar, and the approaches to the attainment of performance coincide with each other. One of the primary objectives of the present study is to ascertain whether the concepts from these two theories may be integrated in the same design.
Relations Between Computer Knowledge and Performance

On several occasions I have mentioned the link between task knowledge and task performance. Obviously, one can expect that some degree of knowledge about the fundamental components of the task will aid in task performance. The relationship between knowing and doing (i.e., performing) is clear when one considers the straightforward and flexible (i.e., widely applicable) concept of declarative knowledge. Here I expected that knowledge of facts about the task (declarative knowledge) would logically relate to task performance. However, I also realize that the relationship between declarative knowledge and performance is far from perfect. Turning again to the bicycle riding example, my knowledge of the names of the different parts of a bicycle and a knowledge of their purpose does not guarantee that I know how to ride bicycles. Therefore, I expected a moderate effect for the declarative knowledge to performance relationship. My first hypothesis suggests that:

**Hypothesis 1a:** The relationship between declarative knowledge and task performance will be significant, positive, and moderate in effect size.

Turning to procedural knowledge, I expected the link between knowing how to perform a task and actual
performance to be exceedingly strong. This relationship is so strong that researchers in the cognitive realm have essentially equated the two. That is, procedural knowledge is typically measured through performance. It is my belief that, while highly related, procedural knowledge and performance are not identical. While it is true that one who possesses procedural knowledge for a skill can not necessarily perform the skill, the converse is more than likely false. With rare exception, an individual who can perform a skill must obviously know how that skill is performed.

One example of how procedural knowledge and performance might not correspond involves the implicit nature of procedural knowledge. Individuals with procedural knowledge may be unable to adequately describe their task performance (i.e., produce their existent internal procedural knowledge in an external fashion). The problem appears to lie in relating an internal event (task knowledge) with an external event (task performance). From this we may conclude that one source of error in the relationship between a procedural knowledge test and a measure of task performance is measurement error. That is, the lack of a perfect relation between the two variables is due in part to a lack of reliability in the measurement of internal procedural knowledge.
A second rationale for the high relationship between procedural knowledge and task performance is the already discussed novice-expert distinction in knowledge acquisition. Individuals in the declarative knowledge phase have performance that is slow and error-prone (Glaser & Bassok, 1989; Gray & Orasanu, 1987; Kanfer & Ackerman, 1989). Conversely, individuals in the procedural knowledge phase have, by virtue of their experience, greater performance in terms of both speed and accuracy (Anderson, 1982).

To summarize, procedural knowledge tests and measures of performance may be construed as two imperfect measures of the same construct. A procedural knowledge test cannot adequately address the implicit aspects of the procedural knowledge construct whereas performance can. On the other hand, performance assesses procedural knowledge by examining outcomes rather than directly measuring knowledge of how tasks are performed. Therefore, I predicted that:

**Hypothesis 1b:** The relationship between procedural knowledge and task performance will be significant, positive, and strong (large effect size).

Given the temporal nature of the two knowledge variables (i.e., declarative knowledge precedes procedural
knowledge) as well as the evidence presented above for the superiority of procedural knowledge in terms of its relation to performance, the third part of my first hypothesis followed:

**Hypothesis 1c**: The relationship between procedural knowledge and task performance will be significantly greater than the relationship between declarative knowledge and task performance.
Knowledge Acquisition and Computer Training

Recall that on the outset of the present effort I endeavored to establish a theoretical base for research in computer technology training. Further, I noted that my interest lies in understanding training with computers for the sake of computer learning as well as for an understanding of theoretical issues in learning. It is my attempt to establish some evidence for both.

I should point out that the use of computer training as the vehicle for examining research questions on technology in general, and cognitive and social learning theory specifically, is purposeful. Even though computers are the most pervasive of technologies in the workplace, there remains a dearth of knowledge on how individuals learn how to use computers. The appropriateness of computers as a tool to examine the two learning theories presented above is suggested by two lines of evidence. First, empirical evidence (presented below) as well as personal experience has identified modeling as important to an individual's understanding of how computers work.

Second, the distinction between declarative and procedural knowledge (see above) and the measurement of the two constructs is much clearer when one considers computer technology. Declarative knowledge of computers is exacting and necessary; there are numerous facts one must know to successfully operate computers. Further,
while procedural knowledge is typically difficult to describe (e.g., how does one swing a golf club so that the ball travels far and straight), this is not the case when one considers computers. Judging on past experience, I find knowledge of how to perform tasks on the computer to be highly explicit in nature. The distinctiveness of computer knowledge has the added benefit of increasing the correspondence between procedural knowledge and performance. Therefore, computer technology can be seen as an appropriate centerpiece for research on both cognitive and social learning theory. This section lays out the existent research related to computer technology training. The theoretical integration of the two learning theories is interwoven within this discussion.

Computer Training

Research has shown that the findings implicating behavior modeling training as a superior training methodology (e.g., Burke & Day, 1986; Latham & Saari, 1979; Meyer & Raich, 1983; Russel et al., 1984) hold true for computer training as well. Two studies conducted by Gist and her colleagues (Gist, Rosen, & Schwoerer, 1988; Gist, Schwoerer, & Rosen, 1989) have indicated that behavior modeling training significantly improves performance on computer software mastery over traditional computer tutorial approaches.
In the Gist et al. (1988; 1989) studies, behavior modeling training consisted of three steps. First, trainees were provided with key learning points to facilitate retention and rule code generation (Decker, 1982). Second, trainees observed a videotaped model demonstrating the behaviors required for task performance. Finally, trainees were given the opportunity to imitate the model's behavior through a series of practice sessions (i.e., enactive mastery). In contrast to this approach, the tutorial method consisted of a form of computer-aided instruction. The purpose of computer-aided instruction is to provide illustrative examples, programmed instruction, repetitive practice, and subsequent feedback on performance all through the computer medium (Goldstein, 1993). Tutorial trainees and modeling trainees received identical information, examples, and practice time. The only difference between the groups was that the modeling condition would pause in the process of the training session to observe the model engaging in the behaviors of the task at hand. Thus the tutorial method is essentially a lecture method provided by computers (i.e., a "tell" approach) whereas the modeling method is a lecture method provided by computers plus a videotaped model (i.e., a "tell and show" approach).

The second set of hypotheses in the present study modify and extend previous evidence for the superiority of
behavior modeling training in computer tasks provided by Gist et al. (1988; 1989). Following the evidence provided above, I expected behavior modeling training to significantly improve knowledge of computer tasks over other methods. Note that the modification of previous research lies in the distinction I made between knowledge and performance. Whereas improvements in task performance implicate increases in task knowledge, task performance does not directly assess task knowledge (see the discussion on pp. 36-37). Further distinguishing this hypothesis was the measurement of two types of knowledge. In addition to indirectly measuring procedural knowledge through task performance, I directly measured both declarative and procedural knowledge.

The extension of the previous research lies in the more thorough examination I offered of the components of modeling training. In addition to examining groups similar to the two used in the Gist et al. (1988; 1989) studies (i.e., lecture-modeling-practice, and lecture-practice), I added two groups to the study: 1) a no lecture group which only received practice on the task, and 2) a modeling-only group which received modeling along with task practice but did not receive a lecture. The addition of these two groups provided me with the opportunity to examine the lecture and modeling components of social learning theory in a fully crossed design. That
is, both modeling and lecture methods may be examined with and without the benefit of their counterparts.

Knowledge Types, Modeling Training, and Lecture Training

Assuming that subjects are randomly selected, there should be no differences between the four training conditions (modeling plus lecture, modeling only, lecture only, and no lecture) in the pre-test assessment of knowledge. Therefore, all the hypotheses in the following section should reflect significant gains in the two types of knowledge (i.e., interactions).

In the context of the present study, the presentation of key learning points can be seen as an effort to transfer declarative knowledge to the trainees. That is, key learning points are the facts about the particular tasks. Therefore, one would expect lecture trainees to show increases in declarative knowledge as a result of training. Further, the observation of a model allows for a less overt yet still meaningful reception of facts about the task. As such, I also expected modeling trainees to possess greater declarative knowledge at post-test. Thus I predicted that:

**Hypothesis 2a:** Trainees receiving behavior modeling training will gain significantly greater declarative knowledge than trainees receiving no modeling.
Hypothesis 2b: Trainees receiving a lecture will gain significantly greater declarative knowledge than trainees receiving no lecture.

Recall from the integration of cognitive and social learning theory that the processes which lead to learning in the Bandurian sense have been shown to be similar to the processes which effect increases in procedural knowledge. Further, procedural knowledge is acquired through practice with the application of declarative knowledge (Anderson, 1982). Therefore, one would expect all training conditions to show some increases in procedural knowledge as a result of task practice. Trainees who receive more declarative knowledge (i.e., through the lecture), however, should have more procedural knowledge since the latter is generated from the former (Anderson, 1982); the more fuel one has (declarative knowledge), the more fire (procedural knowledge) that can be created.

To further differentiate the training conditions, one would expect that any interventions that facilitate trainee knowledge of how the task is performed (i.e., behavior modeling training) would necessarily aid in the acquisition of procedural knowledge. Evidence for the role of behavior modeling training in improving learning criteria and performance over other methods has already
been mentioned (e.g., Latham & Saari, 1979). Further, Gist et al. (1988; 1989) provide evidence that behavior modeling training leads to increased knowledge of how to perform tasks (as measured by task performance). Finally, Best (1989) points out that "procedural knowledge... is more easily shown to someone than it is told" (p. 7). Behavior modeling training clearly advocates showing (modeling) as superior to simply telling. Therefore by virtue of observing models applying their declarative knowledge to specific tasks, one would expect increases in procedural knowledge that are greater than increases solely on the basis of task practice with declarative knowledge. As such, the three components of my third hypothesis were:

**Hypothesis 3a:** Trainees receiving behavior modeling training will gain more procedural knowledge than trainees receiving no modeling.

**Hypothesis 3b:** Trainees who receive a lecture will gain greater procedural knowledge than trainees receiving no lecture.

**Hypothesis 3c:** Trainees receiving behavior modeling training (with or without a lecture) will have greater gains in procedural knowledge than trainees receiving only a lecture.
Computer Attitudes and Computer Training

As mentioned above, the two major research bases encountered in the computer literature are training and attitudes. The goal of the present section is to bring the attitude research into the fold of computer training. Previous work which addresses attitudes within a training context tends to focus on attitudes toward the training program (Goldstein, 1993). This research seeks to use attitudes as a means of determining the quality of the training program (i.e., the reaction criteria; Kirkpatrick, 1976). Further, even within the relatively small computer training literature, studies incorporating attitudes tend to examine reactions to the training program (e.g., Gist et al., 1989) rather than reactions to the training content per se. In contrast to this approach, my interest was in examining changes in attitudes as a function of training and the impact of a priori attitudes on training outcomes.

Effects of Computer Training on Computer Attitudes

Parsons (1988) indicates that training in technology should, like other training programs, begin with an assessment of the needs of the trainees. Further, the author suggests that, in addition to skills training, the attitudes of the individual toward technology are an important factor to consider. In reviewing the literature on attitudes toward computers, I found that individuals'
beliefs concerning computers tend to fall along two distinct dimensions: 1) the belief that the computer is a beneficial tool to be used by humans, and 2) the belief that the computer is an autonomous entity capable of replacing humans at work (Lee, 1970; Zoltan & Chapanis, 1982; Rafaeli, 1986). Empirical applications of this bi-dimensional perspective exist (Coovert & Goldstein, 1980; Kerber, 1983; Brock, 1991). Further, recent research using a confirmatory factor analytic framework has indicated support for the convergent and discriminant validity of the two constructs (Brock & Sulsky, in press).

It is interesting to note that, despite the continued evidence that individuals have two distinct reactions to computers, little research has been conducted to directly examine the development and change of computer attitudes. Rafaeli (1986) and Brock and Sulsky (in press) found computer use to be a significant predictor of both beneficial tool beliefs and autonomous entity beliefs. Additionally, research has indicated that computer avoidance reactions (e.g., phobias, anxieties, and fears), which are related to autonomous entity beliefs, are also significantly negatively related to computer use (Hudiberg, 1989; Nickell & Pinto, 1986; Rosen, Sears, & Weil, 1987; Zakrajsek, Waters, Popovich, Craft, & Hampton, 1990).
Regarding the proposal that beneficial tool beliefs result from using computers, Hill, Smith, and Mann (1987) found that previous experience with computers, when mediated by efficacy beliefs, was the best predictor of decisions to use computers in the future. Results from their study suggest that the important determinant of future computer use was not experience alone, but the beliefs individuals brought with them from the experience. Brock (1991) found efficacy beliefs (using the same measure as Hill et al., 1987) to be strongly related to beliefs in the computer as a beneficial tool. Thus, indirect support exists to suggest that beliefs about the computer are influenced by computer experience.

Recent research by Martocchio (1992) suggests a link between attitudes toward computers and computer training. Using Dutton and Jackson's (1987) framework for decision making, Martocchio (1992) outlined reactions to computer usage in terms of threats and opportunities. Threats and opportunities are similar to autonomous entity beliefs and beneficial tool beliefs, respectively. That is, believing that the computer is an autonomous entity implies that the computer is somehow threatening; believing that the computer is a beneficial tool intimates a conviction that the computer may provide the individual with opportunities in the future. Results from Martocchio's (1992) research indicated that computer training significantly reduced
computer anxieties and increased efficacy beliefs.
Further, these findings were moderated by perceptions of
the computer such that individuals perceiving computer
usage as an opportunity evinced significantly more change
in anxiety and efficacy than individuals who perceived
computer use to be a threat.

Based on the evidence provided by Martocchio (1992)
as well as the research indicating that experience with
computers relates to attitudes toward computers (Hill et
al., 1987; Rafaeli, 1986; Brock, 1991), I proposed that
computer training increases trainee’s beliefs in the
computer as a beneficial tool while simultaneously
tempering trainee autonomous entity beliefs. My first
hypotheses linking computer attitudes and computer
training follow:

**Hypothesis 4a:** Trainees will experience an
increase in beneficial tool beliefs as a result
of practice and training programs.

**Hypothesis 4b:** Trainees will experience a
decrease in autonomous entity beliefs as a
result of practice and training programs.

The preceding hypotheses argue that exposure to
computers (through training) alters an individual’s
attitudes. Following from this line of thought, I will
endeavor to demonstrate how behavior modeling training may
influence attitudes more so than other training forms. Two lines of evidence support the argument that individuals in behavior modeling training can expect more attitude change: one relates to beneficial tool (or positive) attitudes toward computers and the other relates to autonomous entity (or negative) attitudes toward computers.

Regarding beneficial tool attitudes toward computers, my research has indicated a strong association between beliefs in the computer as a beneficial tool and computer self-efficacy (Brock, 1991). Within a consistency theory framework (e.g., Bem, 1972), it is easy to see how increased beliefs in one's ability to perform (i.e., self-efficacy) lead to more positive views of the object or task one is performing upon. That is, to remain consistent, computer efficacious individuals will be more likely to hold positive views toward computers. Continuing with this line of thought, evidence suggests that behavior modeling training is one of the most effective methods for the enhancement of self-efficacy (Bandura & Adams, 1977; Bandura, Adams, & Beyer, 1977). This evidence suggests that individuals who receive modeling will have more beneficial tool beliefs than individuals not receiving modeling training. The first portion of my fifth hypothesis follows:
Hypothesis 5a: Trainees receiving modeling will experience greater increases in beneficial tool attitudes than trainees who do not receive modeling.

Turning to the effects of behavior modeling training on negative or autonomous entity beliefs about computers, the strongest support for this hypothesis emerges from the already noted association between autonomous entity beliefs and computer avoidance reactions (see p. 47). Beliefs in the computer as an autonomous entity capable of supplanting humans are related to computer fears and anxieties both empirically (Hudiberg, 1989; Nickell & Pinto, 1986; Rosen, et al., 1987; Zakrajsek, et al., 1990) and logically (i.e, the idea of an "autonomous entity" evokes feelings of anxiety). The idea that behavior modeling training can mitigate these fears more so than other training methods has support in the social learning theory literature. Indeed, the empirical basis for social learning theory emerged out of clinical research on the treatment of snake phobias which found behavior modeling training to be an effective means of alleviating these fears (Bandura, Blanchard, & Ritter, 1969). Transferring this research to the computer domain, I hypothesized that fear, anxiety, and avoidance of computers (as measured by autonomous entity attitudes toward computers) would be
mitigated most readily in modeling approaches. The latter portion of my fifth hypothesis follows:

**Hypothesis 5b:** Trainees receiving modeling will experience greater decreases in autonomous entity attitudes than trainees who do not receive modeling.

**Computer Attitudinal Effects on Learning**

The previous discussion centered on how the experience of training in general, and behavior modeling training in specific, might influence attitudinal change. The final hypothesis in this study examined how computer attitudes might influence computer learning. Research suggests that highly accessible attitudes influence an individual's interpretation of and response to a situation (Fazio, 1986; 1989). Clearly in this research, attitudes toward computers can be seen as salient given the context (i.e., computer training), and these attitudes are likely to be easily accessed (e.g., Shiffrin & Schneider, 1977). Therefore, theoretical support for the idea that attitudes toward computers can influence computer behavior exists.

As mentioned above (see pp. 21-22), two key elements of behavior modeling training are the amount of attention devoted to the task and the amount of motivation to perform the task (Bandura, 1986). Further, both task attention and motivation to learn are influenced by the
individual's feelings about the task content. Since the focus of this effort is on learning how to use computers, attitudes toward computers can be construed as an important and motivating individual differences variable in training.

In the cognitive learning framework, one of the hallmarks of the declarative knowledge stage is heightened attention toward the task (Kanfer & Ackerman, 1989). Therefore, attitudes toward computers may influence learning by affecting attention and hence the amount of declarative knowledge acquired by the trainee. Motivation toward task performance also plays a key role in learning. Since procedural knowledge is necessary for skilled performance, it seems logical to expect that factors that influence task performance (i.e., motivation) will necessarily influence procedural knowledge. Thus I also expect increases in learning through the effects of increased motivation toward the task. Consequently, the influence of attitudes toward computers on motivation may play a role in the acquisition of procedural knowledge.

Empirical support for this position lies in Martocchio's (1992) research which found that trainees who perceived computer usage as an opportunity evinced significantly greater learning than trainees who perceived computer usage to be a threat. Thus I expect that opportunity-minded individuals (i.e., trainees who
perceive the computer to be a beneficial tool) will acquire more computer knowledge. Conversely, trainees who perceive computers as a threat (i.e., an autonomous entity) will likely acquire less computer knowledge. Therefore, the first two components of my final hypothesis assert that:

**Hypothesis 6a:** Beliefs that the computer is a beneficial tool will be associated with higher levels of computer knowledge acquisition.

**Hypothesis 6b:** Beliefs that the computer is an autonomous entity will be associated with lower levels of computer knowledge acquisition.

In addition to the separate influence of beneficial tool and autonomous entity attitudes on learning, I expect that beneficial tool attitudes will moderate the effects of autonomous entity attitudes on learning. Unfortunately, Martocchio's (1992) research design was such that it did not allow for the determination of an interaction between perceptions of threats and opportunities on training outcomes. Further, other research using beneficial tool and autonomous entity beliefs in the same design with computer use has treated the attitude constructs as dependent variables thereby making the identification of an interaction impossible (Rafaeli, 1986).
My own research (Brock, 1993) suggests that the two computer attitudes interact in their effect on computer use. As can be seen in Figure 1, the form of the interaction is such that individuals who possess few beneficial tool attitudes are unlikely to use computers regardless of their autonomous entity beliefs. Further, when beliefs in the computer as a beneficial tool are high, computer use is also high. However, the combination of high beneficial tool beliefs with low autonomous entity beliefs is associated with higher amounts of self-reported computer use than high scores on both attitude dimensions separately. Therefore, indirect support for an attitude interaction on learning exists. Since it is logical to expect that the more one uses computers, the more one will learn, I expect that, by inference to computer use, the two computer attitudes will interact on computer learning as well. Therefore, my final hypothesis predicts that:

**Hypothesis 6c:** Beliefs that the computer is a beneficial tool and that the computer is an autonomous entity will interact in their effects on computer knowledge acquisition. Autonomous entity beliefs will be negatively correlated with knowledge acquisition when beneficial tool beliefs are high, but unrelated when beneficial tool beliefs are low (see Figure 1 for an example of the form of the interaction).
The present study examines: 1) how concepts from cognitive and social learning theory may be integrated in the same training framework, and 2) how this relationship operates within the context of technology as operationalized by computers.

