Chapter 5

RESEARCH DESIGN

To test the proposed modifications to the original C&H model of computer training, an experiment was conducted. The experiment involved training subjects on the use of an Excel 5.0 feature, Solver, that is used for solving linear programming problems. A between-subjects design was employed with groups of subjects receiving one of three types of training: behavior modeling, lecture-based training, or Computer-Aided Instruction. Task complexity was manipulated at two levels (low and high) by varying the number of constraints in a linear programming problem given to subjects to assess final performance.

Self-efficacy and outcome expectations were measured using instruments developed by C&H. All questionnaires and task performance measurements were administered on computers. These measurements are presented in greater detail in the sections that follow.

The apparatus used in this study to collect responses to questionnaires and performance measures was an Excel spreadsheet that was controlled by a Visual Basic macro. All questionnaires (except an Informed Consent document) were administered on the computer via dialog boxes.

5.1 SUBJECTS

Inherent in any laboratory study is a bias against using students as subjects when addressing the issue of external validity. “Critics point out that the student samples so frequently used in such studies are not representative of the population to which findings are presumably to be generalized” (Griffin and Kacmar, 1991). Berkowitz and Donnerstein (1982), however, argue that such samples and experiments are acceptable for testing causal relationships and theories as long as no attempts are made to infer estimates about population parameters. Indeed, cognitive studies abound with student samples, whereby processes are presumed to remain constant across most groups of adults. Given these contentions, justification for the sample used in this study is provided.

An experiment was conducted during the Spring of 1996 with 353 students at Virginia Tech. This experiment was conducted in two computer labs on Virginia Tech’s campus and administered to approximately 20-24 students at a time. It was performed during the third week of March 1996, with each session lasting approximately one hour. Participants were required to attend one session during their regularly scheduled lab hours on Monday through Thursday of that week. Groups of participants were randomly assigned to one of three training sessions (i.e. behavior modeling, lecture-based training, and Computer Aided Instruction). Within each group, subjects were randomly administered one of two types of performance tests (i.e. low complexity or high complexity).

Subjects were voluntarily recruited from sections of Accounting 1504, (Introduction to Business Information Systems). Because this course used Microsoft Excel to provide an introduction to spreadsheet concepts, all of the subjects possessed a basic knowledge of Excel. This existing knowledge provided a sound basis for assessing prior performance.

Table 5.1 Sample Composition

Number of Subjects		
	Original Data	After Eliminations
Behavior Modeling - High Complexity	69	59
Behavior Modeling - Low Complexity	64	59
Lecture Based - High Complexity	78	62
Lecture Based - Low Complexity	87	65
CAI - High Complexity	27	22
CAI - Low Complexity	28	24
Total	353	291

Because the manipulation that was used involved a specific Microsoft Excel feature, Solver, subjects were initially screened to determine their existing knowledge of this feature. Only those subjects with no prior exposure to Solver or a similar feature in a different spreadsheet program were used in the analysis. Table 5.1 illustrates the composition of the sample used for analysis.

5.2 PROCEDURE

As illustrated in Figure 5.1, at the beginning of each experimental session, subjects were asked to complete a general demographic questionnaire. Subjects then were given a prior performance test on a computer in order to measure their existing knowledge of Excel 5.0. Each subject had sole access to a computer in the training room throughout the experiment. After the initial performance test, subjects were asked to view an 18 minute video involving either behavior modeling or lecture based instruction on the use of Solver. For these groups, a television set was available in the training room. In the CAI groups, rather than viewing a video, subjects were given 18 minutes to complete a computer tutorial on the use of Solver. After viewing the videotape (or completing the tutorial), participants completed two questionnaires regarding their outcome expectations for using a computer and about their own abilities to use a computer (i.e. computer self-efficacy). Finally, the subjects performed a task using the information they learned in the training session. They were given 15 minutes to complete their assigned task. At the completion of each task, subjects completed a questionnaire about the perceived difficulty of the task.