The first hypothesis in the study examines relations between declarative knowledge, procedural knowledge, and
performance. Here I expect both forms of knowledge to be predictive of computer performance. However, procedural knowledge should have a significantly greater relationship to task performance.

Regarding the integration of cognitive and social learning theory within the context of technology, I have hypothesized effects for both computer training and attitudes toward computers. My second and third hypotheses examine the effects of training methodology on two types of knowledge acquired by trainees. Lecture and modeling interventions are hypothesized to produce significant increases in declarative knowledge whereas behavior modeling training is expected to be a superior methodology for the acquisition of procedural knowledge.

The final three hypotheses in the study explore the relation between attitudes toward computers, training methodology, and knowledge acquisition. The fourth and fifth hypotheses examine how the training experience and, specifically, the training methodology influence attitudes toward computers. The final hypothesis examines the role attitudes toward computers have in predicting knowledge acquisition. A summary of all the hypotheses for the study follows:

**Hypothesis 1a:** The relationship between declarative knowledge and task performance is significant, positive, and moderate in effect size.
**Hypothesis 1b:** The relationship between procedural knowledge and task performance is significant, positive, and strong (large effect size).

**Hypothesis 1c:** The relationship between procedural knowledge and task performance is significantly greater than the relationship between declarative knowledge and task performance.

**Hypothesis 2a:** Trainees receiving behavior modeling training gain significantly greater declarative knowledge than trainees receiving no modeling.

**Hypothesis 2b:** Trainees receiving a lecture gain significantly greater declarative knowledge than trainees receiving no lecture.

**Hypothesis 3a:** Trainees receiving behavior modeling training gain more procedural knowledge than trainees receiving no modeling.

**Hypothesis 3b:** Trainees who receive a lecture gain greater procedural knowledge than trainees receiving no lecture.

**Hypothesis 3c:** Trainees receiving behavior modeling training (with or without a lecture) have greater gains in procedural knowledge than trainees receiving only a lecture.

**Hypothesis 4a:** Trainees experience an increase in beneficial tool beliefs as a result of practice and training programs.

**Hypothesis 4b:** Trainees experience a decrease in autonomous entity beliefs as a result of practice and training programs.

**Hypothesis 5a:** Trainees receiving modeling experience greater increases in beneficial tool attitudes than trainees who do not receive modeling.

**Hypothesis 5b:** Trainees receiving modeling experience greater decreases in autonomous entity attitudes than trainees who do not receive modeling.
Hypothesis 6a: Beliefs that the computer is a beneficial tool are associated with higher levels of computer knowledge acquisition.

Hypothesis 6b: Beliefs that the computer is an autonomous entity are associated with lower levels of computer knowledge acquisition.

Hypothesis 6c: Beliefs that the computer is a beneficial tool and that the computer is an autonomous entity interact in their effects on computer knowledge acquisition. Autonomous entity beliefs are negatively correlated with knowledge acquisition when beneficial tool beliefs are high, but unrelated when beneficial tool beliefs are low.
Method

Sample

Undergraduate students (n = 255) enrolled in psychology courses at Louisiana State University participated in the study in exchange for course credit. All subjects involved in the study participated voluntarily and were free to withdraw at any time. The sample composition was 58% female, the average age was 20.9 years, and the average years of college education completed was 3.1. Approximately half (51%) of the respondents were employed. When asked what type of computer they most often used, 45% indicated that they used an IBM or compatible machine, 35% reported that they did not use computers, 12% used Apple or Macintosh computers, and 8% used some other form of computer. Respondents had an average of 1.9 years of experience with computers.

Procedure

Table 1 presents an overview of the procedure used in the study. Note that the training groups were crossed in a two by two design (modeling by lecture). Subjects participated in the study in groups ranging in size from eight to eighteen; individuals in each group performed the entire experiment together. Four groups participated in the experiment under the no lecture condition (n = 56).
### Table 1

**Breakdown of the Procedure Used to Assess the Hypotheses in the Present Study**

<table>
<thead>
<tr>
<th>Group</th>
<th>Lecture</th>
<th>Practice</th>
<th>Modeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling Plus Lecture</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Lecture Only</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Modeling Only</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>No Lecture</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

groups each (modeling plus lecture n = 67, modeling only n = 67, and lecture only n = 65). Prior to the training, all participants were given 30 minutes to complete a series of four pre-test measures assessing their: beneficial tool attitudes, autonomous entity attitudes, declarative knowledge, and procedural knowledge. For all subjects, the computer attitude measures were administered first followed by the two measures of computer knowledge. Following the pre-test assessment, I briefed all individuals on the nature and goals of the experiment. Subsequent to the briefing, individuals in the modeling plus lecture and lecture conditions received a forty-five minute key learning points lecture concerning the fundamentals of DOS (Disk Operating System) based computer operation (see Appendix B and the description below). The
key learning points lecture was videotaped and presented uniformly to all subjects. Research by Decker (1983) indicates that videotaped instruction is an effective means of presenting training information. This method can also alleviate any inadvertent biases due to slight differences in the information presented to subjects in different groups.

Upon completion of the key learning points lecture (or following the pre-test measures for subjects in the no lecture conditions), all subjects engaged in a one hour task practice session. The session covered all aspects of the content presented in the key learning points lecture. Subjects in the modeling conditions received behavior modeling training (via videotape) prior to each module of the practice session. Following the task practice, all subjects completed a post-test of the same four measures administered in the pre-test session. Additionally, a computer performance test (see below) was added to assess the trainees' transfer of learning from the training program to the actual execution of tasks.

I conducted all training sessions in the same computer lab in the psychology department at the university. Each subject had access to a computer with a keyboard and a monitor for the duration of the experiment. Administration of the task practice and computer performance test was achieved using a computer program I
wrote specifically for the training program. All subjects, upon completion of the experiment, were debriefed as to the nature and purpose of the experiment.

**Key Learning Points Lecture**

Unlike previous training efforts that have used specific computer software (e.g., Gist et al., 1988; 1989), this training program sought to provide subjects with more generalizable computer knowledge. Admittedly DOS is a software program, however, almost all IBM compatible personal computers in use today require DOS to operate; the same can not be said for other software programs (e.g., word processors, spreadsheets, databases). In order to determine the content of the training, I reviewed several instructional texts on DOS based computer operation. Based on this review and upon my own personal experience, I gleaned the ten most often used DOS commands and grouped them into four categories. The four categories and their corresponding commands are: simple commands (clearing the screen and setting the date and time), viewing files (use and content of the directory command and how to view the contents of a single file), rearranging files (copying, renaming, and removing files), and rearranging directories (changing, creating, and removing directories).

To facilitate the use of the commands discussed above, I gave trainees a brief introductory discussion of
basic computer concepts such as byte storage, hard and floppy disks, directories, files, and programs. Subsequently, trainees were instructed on the meaning of the DOS prompt and the basic entry format used for all DOS commands (i.e., command-parameter(s)-enter).

The remainder of the forty-five minute lecture was dedicated to providing information on the 10 commands, their syntax, and their use and function. The goal of the lecture was to provide trainees with the facts and features of each command (i.e., declarative knowledge) such that a knowledge base would be present at the commencement of task practice. Appendix A outlines the content of the videotaped key learning points lecture.

Task Practice Session

The content of the task practice was identical for all subjects in the study, only the training method differed. Individuals received a series of tasks that directly corresponded to the key learning points lecture previously discussed. For each of the four task groups, subjects were presented with the commands to be practiced and the amount of time they would have to work on the tasks (e.g., "The following commands center around the simple commands used in the DOS environment. They include clearing the screen, finding the date and time on the computer, and changing the date and time on the computer. You have 7 minutes to practice these commands.").
To determine how much time I should allocate to the task practice, I pilot tested trainees using the no lecture training condition which afforded trainees with the least amount of information (n = 17). The four task group times were set at the pilot group's obtained mean plus one standard deviation. Table 2 presents a summary of the number of tasks and the allotted time for each of the four task groups. Additionally, I present means and standard deviations for the number of practice attempts trainees made for each of the task groups and the average time it took trainees to complete each task group.

Individuals were given a series of tasks to practice (e.g., "Change the date to May 3, 1992.") the number of which varied with the complexity of the task group and its respective commands. Following the presentation of each task, the subject was prompted for a response. In the event of an incorrect response, the computer gave subjects feedback in the form of: 1) a suggestion as to what type of error they had made (e.g., no space between command and parameter, command correct but parameters wrong, command typed incorrectly), 2) an indication of where in their response the error had occurred, and 3) the correct response. If the subject pressed enter or did not type the correct command, feedback was limited to suggesting which command was appropriate. Subjects continued on each task until they correctly responded or time expired.
### Table 2

#### Summary of Task Practice Attempts and Times by Task Group

<table>
<thead>
<tr>
<th>Task Group</th>
<th>Pre-set Levels</th>
<th>Obtained by Trainees</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Practice Attempts</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simple Commands</td>
<td>8</td>
<td>12.1 (3.4)</td>
</tr>
<tr>
<td>Viewing Files</td>
<td>15</td>
<td>38.3 (10.6)</td>
</tr>
<tr>
<td>Rearranging Files</td>
<td>11</td>
<td>22.5 (5.6)</td>
</tr>
<tr>
<td>Rearranging Directories</td>
<td>16</td>
<td>15.6 (5.4)</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>50</td>
<td>88.5</td>
</tr>
<tr>
<td><strong>Practice Time</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simple Commands</td>
<td>7</td>
<td>3.7 (1.6)</td>
</tr>
<tr>
<td>Viewing Files</td>
<td>20</td>
<td>15.6 (4.4)</td>
</tr>
<tr>
<td>Rearranging Files</td>
<td>8</td>
<td>5.1 (1.7)</td>
</tr>
<tr>
<td>Rearranging Directories</td>
<td>20</td>
<td>14.4 (4.5)</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>55</td>
<td>38.8</td>
</tr>
</tbody>
</table>

Note: n = 254. Standard deviations are in parentheses. Times are in minutes.

Within each portion of the practice, tasks differed in the specific parameters required and increased in complexity. This last aspect served to both increase learning as well as to encourage trainees to begin to
generalize their rules of how DOS commands work to a variety of instances. A description of the tasks presented to subjects in the practice session appears in Appendix B.

Modeling Intervention

In the modeling conditions, individuals viewed a videotaped presentation of an actor performing each task. Prior to commencement of instruction for each of the four task groups (simple DOS commands, viewing files, rearranging files, and rearranging directories), the computer program halted and told subjects that they would be given a demonstration of how the various commands actually work on the computer. The model described each of the commands within each task group separately. For each of the commands, the model: 1) defined and described the command, 2) told the subjects what he was going to do, 3) discussed why the solution was effective, 4) performed the keystrokes (verbalizing as he went), 5) discussed the output on the monitor, and 6) used the directory command (when appropriate) to show the trainees the outcome of the command. This approach is very much akin to the "show and tell" methodology used by Gist et al. (1988; 1989) in their computer training research. Following the model's presentation of the commands in a task group, subjects proceeded through the task in the same manner and with the same amount of time provided as participants in the other
conditions thereby keeping the amount of practice time constant across groups. A transcript of one of the task group modeling interventions appears in Appendix C.

Computer Knowledge Measures

I developed the computer knowledge measures using a point-to-point approach. That is, the content of the declarative and procedural knowledge measures was a direct outgrowth of the key learning points lecture and the task practice, respectively. Specifically, I used information acquired in the key learning points lecture as a basis for declarative knowledge item generation. Since procedural knowledge is primarily a result of practice, I logically derived the procedural knowledge test from the task practice session.

Declarative knowledge test. Recall that declarative knowledge involves a knowledge of facts or "knowing what" statements. The development of the declarative knowledge test followed along what may be considered traditional lines of test creation. That is, items on the test sought to determine whether trainees recalled the facts they were taught (e.g., "What is the command to remove a file from the computer?"). I scored the declarative knowledge test as the percent correct for the seventeen items that composed the scale (see Appendix D).

Procedural knowledge test. In contrast to declarative knowledge about computers in general and DOS
commands in specific, procedural knowledge addresses whether or not individuals know how to execute DOS commands. Therefore, the primary source for items generated for this test was the task practice session. The key delineation between the procedural knowledge test and the declarative knowledge test can most easily be seen in the types of questions asked. For example, instead of asking subjects, "What is the command to remove a file from the computer?", I would ask, "How does one remove the file MYDAT.TXT from the OLD directory on the computer?" The correct answer for the former is, "DEL", while the correct answer for the latter is, "Type DEL MYDAT.TXT and press enter." Again, the distinction is between knowing what the command is (declarative knowledge) and knowing how to execute the command (procedural knowledge).

Following Anderson's (1982) discussion of how individuals demonstrate their procedural knowledge, I also generated questions which sought to determine if subjects could generalize a principle from one DOS command to another. For example, in the lecture and task practice, subjects were presented with the application of the global wildcard characters (?) and *) in the directory, copy, and delete commands. One test of the generalizability of trainee knowledge was an item which addressed the use of wildcard characters in the rename command.
Another demonstration of procedural knowledge mentioned by Anderson (1982) concerns how individuals with procedural knowledge arrive at more optimal solutions. Accordingly, I included some items which sought to determine if the trainee could arrive at the best solution to the problem. For example, question number 8 on the procedural knowledge test ("Assume that the directory DATA contains the files: ONE.XY, TRI.XX, FOR.XY, SIX.XX, FIV.ZX, and TOO.XX. What is the fastest way to delete the files that have an XX extension?") could be optimally answered by: "Type DEL \DATA\*.XX and press enter." However, another correct answer would be: "Type DIR \DATA, press enter, look for all the files with a .XX extension, and delete each one using the command: DEL \DATA\ <filename> .XX." Thus, a third means of assessing procedural knowledge was to assess whether trainees knew the best way to execute the command.

Since the procedural knowledge test possessed more information than the declarative knowledge test and involved demonstrating the best solution, I scored the test using a two-point system. A score of two was obtained only for a precise response. Trainees received a score of one if: 1) their responses contained minor syntactical errors, 2) the response was sub-optimal, or 3) one of the major components was missing. Scores for all
thirteen items were summed and transformed to a percent correct format (see Appendix E).

**Pilot Testing.** To determine the psychometric validity of the newly developed knowledge measures, I conducted a pilot study using undergraduates who participated in exchange for extra credit. The pilot study consisted of an administration of the two knowledge measures (n = 135) followed a week later by a lecture on computers (later to become the key learning points lecture) and a re-testing of computer knowledge (n = 53) using the same measures.

Results indicated that the declarative and procedural knowledge measures had strong internal consistency reliability at both pre-test and post-test assessment (pre-test α's = .84 for both knowledge tests; post-test α's = .88 and .90 for declarative and procedural knowledge, respectively). Further, test-retest reliability indicated strong and significant relationships for both declarative knowledge (r = .79, p < .01), and procedural knowledge (r = .74, p < .01). Finally, the measures were found to be highly interrelated at both testing periods (r_{pre-test} = .79, p < .01, r_{post-test} = .91, p < .01).

An additional goal of the pilot study was to identify any test practice effects on computer knowledge. To examine this possibility, I performed a t-test comparing
subjects who received pre-test knowledge assessments with subjects who only received a post-test on post-test declarative and procedural knowledge. Results were non-significant for both measures indicating no practice effects.

Thus, the pilot study: 1) offered evidence for the stability and consistency of my declarative and procedural knowledge measures, and 2) mitigated possible threats to the internal validity of the training program due to testing effects. However, some concern as to the discriminant validity of the two measures is warranted given the strong relationship between the two measures. Specifically, one could argue that, based on the strength of this relationship, the procedural knowledge test is merely a more difficult test of declarative knowledge.

The alternative means of assessing procedural knowledge is to measure trainee performance which, in turn, has its faults. Given that performance is the execution of learned behavior rather than a measure of what is actually learned, it can also be construed as an imperfect measure of procedural knowledge. Therefore, the assessment of procedural knowledge is derived from two imperfect measures of the construct: 1) computer performance (a behavioral test), and 2) procedural knowledge (a written test).
Computer Performance Test

In addition to the assessment of computer knowledge and attitudes at post-testing, trainees were also given a computer performance test. As discussed above, the primary distinction between this test and the procedural knowledge test involves having the trainees actually execute the commands on the computer. In the procedural knowledge test, subjects were asked to describe the commands (or sequence of commands) used. In the task practice, trainees were given a specific task (e.g., "Remove the file STUFF.DAT from the DATA directory.") and feedback on the accuracy of their response. The computer performance test differed from both of these by requiring that trainees actually perform the task on the computer without receiving experimenter provided feedback (through the computer). Trainees were however free to check their work thereby providing themselves with feedback on their performance. For example, once the subject enters the command to remove a file, the directory command could be issued to determine if the file had indeed been deleted. To determine the trainee's performance level, I utilized tasks which required trainees to integrate several of the commands they learned toward the solution of a goal task. For example, the task, "Remove the OLD directory," requires that the trainee: 1) remove all files and sub-directories in the OLD directory, 2) change to a directory
one level above where the OLD directory resides, and 3) remove the directory. Instructions and the list of tasks to be performed were presented to the trainee on a sheet of paper (see appendix F); trainees were given thirty-five (35) minutes to complete the task.

The computer recorded the commands entered and the time spent for each command. Using this information, I obtained three measures of performance. First, accuracy was computed as the total number of points awarded for correct execution. Scores were computed in a two step process. In the first step, the computer automatically scored each question based on both correct execution of the required steps and on the basis of incorrect steps (either doing more than required or doing what is required but to the wrong files or directories). In the second step, I examined the scoring scheme for any omissions in the process and wrote a program to correct those omissions accordingly.

Although accuracy is paramount, a consideration of the speed and efficiency with which trainees perform computer tasks is congruent with characterizations of procedural knowledge acquisition (Anderson, 1982). As such, my second measure of performance was performance speed which was calculated as the total time to complete the test weighted by the number of points scored. Since time alone cannot distinguish between trainees who finish
quickly because they could not complete the test and trainees who are fast because they are knowledgeable, I deemed it necessary to operationalize speed as time per accuracy point scored. Similarly, performance efficiency was computed as the total number of commands entered weighted by the number of points scored in the accuracy measure.

**Computer Attitude Measures**

I derived the computer attitude measures from an item analysis of data collected by Brock (1993). In that study, I measured subjects' responses to 77 items from 5 scales measuring attitudes toward computers. To reduce the number of items to a more manageable size, I examined item-total score correlations using two criteria for item retention: 1) a significant correlation greater than .40 with the attitude dimension that the item was supposed to measure, and 2) a correlation less than .30 with the attitude dimension the item was not supposed to measure. Through this first step, the number of items was reduced from 77 to 37 with 15 items on the beneficial tool scale and 22 items on the autonomous entity scale.

Once this reduction was achieved, I computed internal consistency measures for the two scales. Chronbach's alphas were .84 and .91 for the beneficial tool and autonomous entity scales, respectively. To reduce the number of items on the two scales to a more reasonable
number, I removed all items with item-total score correlations less than .40 for the beneficial tool scale and .50 for the autonomous entity scale. The higher criterion for the autonomous entity scale was necessary due to the greater homogeneity of items on that scale as compared to the beneficial tool scale. This procedure reduced the number of items to 11 and 12 for the beneficial tool and autonomous entity scales, respectively. The revised alphas for the beneficial tool ($\alpha = .82$) and autonomous entity ($\alpha = .90$) scales remained substantially the same. An examination of the attitudes' relations with computer use indicated that the beneficial tool scale correlated .49 with computer use ($p < .01$) while the autonomous entity scale correlated -.23 with computer use ($p < .01$). The correlation between the scales was also significant ($r = -.27$, $p < .01$).

I also conducted a confirmatory factor analysis to determine the fit of the selected items to their respective factors. Using the EQS structural equations modeling program (Bentler, 1989) with a generalized least squares solution, each of the 23 items was hypothesized to load on its respective attitude dimensions. The correlation between the two dimensions was also estimated in the analysis. To demonstrate the fit of the hypothesized model, I used both the $X^2$ goodness of fit statistic and the comparative fit index (Bentler, 1990).
Results of the confirmatory factor analysis yielded a significant $X^2$ goodness-of-fit index ($X^2 = 363.1$, $df = 229$, $p < .01$) suggesting poor model fit. However, due to the sensitivity of the $X^2$ test to sample deviations from linearity and multivariate normality, I used the $X^2/df$ ratio as an alternative index. The $X^2/df$ ratio is 1.6 which is less than the 2.0 ratio needed to indicate acceptable fit (Byrne, 1989). Additionally, the comparative fit index was .99 indicating excellent fit between model and data. Finally, the obtained correlation between the dimensions corresponded closely to the value obtained by simply summing the scales ($r = -.26$, $p < .01$).

Visual inspection of the remaining items revealed two items on the autonomous entity scale with identical content and similar wording. One of these items was removed leaving a total of 22 attitude items (11 per scale). These items were presented to subjects in random order. The items used to assess attitudes toward computers in the present study are presented in Appendices G and H for the beneficial tool and autonomous entity scales, respectively.
Results

The primary independent variables in this study were: lecture condition (lecture versus no lecture), modeling condition (modeling versus no modeling), and time (pre-test and post-test). Declarative knowledge, procedural knowledge, performance (accuracy, speed, and efficiency), beneficial tool computer attitudes, and autonomous entity computer attitudes served as dependent variables. Hypothesis 1 treated the computer knowledge variables as independent variables and Hypothesis 6 treated the computer attitude measures as independent variables.

Measurement Adequacy and Descriptive Statistics

Table 3 presents the descriptive statistics, correlations, and reliabilities for all the variables in the study. All variables possessed acceptable internal consistency by established standards (i.e., $a$'s > .70). Notably, the $a$'s for the computer attitude measures compare favorably with the values obtained in the item analysis reported above (see pp. 75-76). The high alpha coefficient obtained for performance accuracy ($a = .90$) is encouraging as it supports the notion that the tasks within the test were homogenous in content, and also grants some validity to the automated scoring procedure used to assess trainee performance.

Since my measures of declarative and procedural knowledge were created for this study, my analysis of
Table 3  

**Sample Sizes, Means, Standard Deviations, Correlations, and Internal Consistency Measures for All Study Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>n</th>
<th>M</th>
<th>SD</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 PREBT</td>
<td>255</td>
<td>76.8</td>
<td>11.4</td>
<td></td>
<td></td>
<td>.81</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 PREAE</td>
<td>255</td>
<td>49.1</td>
<td>14.2</td>
<td>-.31</td>
<td>(.83)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 PREDK</td>
<td>255</td>
<td>11.5</td>
<td>18.1</td>
<td>.23</td>
<td>-.33</td>
<td>(.89)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 PREPK</td>
<td>255</td>
<td>3.8</td>
<td>11.4</td>
<td>.20</td>
<td>-.24</td>
<td>.78 (.91)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 POSTBT</td>
<td>253</td>
<td>77.8</td>
<td>12.9</td>
<td>.71</td>
<td>-.39</td>
<td>.27</td>
<td>.20</td>
<td>(.86)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 POSTAE</td>
<td>253</td>
<td>47.4</td>
<td>14.4</td>
<td>-.36</td>
<td>.82</td>
<td>.37</td>
<td>-.26</td>
<td>-.46</td>
<td>(.84)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 POSTDK</td>
<td>254</td>
<td>52.5</td>
<td>21.8</td>
<td>.12</td>
<td>-.36</td>
<td>.60</td>
<td>.42</td>
<td>.27</td>
<td>-.40</td>
<td>(.84)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 POSTPK</td>
<td>254</td>
<td>36.2</td>
<td>21.7</td>
<td>.17</td>
<td>-.35</td>
<td>.66</td>
<td>.48</td>
<td>.31</td>
<td>-.41</td>
<td>.79 (.87)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9 ACC</td>
<td>233</td>
<td>35.8</td>
<td>22.6</td>
<td>.15</td>
<td>-.31</td>
<td>.55</td>
<td>.43</td>
<td>.28</td>
<td>-.37</td>
<td>.77 (.84)</td>
<td>.84 (.90)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 SPEED</td>
<td>232</td>
<td>1.7</td>
<td>5.0</td>
<td>-.12</td>
<td>-.03</td>
<td>-.08</td>
<td>-.09</td>
<td>-.08</td>
<td>-.33</td>
<td>-.17</td>
<td>-.17</td>
<td>-.24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 EFF</td>
<td>233</td>
<td>2.5</td>
<td>2.9</td>
<td>-.10</td>
<td>-.19</td>
<td>-.13</td>
<td>-.08</td>
<td>.13</td>
<td>-.30</td>
<td>-.37</td>
<td>-.42</td>
<td>.30</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Variables beginning with PRE are pre-test measures, variables beginning with POST are post-test measures. BT = Beneficial Tool, AE = Autonomous Entity, DK = Declarative Knowledge, PK = Procedural Knowledge, ACC = Performance Accuracy, SPEED = Performance Speed, EFF = Performance Efficiency. All means and standard deviations are expressed as percentages except for speed (in minutes) and efficiency (in number of commands). Numbers in parentheses are Coefficient α’s. p < .05 for r’s > ± .13. p < .01 for r’s > ± .17.
their internal consistency properties was more rigorous. I submitted both knowledge tests using both pre-test and post-test assessments to principle components analyses with varimax rotations. Therefore, a total of four principle components analyses were conducted. In each instance, I compared a one factor solution to a two factor solution to assess which was more representative. As suggested by the high internal consistency coefficients for the knowledge measures (see Table 3), my analysis supported a one factor solution for both knowledge scales. The factor loadings for the pre- and post-test declarative and procedural knowledge tests appear in Appendix I. Below I describe the factor analytic results.

The one-factor solution for pre-test declarative knowledge (Lambda = 7.0) accounted for 41.3 percent of the total variance. Further, most items loaded above the .30 criterion suggested by Tabachnick and Fidell (1989; see Appendix I). The two factor solution (Lambda = 1.8) explained an additional 10.7% of the variance. Examination of the rotated factor matrix revealed that 7 of the 17 items loaded greater than .30 on both factors making the interpretation of a two-factor solution difficult. Further, one item (item 15) had a negative loading on the second factor indicating still greater difficulty in interpreting a two-factor solution.
Post-test declarative knowledge had a one factor solution that accounted for 28.3% of the variance among the items (Lambda = 4.8). The addition of a second factor contributed only 9.6% to the total variance explained and had an eigenvalue slightly greater than one (Lambda = 1.6). Examination of the factor loadings in the one-factor solution revealed a pattern similar to that observed at pre-test. All items loaded above .30 with the exception of item 16. Further, there is a demonstrated correspondence in the magnitude of the factor loadings between pre-test and post-test (see Appendix I) that is not found in the two-factor solution. That is, the relative ranking of the size of the factor loadings was stable from pre-test to post-test in the one-factor solution but not in the two-factor solution. I determined that a one-factor solution was most appropriate for the declarative knowledge test because: 1) the first factor eigenvalues were much larger than the second factor eigenvalues (at both pre-test and post-test), and 2) the pattern of factor loadings was confused in the two-factor solution and quite clear for the one-factor solution.

Results from the analysis of procedural knowledge were clearer than those for declarative knowledge. The one factor analysis for pre-test procedural knowledge yielded a large eigenvalue (Lambda = 7.4) which accounted for 56.6% of the variance among the items. The second
factor (Lambda = 1.5) added 11.2% to the total variance explained. All factor loadings were above .30 in the one-factor solution. The two-factor solution yielded 7 of 13 items which had high loadings on both factors. Thus, the interpretation of two distinct factors would be difficult. Further, two different items on each of the two factors loaded negatively. Finally, the second factor had 4 items which did not load above .30.

The post-test factor analysis revealed a one-factor solution that accounted for 41.9% of the total variance (Lambda = 5.5). The two-factor solution added 9.1% of the variance and, like the pre-test procedural knowledge measure, had an eigenvalue near one (Lambda = 1.2). The factor loadings for the one-factor solution were, with the exception of item 1, all greater than .30. The fact that the first item was simplest (i.e., nearly all trainees answered it correctly at post-test) probably contributed to the low variance for the item and hence a low factor loading. Examination of the rotated factor matrix for the two-factor post-test procedural knowledge test revealed that only the last two items loaded greater than .30 on the second factor. Items 3 and 7 had factor loadings greater than .30 on both factors and item 1 did not load on either factor.

Since specific aspects of procedural knowledge were addressed by a sub-set of the items, I also examined the
two-factor solution in terms of the interpretability of these items. Specifically, items 7, 8, and 12 asked subjects to describe the fastest or most efficient means of executing commands, and items 5 and 9 required subjects to generalize their knowledge to a new domain (see Appendix E). Items 7 and 12 did load greater than .30 on the second factor, but the remaining items mentioned above did not. Further, item 7 also loaded highly on the first factor. Hence, the items on the procedural knowledge test appear to fit best in a one-factor solution. At both pre-test and post-test, one large factor emerged with unambiguous factor loadings. The addition of a second factor neither contributed substantially to the variance explained, nor added clarity to interpretation of the content of the scale.

In addition to the similar factor loadings in pre- and post-test knowledge measures, further evidence for the stability of the knowledge and attitude measures was obtained from test-retest correlations. All were significant and positive (declarative knowledge $r = .60, p < .01$; procedural knowledge $r = .48, p < .01$; beneficial tool beliefs $r = .71, p < .01$; autonomous entity $r = .82, p < .01$). Correlations between the two types of knowledge remained stable and significant across time (pre-test $r = .78, p < .01$; post-test $r = .79, p < .01$). The significant negative relationship between the two types of
attitudes at pre-test ($r = -.31, p < .01$) became stronger at post-test ($r = -.46, p < .01$) although not significantly more so.

Correlations between the different measures of performance, although significant, were relatively low in effect size ($r^2$ ranged from .06 to .18). The low effect sizes were not surprising given that the measures of speed and efficiency had performance accuracy divided out in their calculations (a step analogous to partialling out accuracy variance). Consistent with the notion that performance is a measure of procedural knowledge, the relationship between post-test procedural knowledge and the primary measure of performance (accuracy) was very strong ($r = .84, p < .01$).

**Tests of Hypotheses**

**Knowledge-Performance Relations.** Hypothesis 1 predicted significant relations between declarative knowledge and performance (1a) and between procedural knowledge and performance (1b). Additionally, I expected procedural knowledge to be significantly more predictive of performance than declarative knowledge (1c). The correlations presented in Table 3 provide support for Hypotheses 3a and 3b. The post-test correlations between declarative knowledge and accuracy ($r = .77, p < .01$) and between procedural knowledge and accuracy ($r = .84, p < .01$) were both significant and positive. The other
measures of performance (speed and efficiency) were significantly related to the post-test knowledge measures although their magnitudes were substantially smaller ($r_{DK\cdotSPEED} = -.17, p < .05; r_{DK\cdotEFF} = -.30, p < .01; r_{PK\cdotSPEED} = -.17, p < .05; r_{PK\cdotEFF} = -.37, p < .01$).

In order to explore all the components of Hypothesis 1 in a single design, I conducted a series of regression analyses (see Table 4). Through these analyses I was able to: assess direct relations between declarative and procedural knowledge, determine effect sizes for the relationships between the two knowledge measures and performance, and measure the incremental predictiveness of procedural knowledge over declarative knowledge. The three performance measures (accuracy, speed, and efficiency) were used as dependent variables. For each measure, I analyzed two regression equations using declarative and procedural knowledge as independent variables. In the first equation, I entered declarative knowledge first followed by procedural knowledge. The second equation reversed the order of entry. Therefore, a total of 6 regression equations were analyzed.

The first analysis in Table 4 regressed performance accuracy on to declarative knowledge and then procedural knowledge. Entry of declarative knowledge on accuracy was significant ($\beta = .77$, change in $R^2 = .59, p < .01$). Entry of procedural knowledge in the second step significantly
Table 4

**Blocked Regressions for Declarative and Procedural Knowledge on Computer Performance Measures**

<table>
<thead>
<tr>
<th>Performance Measure</th>
<th>Beta</th>
<th>$R^2$</th>
<th>Change in $R^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accuracy</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DK First Step 1 - DK</td>
<td>.77**</td>
<td>.59</td>
<td>.59**</td>
</tr>
<tr>
<td>Step 2 - PK</td>
<td>.61**</td>
<td>.73</td>
<td>.14**</td>
</tr>
<tr>
<td>PK First Step 1 - PK</td>
<td>.84**</td>
<td>.70</td>
<td>.70**</td>
</tr>
<tr>
<td>Step 2 - DK</td>
<td>.29**</td>
<td>.73</td>
<td>.03**</td>
</tr>
<tr>
<td><strong>Speed</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DK First Step 1 - DK</td>
<td>-.17*</td>
<td>.03</td>
<td>.03*</td>
</tr>
<tr>
<td>Step 2 - PK</td>
<td>-.09</td>
<td>.03</td>
<td>.00</td>
</tr>
<tr>
<td>PK First Step 1 - PK</td>
<td>-.17*</td>
<td>.03</td>
<td>.03*</td>
</tr>
<tr>
<td>Step 2 - DK</td>
<td>-.10</td>
<td>.03</td>
<td>.00</td>
</tr>
<tr>
<td><strong>Efficiency</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DK First Step 1 - DK</td>
<td>-.32**</td>
<td>.10</td>
<td>.10**</td>
</tr>
<tr>
<td>Step 2 - PK</td>
<td>-.33**</td>
<td>.14</td>
<td>.04**</td>
</tr>
<tr>
<td>PK First Step 1 - PK</td>
<td>-.37**</td>
<td>.14</td>
<td>.14**</td>
</tr>
<tr>
<td>Step 2 - DK</td>
<td>-.06</td>
<td>.14</td>
<td>.00</td>
</tr>
</tbody>
</table>

Note: Beta = Standardized beta weight.

* = $p < .05$. ** = $p < .01$. 
increased $R^2$ ($\beta = .61$, change in $R^2 = .14$, $p < .01$). The next equation used the same dependent variable (accuracy) but reversed the order of entry of the knowledge measures. In this instance, procedural knowledge entered first and was significant ($\beta = .84$, change in $R^2 = .70$, $p < .01$). Entry of the declarative knowledge measure produced a significant $\beta = .29$ ($p < .01$) but small (3%) increase in $R^2$ ($p < .01$).

In addition to using the primary measure of performance (accuracy), I also conducted regressions using speed and efficiency. The results for performance speed with declarative knowledge entered first were significant $\beta = -.17$ (change in $R^2 = .03$, $p < .05$). The addition of procedural knowledge was not significant ($\beta = .09$, change in $R^2 = .00$, ns). When procedural knowledge entered first, however, the beta and associated $R^2$ were significant ($\beta = -.17$, change in $R^2 = .03$, $p < .05$). The inclusion of declarative knowledge after procedural knowledge did not increase the prediction of performance speed ($\beta = -.10$, change in $R^2 = .00$, ns).

The final performance measure, efficiency, produced somewhat more interesting results. When declarative knowledge was entered first, the results were significant ($\beta = -.32$, change in $R^2 = .10$, $p < .01$). Further, the inclusion of procedural knowledge explained additional variance beyond declarative knowledge ($\beta = -.33$, change in...
\( R^2 = .04, p < .01 \). When the order of entry was reversed, procedural knowledge was a significant predictor of performance efficiency (\( \beta = -.37 \), change in \( R^2 = .14, p < .01 \)), but the inclusion of declarative knowledge in the second step was non-significant (\( \beta = -.06 \), change in \( R^2 = .00, ns \)).

Results for Hypothesis 1 depended upon the type of performance measure used. Both knowledge variables, when entered first in the equation, were significant for all three performance measures which restates the correlational findings. Further, the effect sizes for performance accuracy were large for both declarative knowledge \( (R^2 = .59) \) and procedural knowledge \( (R^2 = .70) \). These two observations provide full support for Hypotheses 1b. Although the declarative knowledge effect size was smaller than the procedural knowledge effect size, the effect was much larger than expected thereby indicating partial support for Hypothesis 1a.

Procedural knowledge contributed significant incremental variance each time it entered an equation in support of Hypothesis 1c. However, declarative knowledge also contributed incremental variance to the prediction of performance accuracy indicating somewhat more mitigated support for this hypothesis. Using efficiency, procedural knowledge added significant variance when it was entered second whereas declarative knowledge did not thereby fully
supporting Hypothesis 1c. Thus, Hypothesis 1b found full support whereas Hypotheses 1a and 1c found strong but partial support.

Training Condition Hypotheses. Tables 5, 6, and 7 present the results from the ANOVAs conducted to examine the second and third hypotheses which predicted gains in knowledge as a result of training methodology. The analyses were 2 (modeling versus no modeling) X 2 (lecture versus no lecture) X 2 (pre-test and post-test) mixed model ANOVAs. The first two factors were between subjects, the other, within subjects. When performance was used to assess procedural knowledge, the within subjects factor was removed from the analysis since there was no pre-test.

Hypothesis 2 stated that declarative knowledge would increase as a result of modeling training (2a) and lecture training (2b). Hypothesis 2a was supported by a significant time by modeling interaction (see Table 5). The means and standard deviations for pre-test and post-test declarative knowledge sub-grouped by training condition appear in Table 8. At post-test, individuals who received modeling scored significantly greater on declarative knowledge ($\bar{M} = 58.3$) than those who did not receive modeling ($\bar{M} = 46.1; t_{252} = 4.6, p < .01$). Simple effects tests on the means in Table 8 revealed that no differences between modeling groups existed at pre-test
Table 5

Modeling by Lecture by Time ANOVA for Declarative Knowledge.

<table>
<thead>
<tr>
<th>Effect</th>
<th>df</th>
<th>MS</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling (M)</td>
<td>1</td>
<td>5707.7</td>
<td>9.1*</td>
</tr>
<tr>
<td>Lecture (L)</td>
<td>1</td>
<td>141.9</td>
<td>0.2</td>
</tr>
<tr>
<td>M X L</td>
<td>1</td>
<td>109.0</td>
<td>0.2</td>
</tr>
<tr>
<td>Between Error</td>
<td>250</td>
<td>156129.5</td>
<td></td>
</tr>
<tr>
<td>Time (T)</td>
<td>1</td>
<td>208871.0</td>
<td>1394.1**</td>
</tr>
<tr>
<td>M X T</td>
<td>1</td>
<td>4040.8</td>
<td>27.0**</td>
</tr>
<tr>
<td>L X T</td>
<td>1</td>
<td>166.7</td>
<td>1.1</td>
</tr>
<tr>
<td>M X L X T</td>
<td>1</td>
<td>2.4</td>
<td>0.0</td>
</tr>
<tr>
<td>Within Error</td>
<td>250</td>
<td>149.8</td>
<td></td>
</tr>
</tbody>
</table>

Note: ** = p < .01.

(\(t_{253} = .4, \text{ns}\)). Individuals in both groups showed large increases in declarative knowledge from pre-test to post-test (modeling: \(t_{132} = 28.8, p < .01\); no modeling: \(t_{120} = 24.5, p < .01\)). No lecture by time effect was found for declarative knowledge; Hypothesis 2b was not supported.

Hypothesis 3 stated that both modeling (3a) and lecture (3b) would produce significant gains in procedural knowledge. The ANOVA in Table 6 revealed a pattern of
Table 6

Modeling by Lecture by Time ANOVA for Procedural Knowledge Using the Procedural Knowledge Test.

<table>
<thead>
<tr>
<th>Effect</th>
<th>df</th>
<th>MS</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling (M)</td>
<td>1</td>
<td>3513.4</td>
<td>8.6**</td>
</tr>
<tr>
<td>Lecture (L)</td>
<td>1</td>
<td>174.6</td>
<td>0.4</td>
</tr>
<tr>
<td>M X L</td>
<td>1</td>
<td>253.3</td>
<td>0.6</td>
</tr>
<tr>
<td>Time (T)</td>
<td>1</td>
<td>129116.8</td>
<td>780.5**</td>
</tr>
<tr>
<td>M X T</td>
<td>1</td>
<td>4606.9</td>
<td>27.9**</td>
</tr>
<tr>
<td>L X T</td>
<td>1</td>
<td>248.0</td>
<td>1.5</td>
</tr>
<tr>
<td>M X L X T</td>
<td>1</td>
<td>22.4</td>
<td>0.1</td>
</tr>
<tr>
<td>Within Error</td>
<td>250</td>
<td>165.4</td>
<td></td>
</tr>
</tbody>
</table>

Note: * = p < .05. ** = p < .01.

results similar to those found for the second hypothesis. The means and standard deviations for pre-test and post-test procedural knowledge sub-grouped by training condition appear in Table 9. In support of Hypothesis 3a, modeling was found to interact with time to produce significantly greater procedural knowledge at post-test (modeling M = 41.5, no modeling M = 30.3; t_{252} = 4.2, p < .01). Pre-test procedural knowledge means did not differ
Table 7

Modeling by Lecture ANOVA for Procedural Knowledge Using Accuracy, Speed, and Efficiency Performance Measures.

<table>
<thead>
<tr>
<th>Effect</th>
<th>df</th>
<th>MS</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependent Variable = Accuracy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modeling (M)</td>
<td>1</td>
<td>13801.0</td>
<td>31.1**</td>
</tr>
<tr>
<td>Lecture (L)</td>
<td>1</td>
<td>2031.0</td>
<td>4.6*</td>
</tr>
<tr>
<td>M X L</td>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Error</td>
<td>229</td>
<td>443.8</td>
<td></td>
</tr>
<tr>
<td>Dependent Variable = Speed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modeling (M)</td>
<td>1</td>
<td>189.9</td>
<td>7.7**</td>
</tr>
<tr>
<td>Lecture (L)</td>
<td>1</td>
<td>6.4</td>
<td>0.3</td>
</tr>
<tr>
<td>M X L</td>
<td>1</td>
<td>4.7</td>
<td>0.2</td>
</tr>
<tr>
<td>Error</td>
<td>228</td>
<td>24.7</td>
<td></td>
</tr>
<tr>
<td>Dependent Variable = Efficiency</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modeling (M)</td>
<td>1</td>
<td>12.9</td>
<td>1.6</td>
</tr>
<tr>
<td>Lecture (L)</td>
<td>1</td>
<td>27.3</td>
<td>3.3</td>
</tr>
<tr>
<td>M X L</td>
<td>1</td>
<td>15.5</td>
<td>1.9</td>
</tr>
<tr>
<td>Error</td>
<td>229</td>
<td>8.2</td>
<td></td>
</tr>
</tbody>
</table>

Note: * = p < .05. ** = p < .01.
Table 8

**Pre-test and Post-test Declarative Knowledge Means and Standard Deviations by Training Condition**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Lecture</th>
<th>No Lecture</th>
<th>Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Modeling</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Pre-test</td>
<td>11.4 (17.1)</td>
<td>12.5 (18.3)</td>
<td>11.9 (17.7)</td>
</tr>
<tr>
<td>- Post-test</td>
<td>58.9 (21.3)</td>
<td>57.8 (20.3)</td>
<td>58.3 (20.7)</td>
</tr>
<tr>
<td><strong>No Modeling</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Pre-test</td>
<td>11.3 (18.5)</td>
<td>10.6 (18.9)</td>
<td>11.0 (18.6)</td>
</tr>
<tr>
<td>- Post-test</td>
<td>47.6 (21.9)</td>
<td>44.3 (20.6)</td>
<td>46.1 (21.3)</td>
</tr>
<tr>
<td><strong>Combined</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Pre-test</td>
<td>11.4 (17.8)</td>
<td>11.6 (18.5)</td>
<td></td>
</tr>
<tr>
<td>- Post-test</td>
<td>53.3 (22.2)</td>
<td>51.7 (21.4)</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** Abbreviations are the same as in Table 3. Standard deviations are in parentheses.

Tests of differences between the means in Table 9 indicated significant gains in procedural knowledge over time for both modeling ($t_{132} = 22.1, p < .01$) and no modeling ($t_{120} = 17.8, p < .01$) groups.

The results for the lecture mirrored those in the analysis of declarative knowledge; the key learning points
### Table 9

**Pre-test and Post-test Procedural Knowledge Means and Standard Deviations by Training Condition**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Lecture</th>
<th>No Lecture</th>
<th>Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Modeling</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pre-test</td>
<td>2.8 (9.1)</td>
<td>4.1 (10.1)</td>
<td>3.4 (9.6)</td>
</tr>
<tr>
<td>Post-test</td>
<td>41.8 (22.5)</td>
<td>41.1 (20.8)</td>
<td>41.5 (21.6)</td>
</tr>
<tr>
<td><strong>No Modeling</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pre-test</td>
<td>4.6 (14.2)</td>
<td>3.9 (12.0)</td>
<td>4.3 (13.2)</td>
</tr>
<tr>
<td>Post-test</td>
<td>32.4 (21.8)</td>
<td>28.0 (18.6)</td>
<td>30.3 (20.4)</td>
</tr>
<tr>
<td><strong>Combined</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pre-test</td>
<td>3.7 (11.9)</td>
<td>4.0 (10.9)</td>
<td></td>
</tr>
<tr>
<td>Post-test</td>
<td>37.1 (22.6)</td>
<td>35.1 (20.8)</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** Abbreviations are the same as in Table 3. Standard deviations are in parentheses.

A lecture did not significantly increase trainee's procedural knowledge thereby failing to find support for Hypothesis 3b.

Table 7 presents the results from the analysis of procedural knowledge which used performance as the dependent measure. The means for the three performance measures (accuracy, speed, and efficiency) sub-grouped by
training condition appear in Table 10. I conducted a 2 (modeling versus no modeling) x 2 (lecture versus no lecture) ANOVA for each of the three performance measures (accuracy, speed, and efficiency).

Results from the analysis of accuracy (the primary measure of performance) indicated that both main effects were significant and the means were in the expected direction (modeling M = 42.7, no modeling M = 26.8; lecture M = 39.4, no lecture M = 32.7). Results from the analysis using speed as the dependent variable were significant for modeling (modeling M = 0.9, no modeling M = 2.7) but not for lecture. The efficiency ANOVA produced no significant effects. Therefore, using performance accuracy, Hypotheses 3a and 3b were supported; modeling and lecture both improved performance over no modeling and no lecture, respectively. The performance speed analysis supported Hypothesis 3a but not Hypothesis 3b whereas the efficiency analysis failed to support both hypotheses.

Hypothesis 3c stated that individuals with modeling (modeling plus lecture and modeling only) would have greater gains in procedural knowledge than individuals who only received a lecture. To examine this hypothesis, I performed a planned comparison of the two modeling groups against the lecture only group on the change in procedural knowledge from pre-test to post-test. Results from the analysis indicated that modeling was superior to lecture
Table 10

Accuracy, Speed, and Efficiency Means and Standard Deviations by Training Condition

<table>
<thead>
<tr>
<th>Condition</th>
<th>Lecture</th>
<th>No Lecture</th>
<th>Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Modeling</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Accuracy</td>
<td>45.7 (21.9)</td>
<td>39.7 (22.1)</td>
<td>2.7 (22.1)</td>
</tr>
<tr>
<td>- Speed</td>
<td>1.0 (2.3)</td>
<td>0.9 (1.1)</td>
<td>0.9 (1.8)</td>
</tr>
<tr>
<td>- Efficiency</td>
<td>2.1 (4.4)</td>
<td>2.2 (2.2)</td>
<td>2.2 (3.4)</td>
</tr>
<tr>
<td><strong>No Modeling</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Accuracy</td>
<td>30.2 (22.1)</td>
<td>24.2 (17.8)</td>
<td>26.8 (19.9)</td>
</tr>
<tr>
<td>- Speed</td>
<td>3.1 (9.5)</td>
<td>2.4 (4.8)</td>
<td>2.7 (7.2)</td>
</tr>
<tr>
<td>- Efficiency</td>
<td>2.1 (1.1)</td>
<td>3.3 (2.3)</td>
<td>2.8 (2.0)</td>
</tr>
<tr>
<td><strong>Combined</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Accuracy</td>
<td>39.3 (28.2)</td>
<td>32.7 (21.6)</td>
<td></td>
</tr>
<tr>
<td>- Speed</td>
<td>1.8 (6.4)</td>
<td>1.6 (3.4)</td>
<td></td>
</tr>
<tr>
<td>- Efficiency</td>
<td>2.1 (3.4)</td>
<td>2.8 (2.3)</td>
<td></td>
</tr>
</tbody>
</table>

NOTE: Standard deviations are in parentheses. Speed = time in minutes to complete test divided by accuracy. Efficiency = commands entered for entire test divided by accuracy.
in providing trainees with procedural knowledge ($F_{1, 250} = 13.9, p < .01$). I also conducted the same test using performance as the dependent measure of procedural knowledge. For the performance accuracy measure, results from the planned comparison were congruent with those found in the procedural knowledge test analysis. Modeling trainees ($M = 42.7$) had significantly greater performance than lecture only trainees ($M = 30.2; F_{1, 229} = 15.5, p < .01$). The planned comparison which used speed as the dependent measure indicated that modeling trainees ($M = .9$) performed faster than trainees who received only a lecture ($M = 3.1; F_{1, 228} = 6.2, p < .05$). When efficiency was used, the results were not significant (modeling: $M = 2.2$; lecture: $M = 2.1; F_{1, 229} = .1, ns$). Therefore, Hypothesis 3c was supported using the procedural knowledge test, performance accuracy, and performance speed, but was not supported using performance efficiency.

**Attitude Change Hypotheses.** Hypotheses 4 and 5 addressed the effects of the training methods on trainee changes in attitudes toward computers. Specifically, Hypothesis 4 predicted that beneficial tool beliefs would increase from pre-test to post-test (4a) and autonomous entity beliefs would decrease from pre-test to post-test (4b). Hypothesis 5 predicted that modeling trainees would have greater gains in beneficial tool attitudes (5a) and greater losses in autonomous entity attitudes (5b) over no
modeling trainees. To examine these hypotheses, I conducted two ANOVAs each using a 2 (modeling versus no modeling) by 2 (lecture versus no lecture) by 2 (pre-test and post-test) mixed model design. The ANOVA in Table 11 used beneficial tool attitudes as the dependent measure. Table 12 presents the results from the ANOVA using autonomous entity attitudes as the dependent measure.

Hypothesis 4a, predicting an increase in beneficial tool attitudes over time, was not supported (see Table 11). In Table 12, the significant main effect for time indicates support for Hypothesis 4b. Trainee autonomous entity beliefs were significantly reduced as a result of the practice and training programs (pre-test autonomous entity M = 49.1, post-test autonomous entity M = 47.4).

Results from the analysis of Hypothesis 5 (predicting that trainees who received a modeling component would show more change in attitudes than trainees who did not receive modeling) were surprising. Tables 11 and 12 indicate that neither Hypothesis 5a nor Hypothesis 5b was supported. In both instances, the modeling by time interaction was non-significant. However, for both beneficial tool and autonomous entity beliefs, the lecture by time interaction was significant. The means and standard deviations for pre-test and post-test attitudes broken down by training condition appear in Table 13 (beneficial tool) and Table 14 (autonomous entity). As would be expected, beneficial
Table 11

**Modeling by Lecture by Time ANOVA for Beneficial Tool Attitudes.**

<table>
<thead>
<tr>
<th>Effect</th>
<th>df</th>
<th>MS</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling (M)</td>
<td>1</td>
<td>3.1</td>
<td>0.0</td>
</tr>
<tr>
<td>Lecture (L)</td>
<td>1</td>
<td>3.7</td>
<td>0.1</td>
</tr>
<tr>
<td>M X L</td>
<td>1</td>
<td>149.3</td>
<td>2.0</td>
</tr>
<tr>
<td>Between Error</td>
<td>250</td>
<td>76.7</td>
<td></td>
</tr>
<tr>
<td>Time (T)</td>
<td>1</td>
<td>26.3</td>
<td>2.0</td>
</tr>
<tr>
<td>M X T</td>
<td>1</td>
<td>0.5</td>
<td>0.0</td>
</tr>
<tr>
<td>L X T</td>
<td>1</td>
<td>109.8</td>
<td>8.4**</td>
</tr>
<tr>
<td>M X L X T</td>
<td>1</td>
<td>2.6</td>
<td>0.2</td>
</tr>
<tr>
<td>Within Error</td>
<td>250</td>
<td>13.0</td>
<td></td>
</tr>
</tbody>
</table>

Note: ** = p < .01.

Tool beliefs increased from pre-test (M = 75.8) to post-test (M = 78.5) for individuals who received lecture training (t_{129} = 3.2, p < .01) and did not change for trainees who did not receive a lecture (t_{122} = 1.1, ns).

The means for the autonomous entity attitudes revealed an unexpected finding. Individuals who did not receive a lecture showed a **decrease** in autonomous entity beliefs from pre-test (M = 49.5) to post-test (M = 46.5,
Table 12

Modeling by Lecture by Time ANOVA for Autonomous Entity Attitudes.

<table>
<thead>
<tr>
<th>Effect</th>
<th>df</th>
<th>MS</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling (M)</td>
<td>1</td>
<td>47.2</td>
<td>0.4</td>
</tr>
<tr>
<td>Lecture (L)</td>
<td>1</td>
<td>15.1</td>
<td>0.1</td>
</tr>
<tr>
<td>M X L</td>
<td>1</td>
<td>26.1</td>
<td>0.2</td>
</tr>
<tr>
<td>Between Error</td>
<td>250</td>
<td>114.2</td>
<td></td>
</tr>
<tr>
<td>Time (T)</td>
<td>1</td>
<td>101.0</td>
<td>9.3**</td>
</tr>
<tr>
<td>M X T</td>
<td>1</td>
<td>11.7</td>
<td>1.1</td>
</tr>
<tr>
<td>L X T</td>
<td>1</td>
<td>57.6</td>
<td>5.3*</td>
</tr>
<tr>
<td>M X L X T</td>
<td>1</td>
<td>40.9</td>
<td>3.8</td>
</tr>
<tr>
<td>Within Error</td>
<td>250</td>
<td>10.8</td>
<td></td>
</tr>
</tbody>
</table>

Note: * = p < .05. ** = p < .01.

$t_{122} = 3.9$, $p < .01$). Lecture trainees did not change their autonomous entity attitudes ($t_{129} = .5$, ns). An examination of the cell means in Table 14 reveals that the disparity between lecture and no lecture groups was particularly large in the groups that also received modeling. That is, modeling with no lecture produced the greatest decrease in autonomous entity beliefs. The inclusion of modeling as a possible component driving the
Table 13

Pre-test and Post-test Beneficial Tool Attitudes by Training Condition

<table>
<thead>
<tr>
<th>Condition</th>
<th>Lecture</th>
<th>No Lecture</th>
<th>Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Pre-test</td>
<td>76.5</td>
<td>77.0</td>
<td>76.8</td>
</tr>
<tr>
<td>- Post-test</td>
<td>79.4</td>
<td>75.8</td>
<td>77.6</td>
</tr>
<tr>
<td>No Modeling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Pre-test</td>
<td>75.0</td>
<td>78.9</td>
<td>76.8</td>
</tr>
<tr>
<td>- Post-test</td>
<td>77.5</td>
<td>78.4</td>
<td>78.0</td>
</tr>
<tr>
<td>Combined</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Pre-test</td>
<td>75.8</td>
<td>77.9</td>
<td></td>
</tr>
<tr>
<td>- Post-test</td>
<td>78.5</td>
<td>77.0</td>
<td></td>
</tr>
</tbody>
</table>

NOTE: Abbreviations are the same as in Table 3.

strange results is suggested by the three-way interaction in Table 12 which was narrowly non-significant (exact $p = .053$). Although statistically speaking the interaction should not be interpreted, visual inspection of the means in Table 14 suggests that modeling without a lecture was most helpful in reducing autonomous entity attitudes. Based on the fact that: 1) the three-way interaction was narrowly non-significant, and 2) the differences between
### Table 14

**Pre-test and Post-test Autonomous Entity Attitudes by Training Condition**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Lecture</th>
<th>No Lecture</th>
<th>Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Modeling</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Pre-test</td>
<td>47.5 (13.3)</td>
<td>50.0 (13.0)</td>
<td>48.8 (13.2)</td>
</tr>
<tr>
<td>- Post-test</td>
<td>47.7 (15.1)</td>
<td>45.6 (11.5)</td>
<td>46.6 (13.4)</td>
</tr>
<tr>
<td><strong>No Modeling</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Pre-test</td>
<td>50.0 (14.7)</td>
<td>48.7 (16.1)</td>
<td>49.4 (15.3)</td>
</tr>
<tr>
<td>- Post-test</td>
<td>49.1 (15.3)</td>
<td>47.5 (15.8)</td>
<td>48.3 (15.5)</td>
</tr>
<tr>
<td><strong>Combined</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Pre-test</td>
<td>48.8 (14.0)</td>
<td>49.5 (14.5)</td>
<td></td>
</tr>
<tr>
<td>- Post-test</td>
<td>48.4 (15.1)</td>
<td>46.5 (13.6)</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** Abbreviations are the same as in Table 3.

the modeling only group's difference score, and all other group's change scores was so large, I chose to test this effect with a conservative post-hoc analysis. Using a Scheffe test for complex comparisons, I compared the modeling only group (post-test minus pre-test $M = -4.4$) to all other groups (post-test minus pre-test $M = -.7$) in terms of a change in autonomous entity attitudes from pre-test to post-test and found that this was in fact the case.
(F_{3, 249} = 9.7, p < .05). Modeling without lecture training reduced trainee autonomous entity beliefs more so than any of the other training groups. Thus, partial support for Hypothesis 5b is warranted with a caveat noted for the use of a post-hoc procedure for a non-significant interaction.

The results suggest that: 1) lecture training increases beneficial tool attitudes, and 2) no lecture training decreases autonomous entity attitudes. The latter was unexpected given that exposure to a computer lecture should decrease these beliefs rather than leaving them unchanged. Modeling did not contribute to changes in trainee beneficial tool attitudes thereby failing to support Hypothesis 5a. Modeling only training did reduce trainee autonomous entity beliefs in support of Hypothesis 5b, but the overall effect for modeling was non-significant.

Knowledge Change as a Result of Attitudes. The final hypotheses in the study used attitudes in a predictive sense. Rather than identifying changes in attitudes as a result of training, I sought to determine if attitudes had any association with knowledge acquisition. The sixth hypothesis predicted that: high beneficial tool attitudes (at pre-test) would be associated with greater knowledge acquisition (6a); high autonomous entity attitudes (at pre-test) would be associated with lower levels of knowledge acquisition (6b); and the two attitudes would
interact in their prediction of computer knowledge acquisition (6c). In Hypothesis 6c, I expected that autonomous entity beliefs would be related to knowledge acquisition only when beneficial tool attitudes were high.

To test Hypothesis 6, I performed a blocked regression analyses on each of five dependent measures: declarative knowledge, procedural knowledge, performance accuracy, performance speed, and performance efficiency. For the analyses incorporating declarative and procedural knowledge, pre-test knowledge was entered first as a covariate. For all five regressions, the main effects of pre-test beneficial tool and autonomous entity attitudes were entered in the next step. The interaction of the two pre-test attitude measures was entered in the final block. Table 15 presents the regression results using the two knowledge tests as dependent measures. Table 16 summarizes the results from the analyses using the three performance measures.

The analysis using declarative knowledge revealed that, after pre-test knowledge was factored out ($\beta = .60$, change in $R^2 = .36$, $p < .01$), the attitude main effects explained incremental variance (change in $R^2 = .04$, $p < .01$). Examination of the beta weights indicated that autonomous entity attitudes were negatively related to declarative knowledge ($\beta = -.23$, $p < .01$) whereas beneficial tool attitudes were unrelated ($\beta = -.07$, ns).
Table 15

Multiple Regressions for Pre-test Beneficial Tool and Autonomous Entity Attitudes on Knowledge Acquisition

<table>
<thead>
<tr>
<th>Knowledge Measure</th>
<th>Beta</th>
<th>R²</th>
<th>Change in R²</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dependent Variable = Declarative Knowledge</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 1 - Covariate</td>
<td>.60**</td>
<td>.36</td>
<td>.36**</td>
</tr>
<tr>
<td>Step 2 - Attitudes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- BT</td>
<td>-.07</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- AE</td>
<td>-.23**</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 3 - BT X AE</td>
<td>.10</td>
<td>.40</td>
<td>.00</td>
</tr>
<tr>
<td><strong>Dependent Variable = Procedural Knowledge</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 1 - Covariate</td>
<td>.46**</td>
<td>.22</td>
<td>.22**</td>
</tr>
<tr>
<td>Step 2 - Attitudes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- BT</td>
<td>-.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- AE</td>
<td>-.28**</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 3 - BT X AE</td>
<td>.13</td>
<td>.29</td>
<td>.00</td>
</tr>
</tbody>
</table>

Note: Abbreviations are the same as in Table 3. Covariate is pre-test knowledge. Beta = Standardized beta weight. * = p < .05. ** = p < .01.

The inclusion of the interaction between beneficial tool and autonomous entity attitudes on declarative knowledge was also not significant.
Table 16

Multiple Regressions for Pre-test Beneficial Tool and Autonomous Entity Attitudes on Performance Measures

<table>
<thead>
<tr>
<th>Knowledge Measure</th>
<th>Beta</th>
<th>$R^2$</th>
<th>Change in $R^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dependent Variable = Accuracy</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 1 - Attitudes</td>
<td>.10</td>
<td>.10**</td>
<td></td>
</tr>
<tr>
<td>- BT</td>
<td>.06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- AE</td>
<td>-.29**</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 2 - BT X AE</td>
<td>-.83*</td>
<td>.12</td>
<td>.02*</td>
</tr>
<tr>
<td><strong>Dependent Variable = Speed</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 1 - Attitudes</td>
<td>.02</td>
<td>.02</td>
<td></td>
</tr>
<tr>
<td>- BT</td>
<td>-.15*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- AE</td>
<td>-.08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 2 - BT X AE</td>
<td>.72*</td>
<td>.04</td>
<td>.02*</td>
</tr>
<tr>
<td><strong>Dependent Variable = Efficiency</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 1 - Attitudes</td>
<td>.02</td>
<td>.02</td>
<td></td>
</tr>
<tr>
<td>- BT</td>
<td>-.08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- AE</td>
<td>.07</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step 2 - BT X AE</td>
<td>.35</td>
<td>.02</td>
<td>.00</td>
</tr>
</tbody>
</table>

Note: Abbreviations are the same as in Table 3. Beta = Standardized beta weight.

* = $p < .05$. ** = $p < .01$. 
The results from the second analysis were similar to the results from the first. The covariate pre-test procedural knowledge was significant ($\beta = .46$, change in $R^2 = .22$, $p < .01$) and the main effect of attitudes entered in the second step was significant (change in $R^2 = .07$, $p < .01$). Analysis of the beta weights revealed that autonomous entity attitudes were related to procedural knowledge ($\beta = -.28$, $p < .01$) whereas beneficial tool attitudes were not ($\beta = -.01$, ns). Finally, the inclusion of the attitude interaction term was not significant.

The final analysis used performance as a measure of procedural knowledge acquisition. Since performance was assessed at post-test only, no covariate was available. The entrance of the attitude main effects on accuracy was significant (change in $R^2 = .10$, $p < .01$). Examination of the beta weights indicated that autonomous entity beliefs were negatively related to accuracy ($\beta = -.29$, $p < .01$) whereas beneficial tool beliefs were unrelated to accuracy ($\beta = .06$, ns). Unlike the previous analyses, the inclusion of the interaction term resulted in a significant increase in explained variance ($\beta = -.83$, change in $R^2 = .02$, $p < .05$).

The other measures of performance produced somewhat more ambiguous results. When performance speed was regressed on to attitudes, the results were non-significant ($R^2 = .02$, ns). Examination of the beta
weights revealed that beneficial tool attitudes bore a significant negative relationship to performance ($\beta = -0.15, p < .05$) indicating that trainees with high pre-test beneficial tool attitudes performed faster. Autonomous entity attitudes were not significantly related to performance speed ($\beta = -0.08$, ns). The inclusion of the attitude interaction term did not contribute incremental variance to the prediction of performance speed ($\beta = 0.72$, change in $R^2 = 0.02$, ns). The results for the final regression analysis using performance efficiency as the dependent measure were non-significant for the entrance of attitudes (beneficial tool: $\beta = -0.08$, ns; autonomous entity: $\beta = 0.07$, ns; $R^2 = 0.02$, ns) and for the inclusion of the interaction term ($\beta = 0.35$, change in $R^2 = 0.00$, ns).

Across both computer knowledge tests and the primary measure of performance (accuracy), support for Hypothesis 6b is evident. Each time autonomous entity attitudes entered the equation they predicted knowledge. Trainees with initially high autonomous entity beliefs had smaller gains in computer knowledge than trainees with low pre-test autonomous entity beliefs. The lack of support for Hypothesis 6a is also evident. Only the speed regression analysis revealed a relationship between pre-test beneficial tool beliefs and computer knowledge.

Hypothesis 6c was only supported in the analysis using performance accuracy as the dependent measures. To
assure that the form of the interaction obtained was congruent with the hypothesized interaction (see Figure 1), I plotted the interaction between the two attitudes on performance accuracy (see Figure 2). To fully test Hypothesis 6c, I conducted simple effects tests using one standard deviation above and below the mean as my conditional values for high and low beneficial tool beliefs, respectively.

The relationship between autonomous entity attitudes and procedural knowledge (as measured by performance accuracy) was significant and negative when beneficial tool beliefs were high ($\bar{f} = -.65, t = -11.0, p < .01$). When beneficial tool beliefs were low, autonomous entity beliefs were unrelated to performance ($\bar{f} = -.26, t = -1.9, \text{ns}$). Given that performance accuracy was the only measure of trainee computer knowledge in which attitudes interacted significantly, the results suggest rather weak support for this final hypothesis.
Figure 2

Computer Attitude Interaction on Performance Accuracy
Discussion

Using computers as its focus, this study examined several factors involved in technology training in an effort to bring psychological research on technology up to pace with the pervasiveness of technology in the workplace. Within the framework of computer training, I incorporated ideas from two popular theories of human learning: cognitive learning theory and social learning theory. From the cognitive learning theory literature, I integrated the concepts of declarative knowledge (knowing what) and procedural knowledge (knowing how). Behavior modeling training, the training pedagogy used in social learning theory, was chosen as the focal training method for my study. In addition to exploring how the two learning theories functioned in the context of computer training, I also examined two attitudes toward computers. Both the impact of these attitudes on computer learning, as well as the effects of training (and behavioral modeling training specifically) on attitudes were explored.

Summary of Hypotheses

The hypotheses explored here can be broken down into four categories: knowledge-performance relationships (Hypothesis 1), training methodology (Hypotheses 2 and 3), attitude change (Hypotheses 4 and 5), and attitude effects on learning (Hypothesis 6). Below I provide a brief
summary of the results from the six hypotheses. A more
detailed discussion of the findings is presented in the
sections that follow.

Relations between declarative and procedural
knowledge and performance were found to be highly
significant and in support of Hypotheses 1a and 1b. Further, procedural knowledge was more predictive of
performance than declarative knowledge in support of
Hypothesis 1c. The first hypotheses also predicted a
moderate effect size for the relationship between
declarative knowledge and performance (1a) and a strong
effect size for the relationship between procedural
knowledge and performance (1b). Contrary to my
hypothesis, the declarative knowledge to performance
relationship was strong in effect size rather than
moderate. On the other hand, the procedural knowledge to
performance relationship did have a large effect size
thereby fully supporting Hypothesis 1b. Hypothesis 1c was
supported by evidence that the relationship between
procedural knowledge and performance was incrementally
greater than the relationship between declarative
knowledge and performance. In a regression framework,
procedural knowledge contributed incremental variance to
performance over declarative knowledge using both
performance accuracy and performance efficiency.
Declarative knowledge was only incrementally predictive
over procedural knowledge when accuracy was the dependent measure. The regression results therefore indicated somewhat weaker support for Hypothesis 1c.

The second and third hypotheses examined the effects of modeling training and lecture training on trainee declarative and procedural knowledge. Hypotheses 2a and 3a predicted that behavior modeling training would yield increases in declarative and procedural knowledge, respectively, and were both supported. Hypotheses 2b predicted that lecture training would produce increases in declarative knowledge. This hypothesis was not supported. In support of Hypothesis 3b, lecture trainees had greater procedural knowledge when measured by performance. However, the gains in procedural knowledge using the procedural knowledge test were not found thereby failing to support Hypothesis 3b with this measure. Hypothesis 3c compared all modeling trainees to trainees who received a lecture on procedural knowledge. Results indicated that modeling improved procedural knowledge over lecture.

The fourth hypothesis in the study looked at whether the training program as a whole changed trainee attitudes toward computers. I expected trainees to have stronger beliefs in the computer as a beneficial tool (4a) and weaker beliefs in the computer as an autonomous entity (4b) at post-test. Results indicated that, across all trainees, beneficial tool beliefs did not increase;
Hypothesis 4a was not supported. However, autonomous entity attitudes decreased over time in support of Hypothesis 4b.

The fifth hypothesis expected changes in the two measured attitudes to be greater for individuals who received behavior modeling training. The analysis indicated that modeling trainees did not have stronger beneficial tool beliefs, hence Hypothesis 5a was not supported. The examination of autonomous entity beliefs was surprising. Modeling did not change autonomous entity beliefs over time any more so than no modeling. Lecture training also showed no change in attitudes over time. However, trainees who did not receive a lecture did show decreases in autonomous entity attitudes. Further, the three-way interaction between modeling, training, and time was narrowly non-significant ($p = .053$). Results from a post-hoc test comparison indicated that individuals in the modeling only training condition had decreases in autonomous entity beliefs that were greater than any of the other training conditions. Therefore, Hypothesis 5b was not supported, but the follow-up analysis suggests that modeling, when presented alone, may reduce trainee autonomous entity beliefs.

The final hypothesis in the study looked at the predictive validity of beneficial tool and autonomous entity beliefs in terms of knowledge acquisition. I
expected that trainees with high beneficial tool beliefs (6a) and low autonomous entity attitudes (6b) would have greater knowledge acquisition. Further, I expected that the two attitudes would interact in their prediction of computer knowledge acquisition (6c). Each of the hypotheses were tested using three dependent measures (declarative knowledge, procedural knowledge, and performance). Across all three measures, pre-test beneficial tool attitudes did not predict knowledge acquisition thereby failing to support Hypothesis 6a. Support for Hypothesis 6b was found across all three measures; pre-test autonomous entity beliefs were negatively related to computer knowledge acquisition. Support for the last hypothesis (6c) was only evident using performance accuracy as the dependent measure. However, the fact that the pattern of the interaction was consistent with what I hypothesized is encouraging. A significant negative relationship between autonomous entity attitudes and performance accuracy was found when beneficial tool beliefs were high whereas the relationship between autonomous entity attitudes and performance was not significant when beneficial tool attitudes were low.

A summary of the hypotheses indicates that, as a whole, the results were favorable. Support for the first hypothesis suggests that the relations between knowledge and performance are consistent with my expectations.
Relations between training methodology and knowledge acquisition discussed in the second and third hypothesis indicated that modeling was a much more effective technique than lecturing. The fourth and fifth hypotheses, which examined the effects of training methodology on computer attitude change, were also a surprise. Although the results were mixed, they present some interesting suggestions concerning the dynamics of trainee interactions with technology. The portions of the final hypothesis that were supported have clear implications for researchers interested in the impact of trainee attitudes on knowledge acquisition.

Each of the four major sections of the results are discussed in detail below. Relations between knowledge and performance are presented first. With this framework in place, I discuss the impact of training methodology on computer knowledge. This section ties the knowledge based research of cognitive learning theory with the methodology based research drawn from social learning theory. In the final two sections of the discussion, I address the role of attitudes toward computers in training both in terms of what can be done to change attitudes as well as how attitudes can affect training outcomes.

Knowledge-Performance Relations

Of great interest to me in formulating the hypotheses which center around the acquisition of computer knowledge
was how different aspects of knowledge were represented. The use of computers as a context for this research brings with it some implications not typically found in the acquisition of other skills. First, computer use requires an expansive database of facts (declarative knowledge) to supplement one's knowledge of how to perform tasks on the computer (procedural knowledge). This differs from other areas where the facts about a task are outnumbered by the procedural knowledge required to do the task (e.g., problem solving). Second, procedures used in the execution of tasks on the computer have a more salient property to them. Whereas procedural knowledge is typically characterized as difficult to describe, computer procedural knowledge appears to be more explicit. Therefore, both declarative and procedural knowledge appear to possess some qualities in the computer context that make them different from applications in other areas.

The fact that procedural knowledge was strongly related to performance ($r = .84$) is not surprising given that the two tests were conceptualized as being measures of the same construct. The strong relationship is encouraging when one considers the differences in measurement methodology between the two. The procedural knowledge test used a free response paper and pencil test that measured trainee's explicit procedural knowledge (i.e., "How do you do this task?"). In contrast, the
performance test entailed tasks presented and scored on an actual computer. Thus the measures are virtual opposite ends on a continuum of explicit to implicit.

A further distinction between the two measures of the procedural knowledge construct concerns the opportunity for feedback. Feedback was non-existent in the procedural knowledge test and immediately accessible in the performance test (i.e., subjects could look at what they had done immediately following the execution of a task or command). The strong correspondence between the two measures suggests that feedback seeking behavior may have been initiated in a pattern congruent with the level of trainee knowledge. That is, individuals with more knowledge were more apt to check their work when given the opportunity.

The strong relationship between the procedural knowledge and performance tests is encouraging to the extent that it suggests that, congruent with my expectations, knowledge of how tasks on a computer are performed is relatively explicit. The explicit procedural knowledge test correlated highly with the implicit performance test. I also expected facts about computer operations (declarative knowledge) to play a much stronger role in this context. The strong relationship between declarative knowledge and performance ($r = .77$) supports this expectation as well.
I was initially surprised that declarative knowledge was so strongly related to performance (I hypothesized the effect size to be moderate). However, the fact that the two measures of knowledge are so process dependent may help to alleviate some of the confusion. Individuals cannot have procedural knowledge without having declarative knowledge first. Thus, some degree of correlation must exist between the two constructs. Although not directly hypothesized, the high relationship between declarative knowledge and procedural knowledge ($r = .79$) supports the convergent validity of procedural knowledge by means of external parallelism. That is, the fact that the declarative knowledge test predicts the procedural knowledge test about as well as it predicts the performance test supports the idea that the two measures of procedural knowledge are parallel (convergent).

The discriminant validity of the tests of declarative and procedural knowledge is a somewhat more confusing issue. Direct comparison of the correlations between the three measures (declarative knowledge, procedural knowledge, and performance) indicates that procedural knowledge has a significantly greater relationship to performance than does declarative knowledge ($t_{250} = 3.1$, $p < .01$). Thus one might expect that the two are different. However, I note that findings using third variables (e.g., training methodology) were uniform across the two measures
of knowledge. Thus, any affects observed on the declarative knowledge test were similarly observed on the procedural knowledge test. Again invoking the concept of external parallelism, this would seem to argue for the convergent rather than discriminant validity of the declarative and procedural knowledge tests.

The analyses of the hypothesis positing that the procedural knowledge test would be more predictive of performance than the declarative knowledge test were somewhat mixed. Using performance speed (operationalized as time per accuracy point scored) as the dependent variable, neither declarative knowledge nor procedural knowledge were incrementally predictive over the other. Both types of knowledge were equally effective in predicting how quickly trainees performed the task. The other measures of performance, accuracy and efficiency (operationalized as number of commands per accuracy point scored), were consistent with my expectations. In both instances, procedural knowledge contributed significant incremental variance to the prediction of performance over declarative knowledge. Further, in the case of efficiency, declarative knowledge added nothing over procedural knowledge in its prediction of performance. Anderson (1982) suggests that the proceduralization of skill intones an increase in efficiency. Therefore, the results are consistent with the cognitive learning
perspective: procedural knowledge related to efficiency above and beyond declarative knowledge. Taken together, the results are mixed concerning the discriminant validity of the declarative and procedural knowledge tests.

One of the more likely explanations for why discriminant validity was not clearly evident may relate to the task used in the study. A hallmark of the declarative-procedural knowledge distinction is the mapping of the two constructs onto an explicit (declarative) to implicit (procedural) continuum. Typically, the nature of the tasks used to demonstrate the distinction between the two types of knowledge (e.g., bicycle riding, geometry proofs) makes clear the differences between the two in terms of how explicit (or implicit) they are (e.g., Anderson, 1982; Gray & Orasunu, 1987; Best, 1989).

As mentioned at the onset of this section, computer use is markedly different from these other types of tasks. The two main distinctions I put forth that make computer use different from other tasks are that: 1) operating computers requires much more declarative knowledge, and 2) procedural knowledge of computers is much more explicit. These two differences may have acted to blur the lines marking declarative and procedural knowledge to the extent that a clear picture of how they differ may be difficult. If successful operation of computers (procedural
knowledge) requires large amounts of declarative knowledge, then the two will necessarily be highly related and any distinctions between the two will therefore be difficult to detect. Furthermore, the idea that procedural knowledge of computers is more explicit than is typical of other tasks (e.g., playing golf) makes the distinction between it and the already explicit declarative knowledge more difficult. The topic of what types of knowledge are being addressed in different instances will be taken up again in the sections that follow.

Training Methodology

Throughout the present effort, I have spoken of an integration of two major learning theories: cognitive learning theory and social learning theory. My analysis of knowledge changes as a function of training methodology addresses this goal most directly. In short, the thesis of this integration is that the processes by which individuals acquire procedural knowledge are similar to the processes detailed in the methods of behavior modeling training. Conversely, the types of knowledge acquired from behavior modeling training have been shown to be similar in description to procedural knowledge. Thus, my goal was to provide some evidence for this association between the two theories.
One finding that is clear from the results is that modeling training works. Regardless of which measure was used (declarative knowledge, procedural knowledge, or performance), trainees who received modeling had higher levels of knowledge and performance than trainees who did not receive modeling. This finding alone is made more interesting by the way in which the modeling training was implemented in this study. Past research has tended toward defining modeling training as a total package which includes lecture and practice (e.g., Gist et al., 1988; 1989). In such research, modeling trainees are compared to trainees receiving everything but modeling. The primary implication discussed in these past studies has been that modeling’s effectiveness over lecture methods is evident. The oversight in past research lies in the inclusion of lecture training as a seemingly necessary component to behavior modeling training as a whole. The independent manipulation of lecture and modeling components in my study represents the first instance in which the effectiveness of modeling may be directly compared to the effectiveness of the lecture method. The significant findings for the modeling component suggest that providing trainees with a lecture prior to modeling is not necessary for the acquisition of knowledge.

Related to this insight is the surprising finding that lecture training prior to task practice did not
improve trainee knowledge over simply receiving practice. One would expect that being told facts about computers through the lecture would at the very least improve trainee declarative knowledge, but this was not the case. One logical explanation for this result might be that the method of presentation (i.e., a videotaped lecture) is an inferior technique for presenting declarative knowledge. This would argue against the evidence put forth by Decker (1983) asserting the effectiveness of videotaped presentations.

An alternative explanation may be that the task practice was so instructive that it masked any lecture effects that were present. To explore this further, I conducted a supplementary analysis the results of which appear in Table 17. This analysis examined relations between the conditions and the practice data. The intent was to determine if trainees who heard a lecture made fewer errors in practice and/or took less time to complete the practice session than trainees who did not hear a lecture.

Error data (measured as the total number of retries across the 50-task practice session, and maintained by the computer) was the dependent measure in a 2 (modeling versus no modeling) X 2 (lecture versus no lecture) ANOVA. Both the modeling and lecture main effects were
significant (see Table 17). Analysis of the means revealed that modeling trainees (M = 79.7) made fewer errors than non-modeling trainees (M = 97.7), and lecture trainees (M = 85.5) made fewer errors than non-lecture trainees (M = 91.2). To further analyze the practice data, I also examined practice time. Using the same design, I again found both main effects to be significant. Modeling trainees (M = 35.2) were faster than non-modeling trainees (M = 42.8), and lecture trainees (M = 37.6) were faster than non-lecture trainees (M = 40.0).

The results from the analysis of the practice data appear to support the idea that lecturing was effective. That is, lecture trainees took less time and made fewer errors while practicing. Conversely, and of greater interest, trainees who did not receive a lecture spent more time practicing. The implication of this observation is that the extra practice time devoted by the no lecture trainees may have been sufficient to equate the two groups on knowledge.

Further evidence to support this contention comes from results obtained in the pilot study (see p. 71). Recall that in pilot testing the knowledge measures, 53 of the 135 subjects I pre-tested received an earlier version of the key learning points lecture one week after the initial knowledge test administration. Subsequent to the lecture, these trainees were post-tested on declarative
Table 17

**Modeling by Lecture ANOVAs for Practice Errors and Practice Time**

<table>
<thead>
<tr>
<th>Effect</th>
<th>df</th>
<th>MS</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dependent Variable = Practice Errors</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modeling (M)</td>
<td>1</td>
<td>20430.9</td>
<td>52.0**</td>
</tr>
<tr>
<td>Lecture (L)</td>
<td>1</td>
<td>2593.3</td>
<td>6.6*</td>
</tr>
<tr>
<td>M X L</td>
<td>1</td>
<td>381.6</td>
<td>1.0</td>
</tr>
<tr>
<td>Error</td>
<td>249</td>
<td>409.5</td>
<td></td>
</tr>
<tr>
<td><strong>Dependent Variable = Practice Time</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modeling (M)</td>
<td>1</td>
<td>3700.3</td>
<td>35.4**</td>
</tr>
<tr>
<td>Lecture (L)</td>
<td>1</td>
<td>455.5</td>
<td>4.4*</td>
</tr>
<tr>
<td>M X L</td>
<td>1</td>
<td>2.6</td>
<td>.9</td>
</tr>
<tr>
<td>Error</td>
<td>249</td>
<td>104.6</td>
<td></td>
</tr>
</tbody>
</table>

Note: * = p < .05. ** = p < .01.

and procedural knowledge. Results from the pilot study indicated that individuals receiving the lecture had significant gains in declarative knowledge (pre-test $M = 20.4$, post-test $M = 51.2$; $t_{36} = 11.3$, $p < .01$) and procedural knowledge (pre-test $M = 10.2$, post-test $M = 34.3$; $t_{36} = 7.9$, $p < .01$). Comparing subjects who received pre-test knowledge tests with subjects who did
not on post-test knowledge revealed a non-significant result (declarative knowledge: $t_{30.4} = 0.1$, ns; procedural knowledge: $t_{35.3} = 0.3$, ns) indicating that no test practice effects occurred. Further, I examined the possibility of subject self-selection (i.e., were lecture subjects choosing to go to the lecture because of their higher computer knowledge) by comparing subjects who post-tested and subjects who did not on pre-test knowledge. Results indicated that this was not the case (declarative knowledge: $t_{49.4} = 1.7$, ns; procedural knowledge: $t_{45.6} = 2.0$, ns). This evidence drawn from the pilot study suggests that the lecture method is effective in improving knowledge when practice is not given. The results are also congruent with the notion that practice may have made up for any decrement no lecture trainees experienced thereby resulting in no lecture effects. However, I should note that the key learning points lecture given in the pilot study was performed "live." Therefore, the possibility that subjects failed to respond to the lecture because of the medium (videotaped instruction) remains.

As a method for instructing individuals on how to perform tasks, this study suggests that the lecture was not successful at improving knowledge over practice alone. However, when I examined training method differences using performance as the dependent measure, lecture trainees were found to have higher performance accuracy than no
lecture trainees. Evidently, trainees did acquire information from the lecture that enabled them to perform more accurately, but at the same time they were unable to represent that information on either the declarative or procedural knowledge tests.

One possible explanation that exists for these findings is that the information acquired in lecture was somehow encoded by the trainees in a manner more implicit than expected. That is, the lecture imparts declarative knowledge which tends to be explicit in nature, but the pattern of results here suggests that the explicit knowledge tests were ineffective in bringing out the declarative knowledge-based lecture differences. In contrast, lecture training did have an impact on the implicit test of knowledge (i.e., performance). The practice data presented in Table 17 may shed some light on this issue since, like performance, they too are implicit in nature. Recall that both ANOVAs produced results identical to that which was found for performance accuracy; modeling and lecture trainees made fewer practice errors and required less practice time. An added advantage of the practice data concerns the possibility of evaluation apprehension. Trainees were told prior to the commencement of the performance test that their performance would be monitored. Conversely, prior to the practice session, no indication was given that their
performance was being monitored. Rather, trainees were encouraged to take their time and attempt to understand where their errors had occurred. Therefore, the analyses help to rule out the possibility that the performance data was spurious.

The analyses are encouraging to the extent that they support the viability of the lecture method although not for knowledge per se. Lecture training with practice appears to be superior to simply practicing on the performance-based implicit aspects of computer knowledge (i.e., practice and performance accuracy), but does not appear to be superior on explicit measures of computer knowledge. The results should not be construed as an indication that lecture is ineffective as a means of transferring knowledge. Rather, the lecture simply did not improve trainee knowledge over merely practicing.

I also note that the results presented here should not detract from the fact that behavior modeling is a superior training methodology. In six of seven analyses conducted on the training methods (declarative knowledge, procedural knowledge, performance accuracy, performance speed, practice errors, and practice time), modeling effects were significant and corresponding effect sizes were larger than those obtained from the lecture factor. In contrast, lecture effects were significant in only 3 of the seven analyses conducted (performance accuracy,
practice errors, and practice time). Further, the a priori contrast analysis which compared modeling to lecture indicates that, regardless of whether modeling was supplemented by lecture or not, modeling is superior to lecture.

To summarize the hypotheses focusing on the effects of training methodology, I found that modeling training improves declarative and procedural knowledge. Lecture training did not increase trainee declarative knowledge over practice alone. When procedural knowledge was measured explicitly (i.e., via a test) lecture training did not increase procedural knowledge. However, implicit measures of procedural knowledge (performance and practice) indicated that lecture training was effective in increasing trainee knowledge.

Before moving to the next section, I must comment on the counter-intuitive finding that lecture improved implicit measures (performance accuracy and practice) but not explicit measures (the procedural knowledge test). The most likely explanation for this phenomenon is that the benefit of lecture training was not realized until application of that knowledge to an actual situation was required. The post-test knowledge measures, by their nature, may not have provided sufficient context for the activation of the knowledge acquired in the lecture.
Since the key learning points lecture was heavily laden with examples and applications (see Appendix A), lecture trainees may have relied on this when working on the computer as a cue to tap their knowledge. Therefore, failure to show superior knowledge over no lecture trainees may simply be an artifact of the test type (paper and pencil versus performance).

**Training Effects on Computer Attitudes**

The inclusion of computer attitudes in my research marks a departure from previous training research which has examined attitudes toward training as opposed to content-based attitudes (Goldstein, 1993). Further, little research on computer attitude change over time has been conducted. As such, no indication of the malleability of computer attitudes over time or from training interventions has been suggested. A further demarcation point in the use of attitudes in training paradigms concerns the implications the two types of attitudes toward computers (beneficial tool beliefs and autonomous entity beliefs) have for training outcomes. Because the attitudes focus on the content of the training rather than the training pedagogy itself, some indication of the strength of the attitudes for improving knowledge is both warranted and of interest to researchers attempting to design future computer training systems. In the section that follows, I examine the results from my
research which address the issues of computer attitude change and attitude effects on learning.

An overview of the attitude change hypotheses reveals that the results were very different for the two types of attitudes. That is, the effects of training on trainee changes in their beliefs in the computer as a beneficial tool were quite different from the changes observed in trainee autonomous entity beliefs. Across all groups, beneficial tool beliefs did not increase as a result of the training program. In contrast, beliefs that the computer is an autonomous entity decreased from pre-test to post-test.

Some light can be shed on these findings if one considers the theoretical model put forth by Brock and Sulsky (in press). In this research, we tested a hypothesis put forth by Rafaeli (1986) that beliefs in the computer as an autonomous entity were a result of unfamiliarity with computers whereas beliefs in the computer as a beneficial tool were a result of experience with the technology. Our results supported the idea that autonomous entity beliefs preceded computer use whereas beneficial tool beliefs were a result of computer use. In the context of this research, the three hour training session gave trainees sufficient exposure to allay their fears and anxieties about computers (autonomous entity beliefs), but did not provide enough experience with the
machine for trainees to form beliefs in the computer as a beneficial tool. I would expect that, if exposure to the computer continued, trainees would begin to formulate more beneficial tool beliefs as they consequently became aware of the utility of computers.

The finding that modeling training did not improve trainee computer attitudes is also congruent with Brock and Sulsky's (in press) model. Since modeling training does not necessarily guarantee more exposure to computers, the increase in beneficial tool beliefs should not be that great. In contrast, my results found that lecture training significantly increased trainee beneficial tool beliefs. Although I did not hypothesize this to be the case, a consideration of this phenomenon within the model presented above also helps to explicate the finding. The lecture process can be seen as an hour-long sales pitch for the utility of computers. Rather than explaining how commands on the computer worked (as in the modeling training), the lecture told trainees what could be done with computers. Therefore, trainees were informed as to how the computer was a beneficial tool.

By the same processes that essentially "sold" trainees on the idea that computer are beneficial tools (i.e., the key learning points lecture), a very different result occurred for trainee autonomous entity beliefs. Recall that, across all groups, the training session
reduced trainee autonomous entity beliefs. The main effect for time, however, is qualified by a significant interaction between lecture training and time. Analysis of the simple effects revealed that trainees who did not receive a lecture had decreases in autonomous entity beliefs whereas lecture trainees remained unchanged. Thus, the same method that evidently "sold" trainees on the idea that computers were beneficial tools may have inadvertently scared some trainees away.

Since the assertion that a lack of something (in this case lecture training) can cause changes in attitudes seems somewhat dubious, I explored this finding further. Figure 3 graphically depicts the pre- and post-test autonomous entity means for the four training conditions. Several observations concerning the graph are noteworthy. First, the modeling plus lecture group appears to differ from the other training conditions in that the slope of the line from pre-test to post-test is nearly flat. Conversely, all other groups showed a downward trend. Thus, when collapsed together, a main effect for time is evident. Second, the odd behavior of the modeling plus lecture training condition appears to have contributed to the interaction between time and lecture. Collapsing across the two lecture groups reveals that the slope of the line from pre-test to post-test is approximately flat. Conversely, the two no lecture groups show a decrease in
autonomous entity beliefs from pre-test to post-test. This observation is congruent with the simple effects tests which indicated that only the pre-post effect for the no lecture groups was significant. A third and final observation concerns the large decrease in autonomous entity beliefs which occurred in the modeling only group. Consistent with a visual inspection of Figure 3, post-hoc
Scheffé tests revealed that modeling only trainees had significantly greater decreases in autonomous entity beliefs than any of the other three training conditions.

Two comments are in order concerning the interpretation of these findings. First, I speculate that the general trend indicating a decrease in trainee autonomous entity beliefs was moderated by the lecture rather than a lack thereof. As I suggested above, the lecture training may have inadvertently intimidated trainees by presenting a large amount of information in a short period of time. Congruent with this expectation, trainees who merely received modeling had the greatest decreases in autonomous entity beliefs. The fact that the modeling plus lecture and lecture only groups showed no change in autonomous entity beliefs also supports this contention. Concerning the modeling plus lecture group, the combination of lecture (which may have intimidated trainees thereby increasing autonomous entity beliefs) and modeling (which eased trainee fears thereby decreasing autonomous entity beliefs) may have resulted in a canceling out of effects. Hence no change occurred from pre-test to post-test.

A second comment regarding my interpretation concerns the fact that my post-hoc examination of differences between the four training groups was not based on a significant interaction. Although the analysis used was
the most conservative of estimates among post-hoc procedures, and the obtained three-way interaction was narrowly non-significant, the reader may wish to interpret these results with caution.

**Attitude Effects on Computer Training**

The final analyses in the study moved from examining attitude change toward examining the effects of trainee attitudes on computer knowledge acquisition. As I alluded to previously, attitudes toward training have typically been unsuccessful in predicting learning (Goldstein, 1993). However, the attitudes measured in this study relate directly to the content of the training. This appears to be a likely explanation for the findings indicating that attitudes do, in fact, predict trainee knowledge change. Also of interest is the observation that, across all forms of knowledge measurement (declarative knowledge and procedural knowledge measured explicitly by test and implicitly through performance), the results were consistent. Autonomous entity attitudes significantly predicted computer knowledge acquisition whereas beneficial tool beliefs did not.

These results also tie in nicely with the Brock and Sulsky (in press) model. Given that the level of experience with computers among the trainees was relatively low ($M = 1.9$ years of exposure to computers), it is not surprising that autonomous entity attitudes
would be prevalent. If, as Brock and Sulsky (in press) suggest, autonomous entity attitudes precede computer use and beneficial tool beliefs form following the use of computers, then inexperienced trainees are likely to be undifferentiated on beneficial tool beliefs prior to coming in contact with the machines. As such, beneficial tool beliefs would lack the predictive validity for identifying changes in knowledge as a result of a training intervention. This appears to be the most likely explanation for the findings in this study. Trainees high in autonomous entity beliefs learned less than trainees low in autonomous entity beliefs. Conversely, beliefs that the computer is a beneficial tool theoretically form from continuous exposure to and work with computers. If this is the case, pre-intervention assessment of beneficial tool beliefs would not be predictive of changes in knowledge because trainees did not have the exposure to computers necessary for the formation of those beliefs. Again, the findings in this study appear to support this explanation.

It is interesting to note that the analysis of the final sub-hypothesis examining knowledge acquisition as a result of prior attitudes toward computers followed a pattern similar to that which was observed in reporting the previous results; different results occurred when performance accuracy was used as the dependent measure.
Results for the declarative and procedural knowledge tests were consistent in indicating that the two attitudes did not interact in their prediction of knowledge acquisition. However, an interaction was found when performance was used as the dependent measure.

Although the obtained interaction was consistent with my hypothesis (beneficial tool beliefs moderated autonomous entity beliefs such that the beliefs only impacted learning when beneficial tool attitudes were high), the results should be interpreted with caution. Since there was no pre-test measure of performance, there was no way to control for prior ability with computers. Based on the strong relationship between procedural knowledge and performance, one alternative would be to assume that pre-test performance was at a level similar to the level of pre-test procedural knowledge. Following this assumption, I conducted an exploratory analysis using performance accuracy as the dependent measure and pre-test procedural knowledge as a covariate. Attitudes were entered in the second step followed by the interaction term. The interaction of the two attitudes on computer performance (controlling for procedural knowledge) was found to be non-significant ($\beta = -.01$, $t_{228} = -1.3$, ns). This analysis suggests the possibility that the significant interaction between computer attitudes and computer performance may have been due to a priori
performance differences that were related to computer attitudes.

A final explanation for why attitudes did not interact in their effect on knowledge also concerns the dependent measures examined. Previous evidence for the finding of an attitude interaction was obtained using computer use as the dependent measure (Brock, 1993). Since using the computer and knowing how computers work are two different things, it may be that the two dependent variables are more different than they are alike. If this is the case then attitudes may interact to determine whether an individual uses computers but not to determine how much they learn about computers.

This summary of the findings of interest in the study suggests numerous implications for future research in the social and cognitive learning realms. Further, the study's basis in training coupled with the focus on computers and technology has ramifications for practitioners interested in improving employee interaction with technology. Below I present these ramifications in an attempt to broaden the scope of this research.

Implications, Limitations, and Future Research

At the onset of this research, I stated two main objectives which I intended to explore. The first objective was to contribute to the psychological research on technology in an effort to narrow the gap between the
rapid expanse of technology in the workplace and our understanding of that technology. This is perhaps the most applicable area of my research in that my results relate directly to the field of technology. The two foremost areas in which I saw a need for further exploration were training in technology and attitudes toward that technology. In essence, I examined how individuals react to technology and, at the same time, how they adapt to and learn about that technology.

The second objective in my research was to examine how readily components of two popular learning theories (cognitive and social learning theory) could be integrated in a single design. This aspect of my research sought to examine the effectiveness of behavior modeling training in explicating the declarative/procedural knowledge distinction, and how well the two knowledge constructs aided in the explanation of the phenomenon of modeling. My research explored behavior modeling training from social learning theory, and the declarative-procedural knowledge distinction from cognitive learning theory.

I also explored specific issues within each of the theories that are worthy of note here. For behavior modeling training, I examined the interplay between lecture and modeling in an effort to determine what aspects of the training methodology as a whole were most effective. In examining declarative and procedural
knowledge, I sought to determine: 1) how the two concepts could be operationalized within the context of computers, and 2) how well procedural knowledge could be explicated through the use of measures other than performance (viz., a paper and pencil test). Finally, since these components of the two theories were examined in the same study, I was able to address the feasibility of an integration of the two theories.

In the sections that follow, I bring out the implications of my research for these two major objectives (technology research and the learning theory integration). When appropriate, suggestions for future research are provided. Discussion of the limitations in the research are interwoven within the dialogue at appropriate points.

**Technology Training.** The primary implications for future research in the area of technology which may be drawn from my research center around training and attitudes. Concerning training, my research supports that of Gist and her colleagues (1988; 1989) in suggesting that the best way to learn about computers is to watch someone else execute the tasks first. The finding that modeling was effective in improving learning over practice alone is not surprising. In my personal experience working with individuals on computers, I find that the skills needed to perform tasks on the computer are much easier shown than they are told.
Future research which explores the various components of technology training will, I believe, discover that the nature of the topic is what drives the success of modeling. That is, technological manifestations such as computers have certain characteristics that readily lend themselves to modeling. Procedures for performing operations on the computer (e.g., copying a file) are unlikely to change from computer to computer thereby making the acquired knowledge widely applicable and relatively stable. Further, computer operation relies heavily on the visual medium. The interaction between human and machine takes place on a monitor where all feedback from the computer occurs. As such, methods like modeling are ideal for teaching individuals how to use computers since they offer so much information on how to interact with the computer and what to expect from the computer in different instances.

This research possesses numerous avenues and implications for organizations interested in technology training. Foremost, individuals interested in the design of training seminars can be assured that methods which use a model who demonstrates the desired behavior are an effective means of transferring training content. Additionally, my research suggests that videotaped presentation of the model is an effective means of transferring the information to trainees. This suggests
that organizations can save time and resources normally dedicated to live seminars in deference to a prepared videotape. A final organizational implication concerns the need for future research into the training content being modeled. My intervention utilized a widely applicable, but not so user-friendly, software package (i.e., DOS). Care should be taken to note how different types of software may be more or less amenable to the modeling process. Specifically, more technical software (e.g., drafting programs, statistical packages) may require a greater emphasis on content before the execution of a modeling intervention. Additionally, more computer-intensive programs may require more input from the model. In my study, the model was simply a guide who showed trainees what the various commands looked like on the computer. Another implication then, is that software with an extensive degree of computer involvement may require more of a "human supplement" in the modeling process.

Attitudes Toward Technology. The second aspect of my research related to technology concerned the ways trainees react to computers. Several implications lie in the results obtained from the attitudinal findings. In general, the notion that trainees hold two distinct attitudes toward computers put forth by Brock and Sulsky (in press) was supported in this research. Beliefs in the computer as a beneficial tool and beliefs in the computer...
as an autonomous entity had different implications for trainees. For example, I found that lecture training improved beneficial tool attitudes but did not change autonomous entity attitudes.

The idea that different types of training can hold different implications for attitudes deserves further mention. The pattern of results which appeared in my research suggested that lecture training improved beneficial tool attitudes but was not generally successful at improving learning. Thus, beliefs in the computer as a beneficial tool appear to be a result of trainees being "sold" on the idea that computers are useful. Further, beneficial tool beliefs were not found to be predictive of knowledge acquisition.

In contrast, autonomous entity beliefs hold a more interesting set of implications. Autonomous entity attitudes significantly predicted knowledge acquisition suggesting that trainees with low levels of these attitudes were able to learn more. Two implications for future research lie in this finding. First, if my research is generalizable to field settings, organizations could use an assessment of employee attitudes toward computers as a means of determining which employees would be most successful in future technology training programs. Related to this, the assessment of trainee attitudes could also serve as a diagnostic tool to determine where
resistance to the introduction of new forms of technology may lay.

A second implication directed toward researchers in technology pertains to determining the best method for reducing negative computer attitudes. The intervention utilized here was successful in reducing autonomous entity beliefs across all conditions. Further, conditions that did not receive a lecture showed a significant decrease in autonomous entity beliefs whereas lecture trainees showed no change. Therefore, the lecture convinced trainees that computers are a beneficial tool and an autonomous entity (or at least the lecture aided in maintaining trainee autonomous entity beliefs).

The fact that modeling did not significantly reduce trainee autonomous entity beliefs is discouraging. However, given that this is the first attempt made at identifying the best methods for reducing these attitudes, the interested researcher should consider the possibility of other interventions which may more directly temper trainee autonomous entity beliefs. One suggestion would be to design a modeling intervention that places more emphasis on "humanizing" the computer. My modeling intervention stressed the actions taken on the computer and down-played the human component. Future interventions might utilize the model in a more active role where
trainees could infer more attitudinal information than was presented here.

A final suggestion for future research concerns the interaction of the two attitudes on knowledge acquisition. I have already addressed the possibility that the weak support for this hypothesis may have been indicative of an inappropriate dependent measure (i.e., attitudes may interact on computer use but not on knowledge). However, future research could be instructive on this point if it were to examine the phenomenon in other training settings. Perhaps the lack of an interaction was in part due to the varying effects lecture and modeling training had on trainees.

The preceding pages suggest numerous avenues for future research in technology. I have presented a few of those which pertain directly to training and attitudes and which fall directly out of the results from my research. I should note that the implications herein should not be considered exhaustive. My research focused on computer attitudes and training which represent two of the many other aspects of computers that could be addressed (e.g., the introduction of new technology into organizations, the effects of computer monitoring, ergonomic issues in human-computer interaction). Additionally, computers are just one form of technology, research that explores other areas (e.g., telecommunications, robotics, artificial
intelligence) is strongly recommended. Given that the psychological research on technology and its effects on humans is in its infancy, much remains to be done in exploring these other aspects of technology. A third caveat relates to the scope of the intervention. My research examined the reactions and learning of trainees under relatively individualized conditions. Other research might focus more at the group or systems level when examining technology and its impact on workers.

Social Learning Theory. The aspect of my research on behavior modeling training which sets it apart from previous examinations concerns the partialling out of the lecture and modeling components of the methodology. In so doing, I was able to determine that modeling alone was just as effective as having modeling supplemented by lecture. This finding adds to the extant literature on the effectiveness of modeling for computer training by showing that the lecture component is not as critical to the modeling process as was once thought (e.g., Gist et al., 1988; 1989).

I see two main implications of the finding that modeling does not require a lecture component. First, future research on behavior modeling training may be more parsimonious in design. That is, modeling interventions may not need to include a "tell" component in the "tell and show" package. In many instances, especially in
practical settings, this "trimming" of the training methodology could result in shorter training programs (which are therefore more cost effective) or could allow for more time to be spent in other aspects of the methodology (e.g., extended task practice, more elaborate modeling).

A second implication pertains to the generalizability of the finding to other research domains. As I have noted on numerous occasions, there are specific aspects of the training content (computers) that may make results from the research different from those found in other training settings. It may be the case that the learning process relies so heavily on interacting with the machine that attempts to acquire knowledge away from the computer are hampered in their effectiveness. Therefore, lecture training may be too far removed from the "hands-on" training that is so valuable for computer knowledge acquisition. Future research should examine modeling only training in other domains to see if the results obtained here generalize.

Related to the observation that behavior modeling training does not necessarily require the inclusion of a lecture sub-component, the nature of the training design (i.e., a two by two crossed design) allowed for several unintended discoveries. Specifically, a comparison of the modeling plus lecture and modeling only groups provides
some interesting insights into the dynamics operating in the modeling process. Individuals in the modeling only training condition had greatly reduced autonomous entity beliefs from pre-test to post-test (see Figure 3). In contrast, individuals with both modeling and lecture showed no change in their beliefs in the computer as an autonomous entity. The modeling only condition may be construed as a more pure form of how behavior modeling training was originally conceptualized. That is, trainees in this condition are simply shown how the tasks are performed and then given an opportunity to practice. Traditional modeling interventions have shown that the technique is an effective means of reducing fears and anxieties (Bandura, et al., 1969). Thus if we construe the modeling only group to be the best characterization of how true modeling operates, then it is not surprising to find that trainees in this group showed the greatest reduction in computer-related fears and anxieties (i.e., autonomous entity beliefs). Given that this finding was unintended, future research should examine modeling only training to determine if the effects on autonomous entity beliefs reported above are indeed robust.

If the results presented here are replicated, the implication is that researchers and practitioners should be careful to design their training interventions to fit the needs of their trainees. If a reduction in negative
attitudes toward computers is desired, my research suggests that a straightforward modeling approach (without lecture) is warranted. On the other hand, drawing from evidence presented earlier, a lecture approach is warranted if the goal is to increase trainee positive attitudes toward computers (i.e., beneficial tool beliefs).

A further note is in order concerning the modeling plus lecture training condition. I expected trainees in this condition to acquire more knowledge than trainees in any other condition. However, the results suggest that these trainees performed at a level approximately equivalent to that of the modeling only group. I have suggested that the lecture intervention’s failure to produce significant changes in trainee knowledge may have been a result of the quality of task practice; any decrement the no lecture trainees faced was compensated for by the practice session. This observation may also be extended to the findings for the modeling plus lecture condition. Future research could contribute to a better understanding of why the lecture did not aid modeling by examining other conditions not explored in this study. Specifically, the addition of several training groups which examined the effectiveness of the two training interventions (modeling and lecture) when coupled with a no practice condition might help to determine the
contribution of practice to the training outcomes. Here I am proposing research which manipulates lecture, modeling, and practice in its design rather than lecture and modeling alone. A study such as this could, therefore, more directly determine whether the effectiveness of behavior modeling training can be improved by the addition of a lecture sub-component.

Cognitive Learning Theory. In utilizing the cognitive learning theory literature, the most significant contribution of my efforts lays in the means by which I have characterized the constructs of declarative and procedural knowledge. Although examinations of this knowledge distinction are replete in the cognitive literature (see Weiss, 1990) its use in industrial psychological training contexts has been limited to a very few examinations (e.g., Kanfer & Ackerman, 1989). Therefore, the general contribution of this effort was twofold: 1) I presented a utilization of the declarative and procedural knowledge constructs in a more applied context, and 2) I operationalized the construct of procedural knowledge in a manner different from that previously conceived.

The measurement of procedural knowledge in this research marks a first attempt at examining an alternative approach to understanding the construct. The long-held demonstrative measure of procedural knowledge has been
performance. I propose that performance is an imperfect measure of procedural knowledge because it fails to directly measure knowledge. Performance measures the outcome of procedural knowledge and is therefore measuring a result rather than the knowledge itself.

In developing the knowledge measures, I found my assessment of the declarative knowledge test to be entirely congruent with previous research measuring the construct (e.g., Anderson, 1982; Kanfer & Ackerman, 1989). That is, tests of what facts a subject knows about a task are administered by asking the subject that very question: "What do you know?" However, when I turned to examinations of procedural knowledge measurement I found the approach to be much less direct. Arguably, the fact that procedural knowledge is a more complex construct than declarative knowledge supports the contention that procedural knowledge measurement should be more involved. However, to my knowledge, no attempts at directly measuring procedural knowledge have been made. This is exactly what I proposed to do. I measured trainee procedural knowledge by asking the trainees how they would perform tasks on the computer.

A major limitation of my research lies in my inability to effectively discriminate between declarative and procedural knowledge. Principally, this failure lies in the non-significant lecture effects observed for both
declarative and procedural knowledge. Because this research marks the first time that procedural knowledge has been measured explicitly, I am unable to utilize past research to suggest why the study failed to show differential results across training conditions. My suspicion is that the blame for why the two knowledge tests could not be differentiated lies with the new method I used to measure procedural knowledge. Further, I believe that the explanation for why the procedural knowledge test was not distinguishable from the declarative knowledge is test content-based rather than construct-based. Nevertheless, future research should examine this means of measuring procedural knowledge to determine if the limitation lies in the study or in the theory. Below I discuss some implications for improving upon procedural knowledge assessment in future research.

The primary criticism that can be leveled at this method of measuring procedural knowledge is that procedural knowledge is typically characterized as difficult to describe (Best, 1989). Therefore, any attempt at directly measuring procedural knowledge will fall short in explicating the implicit components of the knowledge. Two arguments may be made in defense of my application of procedural knowledge. First, from a content perspective, the focal task (computers) is such that the aspects of knowledge which are procedural are
also more explicit. That is, computer procedural knowledge is, based on my personal experience, more easy to describe.

A second, empirically-based, argument is that my new measure of procedural knowledge and the more traditional measure of the construct (performance) were highly related. This observation lends itself to numerous suggestions for future research. For example, the high relationship between the two measures suggests the possibility that some components of implicit knowledge are being addressed explicitly. Future examinations might address whether proceduralized knowledge has been compiled in a manner that is **accessible** in memory (i.e., explicit). Still another implication of the relationship between the procedural knowledge test and performance pertains to how the two might be combined to optimize construct explanation. Here the methods used to achieve the goal could be strictly statistical (e.g., MANOVA), or theoretical.

Future research which examines the measurement of procedural knowledge might also benefit from a more thorough examination of the content of the items on the declarative and procedural knowledge tests. That is, what aspects of knowledge are addressed in the procedural knowledge test that make them different from the declarative knowledge test? Given that my implementation
was narrow in scope (i.e., ten commonly used DOS commands), the content of the two knowledge tests had substantial overlap (see Appendices D and E). This suggests that any differences between the test must be real (i.e., due to the type of knowledge being tapped rather than the content of the items). However, it could also be argued that procedural knowledge items are merely more difficult or more involved demonstrations of declarative knowledge. That is, the items simply require the integration of several pieces of declarative knowledge rather than a transformation of that knowledge into a procedure. Future research examining these issues could contribute greatly to clarifying this limitation.

Still another implication along a similar line pertains to the overlap between items on the procedural knowledge test and the performance measure. Although the content of the items on the two measures was markedly similar (see Appendices E and F), future research might benefit from an implementation that directly addressed each component of a procedural knowledge test in a subsequent performance test. An examination such as this could also explore the relative effectiveness of the two assessment in terms of providing cues to subjects. There remains a possibility in my research that subjects who were asked to recall how tasks are performed in the procedural knowledge test were inadvertently
directed toward which components of the training were more important for the performance test. Therefore, the order of presentation could be staggered in future designs to determine if one test aided performance on another test.

To summarize thus far, I have presented implications for future research in behavior modeling training and in examinations of the declarative-procedural knowledge distinction. I conclude my discussion of the study's implications with a dialogue on the theoretical integration of cognitive and social learning theory in an effort to address this objective of my research.

Integrating the Learning Theories. As stated above, a large portion of the present effort directed itself at the implications of an integration of cognitive and social learning theory. At its most basic level, the results from my research did not paint the clear picture I would have liked. The training interventions (modeling and lecture) produced similar results on the two knowledge variables making distinctions between the two on the basis of training methodology weak.

My initial explorations into the two learning theories brought me to two simple links which formed the basis of this integration. Lecture training appeared to be a process of transmitting declarative knowledge (i.e., teaching them what) whereas modeling training involved the process of conveying procedural knowledge (i.e., teaching
them how). Ideally then, the results of my research would show significant lecture effects for declarative knowledge but not procedural knowledge, and significant modeling effects for declarative and procedural knowledge (since declarative knowledge must precede procedural knowledge). Unfortunately, the pattern of results did not hold up for the lecture method. Lecture trainees did not acquire greater declarative knowledge over trainees without benefit of the lecture. Further, contrary to my predictions, lecture trainees had more knowledge than no lecture trainees when performance was used to assess procedural knowledge. On the other hand, modeling training did lead to an increase in both types of knowledge thereby affirming my expectations.

Although the results from the study suggest that my attempted integration of the two theories was not without its faults, there are components of the results which may hold some merit. I propose that the primary disconfirming finding, that lecture training did not contribute to declarative knowledge gains, is a result of limitations in my research. As I have already mentioned, the design of my study was such that all subjects received practice and this feature has been suggested as an explanation for why the lecture method did not improve knowledge scores. The quality of practice may have been so great that it washed out any existent lecture effects. Recall that the
acquisition of procedural knowledge occurs through repeated application of declarative knowledge (i.e., practice). Therefore, one explanation for the non-significant lecture findings is that the no lecture trainees were able to infer declarative knowledge from the practice session thereby equating themselves with the lecture trainees on post-test declarative knowledge. This inferential knowledge acquisition may also have been sufficient to make differences between lecture and no lecture trainees nil on the procedural knowledge test. However, as evidenced by the significant findings for performance accuracy, lecture trainees did gain some knowledge that trainees without the benefit of the lecture did not.

The proposed integration of the two learning theories can best be summarized by considering the three stages each has. The first stage in cognitive learning theory (declarative knowledge) was theoretically likened to the lecture portion of behavior modeling training. The second stage, knowledge acquisition, is subsequently associated with task practice in modeling training. The procedural knowledge stage is the final level for trainees in cognitive learning theory. Its analog in social learning theory is the application of modeled behavior. I believe that the key to the success of the marriage between these two theories lies in clarifying the distinctions between
the three stages. I see three main limitations of my research in which future research could contribute to a clarification of the distinctions between these stages. First, as discussed above, the study did not measure trainee knowledge following the lecture. Future research could make use of this assessment as a means of eliminating the possibility that practice masked the effects of lecture. Second, the critical interim phase in knowledge acquisition (knowledge compilation) deserves more careful examination in future research. Subjects in my research received not quite one hour of practice in which to proceduralize their declarative knowledge. The implication here is that, with more practice, the distinction between declarative and procedural knowledge might become more noticeable. With more time to practice, subjects could begin to formulate more elaborate procedural knowledge. A third limitation to my research concerns the need for a better operationalization of the two knowledge measures. Specifically, the high relationship between declarative and procedural knowledge might be lowered by a clarification of the differences between the two types of knowledge. Clearer operationalization of the measures in the construct development phase might aid in distinguishing the two stages of knowledge acquisition from each other. Alternatively, this distinction may be more apparent in
research which utilizes content domains which possess characteristics that make the knowledge constructs more distinguishable than they are for computers.

Based upon the discourse above, it is clear that my attempt at integrating the components of cognitive and social learning theory is not without its limitations. However, the strong effects for modeling as a means of acquiring procedural knowledge can not be discounted. In essence, this research has identified one possibility for why behavior modeling training is so effective. The methodology supports and encourages all those tenets of cognitive learning theory which are deemed necessary for the acquisition of procedural knowledge. Therefore, concerning modeling, the marriage between the theories does not appear to be a difficult one. The results from this study suggest that future examinations into how the integration between the two theories might be improved upon is certainly warranted.

Conclusions

In the broadest sense, the present research has attempted to shed some light on the processes involved in acquiring skills related to technology. Because research in the area of technology training and human-computer interaction is so sparse, the present study offers numerous avenues for further pursuit in the research realm. The paucity of research examining human reactions
to computers (both affective and behavioral) has been noted before (Turnage, 1990). In a sense, this effort attempts to answer that call. Below I provide a brief summary of the answers obtained in my research.

The results from my research suggest that behavior modeling training is an excellent methodology for training individuals on how to use computers. Further, I determined that supplementing behavior modeling training with a lecture was not necessary for trainee learning. Lecture training was found to be an effective method using implicit measures of learning (performance and practice), but did not differentiate between groups on explicit knowledge tests (declarative and procedural). Further, I found some evidence for the convergent and discriminant validity of declarative and procedural knowledge. Concerning the latter (discriminant validity), the support was weak. I expect that the problem of discriminating between declarative and procedural knowledge lies in the heavy process-dependence the two constructs possess, particularly in the computer domain.

As a whole, the results revolving around the application of computer attitudes to computer training are also promising. Autonomous entity beliefs were significantly (and negatively) related to computer knowledge acquisition. Further, the training intervention was effective in decreasing autonomous entity beliefs.
Beneficial tool attitudes were improved in conditions where trainees received a lecture on computers. Autonomous entity beliefs, on the other hand, decreased when trainees did not receive a lecture.

The summary presented above suggests a profusion of avenues for future research. My examination of the relationship between cognitive and social learning theory can serve as a springboard for research that attempts to determine how well the two theories operate together. If the two theories are indeed compatible, then it appears that through two distinctly different research foci (macro and micro) we arrive at the conclusion that the fundamental tenets are in fact very similar. Still, further exploration is warranted into the viability of this cognitive-social learning theory marriage in other realms besides computers and technology. My concluding remarks turn to this realm and leave behind the theoretical discussions of learning.

The so-called computer age in which we now live has numerous doorways for human advancement. Unfortunately, our understanding of how to approach these doorways has fallen short. My research presents an attempt at exploring how individuals learn to use computers and how they feel about the machines. Whether we feel that computers are assuming control over our lives or are simply a useful instrument for the betterment of
mankind, one thing is certain: computers and technology are here to stay.

With this in mind, we should attempt to explore those aspects that help us to understand this eminent presence. Through my exploration of how individuals learn to use technology, I have discovered that the most salient component is a need to observe how others behave when interacting with the machine. Perhaps this is the "high-touch" component in a "high-tech" world which Naisbitt (1982) spoke of as so important to our needs as humans in the technology-laden world of the future. Hereafter, inquiries into the nature and consequences of technological advancement should continue to examine and develop these "high-touch" components in the hopes of improving the interaction between human and machine.
References


164


malleability. Academy of Management Review, 17, 183-211.


Appendix A

Outline for Key Learning Points Lecture

I. OVERVIEW
   A. Description of the computer
   B. Important keys on the keyboard
   C. Internal and external drives
   D. Introduction to DOS

II. FILES AND FILE STORAGE
   A. Filing cabinet analogy
      1. Drives like filing cabinets named A, B, or C
      2. Directories like file folders in cabinets
   B. Sample directory tree (filing cabinet)
      1. Root Directory
      2. Sub-directories
         a. underneath the root directory
         b. underneath other sub-directories
   C. File folders have files in them
      1. File naming
         a. up to 8 letters & numbers plus option for 3
            more letters or numbers
      2. Examples of filenames
   D. The DOS Prompt
      1. (drive letter):\(directory)\>
      2. DOS syntax explained (command plus parameters)

III. WORKING IN DOS
   A. The DOS Prompt - (drive letter):\(directory)\>
B. Entering a command - type the command, use backspace to correct (if necessary), press <ENTER>

C. Format of commands - drive:\directory\filename.ext
1. Can omit drive/directory if in the one you want
2. Can omit filename.ext for entire directory

IV. SIMPLE DOS COMMANDS

A. CLS - clear screen

B. DATE
1. Finding out the date
2. Setting a new date
3. Examples

C. TIME
1. Finding out the time
2. Setting a new time
3. Examples

V. VIEWING FILES - The DIR and TYPE commands

A. Viewing the contents of directories (DIR)
1. Formats
   a. DIR view all files in current directory
   b. DIR drive:\directory
   c. DIR drive:\directory\filename.ext
2. Information you get from DIR
   a. filename and extension
   b. size (in bytes)
   c. creation date and time
   d. bytes in directory and on entire drive
3. Switches (the "/" means a switch is in use)
   a. /P = do a directory one page at a time
   b. /W = view only file names in a wide format
4. Global wildcard characters
   a. * for all
      (1) all files must be *.*
      (2) *.x = all files with an "x" extension
      (3) x.* = all "x" files with any extension
      (4) x*.x = all files beginning with "x"
   b. ? for a character
      (1) xyy.a?
      (2) ??progs.dat
   c. Examples
5. Use DIR as feedback for other commands

B. Viewing the contents of a single file (TYPE)
1. Formats
   a. TYPE drive:\directory\filename.ext
   b. TYPE filename.ext
2. Examples

VI. REARRANGING FILES

A. The RENAME (REN) Command
1. Formats
   a. REN drive:\directory\oldname.ext newname.ext
   b. REN oldname.ext newname.ext
2. Can't rename to a new directory
3. Can't rename to a file that already exists
4. Much faster than copying the file to a new name
5. If in the directory where the file is and you want the newly named file in that directory, you can omit the drive and directory parts of the command

B. The DELETE (DEL) or ERASE Command - equivalent

1. Formats
   a. DEL drive: \directory\filename.ext
   b. DEL filename.ext

2. DEL and the wildcard characters
   a. DEL *.XXX - all files ending in "XXX"
   b. DEL MYPROG.* - all "MYPROG" files
   c. DEL *.* - be careful, deletes all files

C. The COPY Command

1. Formats
   a. COPY drive: \directory\source.ext drive: \directory\destiny.ext
   b. COPY drive: \directory\source.ext drive: \directory
   c. COPY drive: \directory\source.ext drive: \directory
   d. COPY source.ext drive: \directory\destiny.ext
   e. COPY source.ext drive: \directory

2. You can overwrite the destination file

3. COPY and the wildcard characters
   a. COPY C:\PROG\MYFILE.? C:\NEW\NEWFILE.?
   b. COPY C:\PROG\MYDAT C:\DOS\MYDAT2
   c. COPY C:\PROG\*.XXX C:\NEW
d. COPY *.DAT C:\
e. COPY *.* C:\NEW
f. COPY TOM HARRY

VII. REARRANGING DIRECTORIES

A. The CHDIR or CD command - changing directories

1. Formats
   a. CD dirame - go down one directory level
      (will get an error if directory does not exist)
   b. CD\dirame - go to a directory that has the same parent directory as the one you’re in
   c. CD\ - go to root directory
   d. CD.. - go up one directory level

2. The concept of directory paths - moving around the directory tree

3. Examples of moving around the directory tree

B. The MKDIR or MD command - making a directory

1. Formats
   a. MD dirame - make dir one level down
   b. MD\dirame - make dir as specified

2. If a directory exists with the same name, you cannot make a new one

3. Makes a directory one level below the current directory
   a. most computers have just the root directory and its sub-directories
   b. can have many more levels (tree analogy)
C. The RMDIR or RD command - removing a directory

1. Formats
   a. RD directory - remove a directory one level below the current directory
   b. RD\dirname - remove directory following the specified path

2. Directory removal procedure
   a. remove all files from directory to be removed
   b. remove all sub-directories from the directory to be removed (including any files in those directories)
   c. execute the remove directory command
Appendix B

Task List for Practice Session

For the following, the current directory (blank for the root directory) and the correct answer are placed in brackets next to the task statement. Equifinality exists with respect to many of the commands. In these instances, the optimal answer is given (credit for a correct response in the practice was given only for optimal responding). Additionally, all task practice was performed on floppy disk drive A:. As such, no drive specifications are given or required.

Simple DOS Commands

The following commands center around the simple commands used in the DOS environment. They include clearing the screen, finding the date and time on the computer, and changing the date and time on the computer.

- Clear the screen. [, CLS]
- Identify the current date on the computer. [OLD, DATE]
- Change the time on the computer to 12:15 a.m. [, TIME 12:15]
- Change the date on the computer to May 3rd, 1992. [NEW, DATE 5/3/92]
- Change the time on the computer to 2:31 p.m. [NEW, TIME 14:31]
- Change the time on the computer to 9:15 a.m. [, TIME 9:15]
- Change the date on the computer to March 2nd, 1993. [DATE 3/2/93]
- Change the date on the computer to October 31st, 1982. [FRED, DATE 10/31/82]

**Viewing Files**

The following set of commands are designed to provide you with practice on how to view the contents of directories on the computer and how to view the contents of individual files on the computer.

- List the files contained in the PROG directory. [PROG, DIR]

- Display the contents of the file MYDAT.TXT in the OLD directory. [OLD, TYPE MYDAT.TXT]

- Display the files in the PROG directory. [FRED, DIR \PROG]

- Display information on the file ME.TXT in the directory OLD. [NEW, DIR \OLD\ME.TXT]

- Display the contents of the file ME.TXT in the LETTERS directory. [NEW, TYPE \LETTERS\ME.TXT]

- List the files with a TXT extension in the OLD directory. [OLD, DIR *.TXT]

- Display all the files that start with the letters ME that are in the NEW directory. [NEW, DIR \NEW\ME*.*]

- Display all the files in the root directory using the wide format. [OLD, DIR \ /W]
- Display the contents of the file YOURDAT.TXT in the OLD directory. [FRED, TYPE \OLD\YOURDAT.TXT]
- Display the files in the OLD directory one page at a time. [LETTERS, DIR \OLD /P]
- List the files in the PROG directory that have the letter A in the third position of the extension. [NEW, DIR \PROG\*.??A]
- Using the wide format, list all files with a filename of MYDAT in the NEW directory. [NEW, DIR MYDAT.* /W]
- Display the contents of the file BILL.BAT in the root directory. [PROG, TYPE \BILL.BAT]
- Display all files in the NEW directory that have a TXT extension one page at a time. [OLD, DIR \NEW\*.TXT /P]
- List all files with a filename of JACK in the LETTERS directory. [PROG, DIR \LETTERS\JACK.*]

Rearranging Directories

The following commands instruct you on how to rearrange directories on the computer. This involves moving from one directory to another, creating directories, and removing directories.
- Create a first level directory called BOB. [, MD BOB]
- Remove the directory called BILL. [, RD BILL]
- Go to the directory called BOB. [BILL, CD \BOB]
- Go up one directory level. [DATE\ONE, CD ..]
- Create a second level directory called DAY underneath the first level directory called NIGHT. [, MD NIGHT\DAY]
- Go to the OLD directory. [NEW, CD \OLD]
- Go to the directory DATA which is underneath the first level directory called BOB. [BILL, CD \BOB\DATA]
- Remove the second level directory called FILES which is underneath the BOB directory. [DATA\TWO, RD \BOB\FILES]
- Go down one directory level to the TEXT directory. [BILL, CD TEXT]
- Go to the root directory. [BILL\TEXT, CD \]
- Go to the directory FILES which is underneath the directory OLD. [OLD, CD FILES]

Rearranging Files

The following practice revolves around teaching you how to use the commands needed to rearrange files on the computer. These commands allow you to rename files, delete files, and copy files.

- Give the file AAA.DAT the new name BBB.TXT. [BACK, REN AAA.DAT BBB.TXT]
- Rename the file BILL.1 in the OLD directory to BOB.2. [NEW, REN \OLD\BILL.1 BOB.2]
- Copy the file MYFILE in the FRED directory to the file MY.OLD in the OLD directory. [FRED, COPY MYFILE \OLD\MY.OLD]
- Erase all files with the XXX extension in the OLD directory. [PROG, DEL \OLD\*.XXX]
- Delete all files in the BILL directory. [BILL, DEL *.*]
- Remove all files in the PROG directory that begin with the letter Q. [BILL, DEL \PROG\Q.*]
- Remove the file MYPROG.XXX from the DATA directory. [DATA, DEL MYPROG.XXX]
- Delete all files in the LETTERS directory. [DEL\LETTERS\*.]
- Copy the file DATA.5 from the NEW directory to the OLD directory. [, COPY \NEW\DATA.5 \OLD]
- Copy all the files with a BAT extension from the root directory to the OLD directory. [NEW, COPY \*.BAT \OLD]
- Rename the file MIKE in the BBALL directory to JORDAN. [BILL, REN \BBALL\MIKE JORDAN]
- Copy all the files that start with the letters ME in the BACK directory to files that start with the letters YOU in the NEW directory. [BACK, COPY ME.*.* \NEW\YOU.*.*]
- Change the name of the file JONES (which is in the root directory) to the filename JOHNSON. [BACK, REN \JONES JOHNSON]
- Erase all files in the root directory that have an extension with the number 5 in the third position. [MIKE, DEL \*.??5]
- Copy all the files in the NEW directory to the OLD directory. [NEW, COPY \*.\* \OLD]
- Copy all files in the DATA directory that have a TEXT filename to the root directory. [DATA, COPY TEXT.* \]
Appendix C
Sample Modeling Intervention Script

To demonstrate the operation of the modeling intervention, I have provided a transcript from the modeling videotape. This particular excerpt was drawn from the "Simple DOS Commands" section which presented a model performing and describing the CLS, DATE, and TIME commands. The excerpt provides an introduction to what DOS looks like on the screen and then describes the CLS and DATE commands. Bracketed information in the text is provided to describe the actions being taken by the model during the discussion.

"I'd like to start off by describing some of the simple commands you can use in DOS. Notice that we are on the A drive, that's the A: you see on the screen and there's a backslash telling us what directory we're in. [model points to prompt with pencil] The letters OLD tell us we're in the OLD directory. The greater than sign (that's the prompt) follows and then the flashing cursor telling us that DOS is ready to accept our commands."

"The first command I'd like to tell you about is the command to find out the date on the computer. To find out what the date is you simply type in date [model types DATE on keyboard], D-A-T-E and press enter [model presses enter]. Here you see on the computer that the current date is April 8th, 1980. [model points to date with
After the date you see a prompt to change to a new date. At this point, simply press enter to ignore this [model presses enter] When you do this, the DOS prompt returns on the next line below the date prompt."

"Now suppose we wanted to change the date to May 1st, 1993. To do that we would type the DATE command, a space, and the date you want to change to. This date is the parameter for the DATE command. In this case, we would type D-A-T-E [model types], space May, slash one for the first, slash ninety-three for the year. [model types in date]. Then press enter [model presses enter] and DOS will change its date to May 1st 1993. Notice that you get the DOS prompt back immediately. This tells you that you have not made an error. [model points to DOS prompt with pencil] If you want to check and make sure that the date is correct, all you need do is type DATE and press enter. [model types in date command and presses enter] Again we just press enter at the prompt asking if we want to change the date. [model presses enter] Notice that the new date is correct. [model points to new date with pencil]"
Appendix D

Declarative Knowledge Test

For the following questions, write the correct answer next to the question on the right hand side of the page.

1. What is "A:\>" called?
2. What command gives you the current date on the computer?
3. What is the command to clear the computer screen?
4. What command will tell me how many bytes are in a file?
5. What command allows you to view the contents of a directory?
6. What command tells you how much disk space you have?
7. What command allows you to view the contents of a file?
8. To change the name of a file, what command would one use?
9. What is the command to remove a file from the computer?
10. What command allows one to duplicate a file's contents?
11. What command will over-write an already existent file?
12. If I wanted to go directly to the root directory, what command would I use?
13. What is the command to create a directory?
14. What is the command to erase a directory?
15. What command will always allow you to go up one directory level?
16. If I want to move all files from one directory to another, what two commands must I know?
17. In general, what three commands are needed to prepare and execute the removal of a directory?
Appendix E

Procedural Knowledge Test

DIRECTIONS: For the following questions, be as specific as possible in describing how you would perform the tasks on the computer. NOTE: 1) what you would type, 2) when you would press enter, and 3) what you would look for once the command has been entered.

EXAMPLE: How would you find out what time it is according to the computer?

ANSWER: Type TIME and press (enter).

1. How would you set the computer’s date to April 19th, 1993?

2. How does one erase all the files in the NEW directory that have a BAT extension?

3. Describe the steps you would take to remove the OLD directory (assume that there are files in the OLD directory and that you are in that directory)?

4. How would you know for sure that the files you were to remove in question 2 have actually been removed?

5. Assuming that I am in the FRED directory, how would I change the names of all the files in this directory that start with the letter A to filenames that start with the letter B?

6. Describe the procedure for copying all of the files in the PROG directory to a new directory called FRED (assume FRED currently does not exist as a directory).
7. What is the fastest way to change to the directory BILL if you are currently in the directory BOB (assume that both are under the root directory)?

8. Assume that the directory DATA contains the files: ONE.XY, TRI.XX, FOR.XY, SIX.XX, FIV.ZX, and TOO.XX. What is the fastest way to delete the files that have an XX extension?

9. Describe the procedure you would use to copy all files in the PROGS directory that have DB as the second and third letters of the extension to the root directory.

10. How would you obtain a wide format display of all files with a DAT extension in the directory BILLS which is underneath the directory COWBOYS?

11. Suppose you want to create the directory LETTERS under the directory EDIT which will be under the root directory assuming that neither of these directories exist, how would you execute this command?

12. Suppose you have a large amount of data in a file called TEXT and a 100 page research paper in a file called DATA. Assuming that both files are in the root directory and you are currently in the root directory, what is the fastest way to put the data in the DATA file and the text in the TEXT file?

13. You have all of your letters in a directory called DATA underneath another directory called TYPER. Your task is to place all your letters in a new directory under the
TYPER directory called NEWLETS and remove the DATA directory. How would you do it?
Appendix F

Computer Performance Test

Instructions

a) For the following questions, your job is to execute the commands necessary to successfully perform the task on the computer.

b) You may need to perform intermediate steps to complete the task. For example, if a directory does not exist, you will have to create it.

c) Try to complete the tasks as quickly and accurately as possible using the fewest number of keystrokes possible.

d) DO ALL COMMANDS IN THE ORDER IN WHICH THEY APPEAR BELOW!

1. Go to the root directory. Change the current date to April 19th, 1993 (NOTE: do this from the DOS prompt). Once this is done, clear the screen, and change the date back to today's date.

2. Go to the PROG directory. List all the files in the PROG directory that have a TXT extension. From this list, rename any files that start with the letter O to filenames that start with the letter X.

3. Go to the root directory. Find out what the current time is. Next, clear the screen. Finally, list all the files in the root directory that have no extension.

4. Go to the OLD directory. Delete all the files in the OLD directory and then remove the OLD directory.
5. Go to the BACK directory. List the files in the BACK directory. Using this listing, erase any files with a date that is before December 1st, 1992.

6. Go to the BATCH directory. Display a list of the files in the BATCH directory that have a BAT extension. Using this listing, copy any files that have a size larger than 100 bytes to the root directory.

7. Go to the FRED directory. Display a list of the sub-directories underneath the FRED directory. For each of the directories underneath the FRED directory: a) Go to that sub-directory, and b) delete any files that have a BAB extension.

8. Go to the PROPLUS directory. Display a list of the sub-directories underneath the PROPLUS directory. One of these sub-directories has a text file (that you can read) called INFO.Q6 in it. Find this file, display its contents, and follow the instructions contained in the file.

9. Go to the NEW directory. Display a wide format listing of all the files in the NEW directory. Create two sub-directories called ONE and FOUR underneath the NEW directory. Move (that is, don’t leave an old copy behind) all the files with a 1 extension to the ONE sub-directory and all the files with a 4 extension to the FOUR sub-directory. Finally, rename all the files with a 2 extension to filenames with a 5 extension.
10. Go to the root directory, clear the screen, and notify the experimenter that you are done!
Appendix G

Beneficial Tool Attitude Scale

DIRECTIONS: Please answer the questions below as accurately as possible by circling the appropriate response. Pay close attention to the answer key at the top of each page.

1 2 3 4 5

STRONGLY AM UNCERTAIN STRONGLY
DISAGREE

1. I would like to see all or part of my work done by a computer.

2. Computerizing part of my job/school would make me more competitive in the job market.

3. Using a computer at work/school would (has) significantly increase(d) my satisfaction at work.

4. Using a computer at work/school would (has) significantly increase(d) my productivity.

5. If I use(d) a computer, I can (could) save time and work.

6. Using a computer can (could) be enjoyable.

7. I look forward to computers taking over certain routine tasks of my home and job.

8. Computers are responsible for many of the good things we enjoy.
9. The use of computers is enhancing our standard of living.

10. Life will be easier and faster with computers.

11. Computers will help bring about a better way of life for the average person.
Appendix H

Autonomous Entity Attitude Scale

DIRECTIONS: Please answer the questions below as accurately as possible by circling the appropriate response. Pay close attention to the answer key at the top of each page.

1 2 3 4 5

STRONGLY AM UNCERTAIN STRONGLY

DISAGREE AGREE

1. Computers intimidate and threaten me.
2. Even though computers are valuable and necessary, I still have a fear of them.
3. Computers are being forced on us; we are having our decision process replaced by them, making us lose control of our lives.
4. Soon our lives will be controlled by computers.
5. Computers are dehumanizing to society.
6. Computers are lessening the importance of too many jobs now done by people.
7. Computers intimidate me because they seem so complex.
8. Someday in the future, these machines may be running our lives.
9. Electronic brain machines are kind of strange and frightening.
10. These machines help to create unemployment.

11. I know that I will never understand how to use computers.
## Appendix I

**Factor Loadings for Knowledge Items**

<table>
<thead>
<tr>
<th>Item Number</th>
<th>Pre-test DK</th>
<th>Post-test DK</th>
<th>Pre-test PK</th>
<th>Post-test PK</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.48</td>
<td>.56</td>
<td>.32</td>
<td>.14</td>
</tr>
<tr>
<td>2</td>
<td>.49</td>
<td>.41</td>
<td>.87</td>
<td>.70</td>
</tr>
<tr>
<td>3</td>
<td>.67</td>
<td>.54</td>
<td>.89</td>
<td>.62</td>
</tr>
<tr>
<td>4</td>
<td>.63</td>
<td>.65</td>
<td>.61</td>
<td>.39</td>
</tr>
<tr>
<td>5</td>
<td>.72</td>
<td>.44</td>
<td>.81</td>
<td>.73</td>
</tr>
<tr>
<td>6</td>
<td>.70</td>
<td>.62</td>
<td>.92</td>
<td>.79</td>
</tr>
<tr>
<td>7</td>
<td>.78</td>
<td>.67</td>
<td>.77</td>
<td>.67</td>
</tr>
<tr>
<td>8</td>
<td>.67</td>
<td>.54</td>
<td>.72</td>
<td>.71</td>
</tr>
<tr>
<td>9</td>
<td>.55</td>
<td>.47</td>
<td>.72</td>
<td>.77</td>
</tr>
<tr>
<td>10</td>
<td>.58</td>
<td>.60</td>
<td>.76</td>
<td>.76</td>
</tr>
<tr>
<td>11</td>
<td>.27</td>
<td>.38</td>
<td>.92</td>
<td>.75</td>
</tr>
<tr>
<td>12</td>
<td>.75</td>
<td>.58</td>
<td>.43</td>
<td>.49</td>
</tr>
<tr>
<td>13</td>
<td>.83</td>
<td>.58</td>
<td>.78</td>
<td>.57</td>
</tr>
<tr>
<td>14</td>
<td>.84</td>
<td>.45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>.51</td>
<td>.66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>.29</td>
<td>.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>.80</td>
<td>.48</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Vita

Drew Brock was an honor graduate of Paris High School in Paris, Texas. Upon his graduation in 1985, Drew attended the University of Texas at Austin. Studying psychology, he completed his Bachelor of Arts degree in 1988. His senior thesis examined a theoretical integration of the constructs of locus of control, perceived control, and the Type A behavior pattern. In the fall of 1989, Drew entered the Industrial/Organizational Psychology graduate program at Louisiana State University. He received his Master of Arts in Arts and Sciences from the University in 1991. Drew's research examined the relationships between computer use, computer attitudes, and work attitudes in a field study. The focus of his research during his graduate career has been on human-computer interaction and attitudes toward computers. He has presented two posters at conventions on this topic and has written a paper (in press) examining the construct validity of computer attitudes and the relationships of those attitudes to computer use. Currently, Drew works as a Human Resource Intern for a large hospital in Baton Rouge.
Candidate:  Drew Benjamin Brock

Major Field:  Psychology

Title of Dissertation:  Computer Training: The Role of Computer Attitudes and Behavior Modeling in the Acquisition of Declarative and Procedural Knowledge

Approved:

Major Professor and Chairman

Dean of the Graduate School

EXAMINING COMMITTEE:

Dwight Lane

Richard Stein

Robert Matthews

Lynne Handzel

Date of Examination:

September 8, 1993