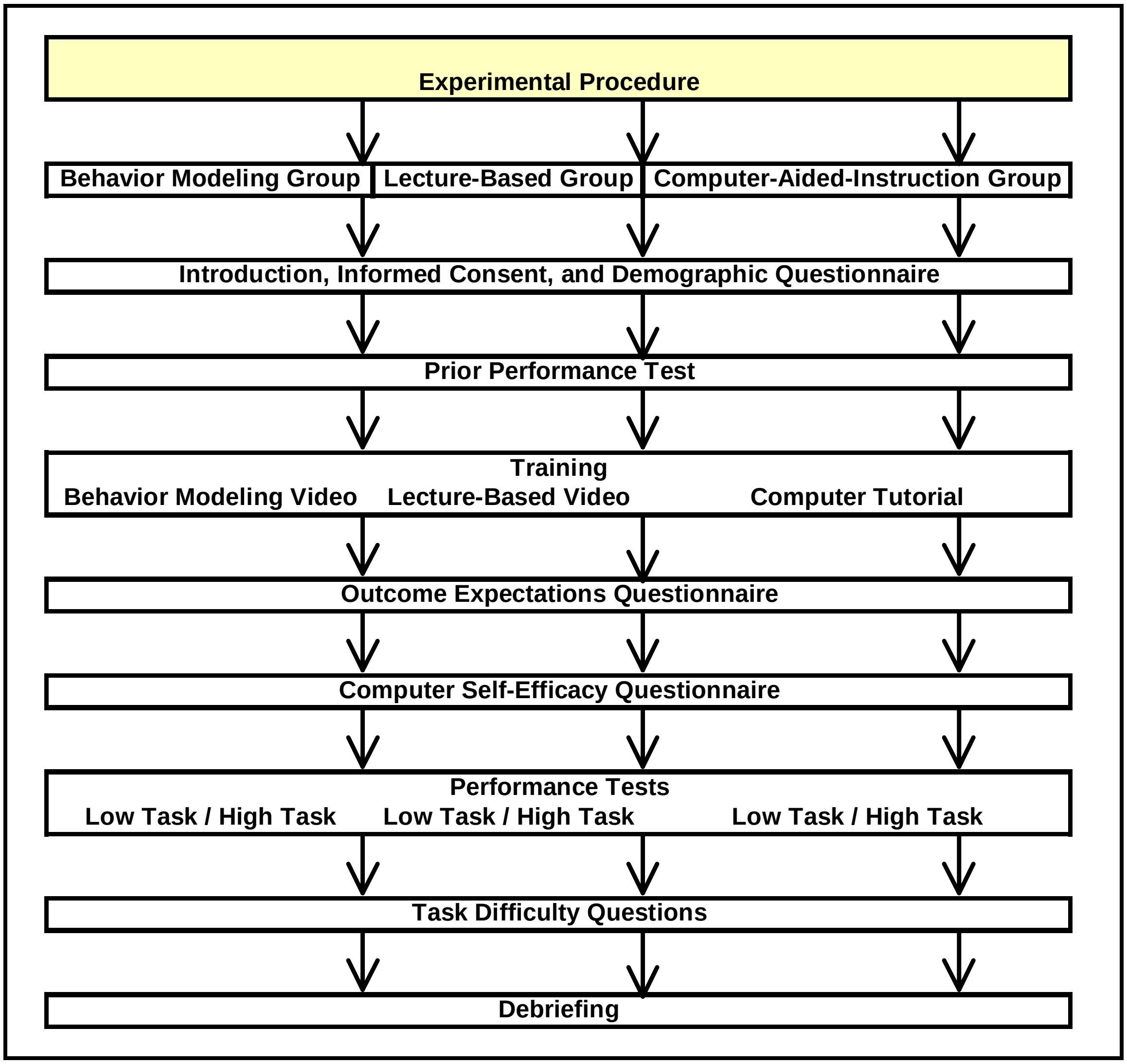


Figure 5.1 Experimental Procedure

The duration of each experimental session was one hour.

In this study, training methods were designed to convey the same amount and type of information to facilitate comparisons among the methods. For example, no instructor (or computer) feedback was provided in any training session in this study. The purpose of this restriction was to eliminate any confounds caused by individual instructor characteristics and potential instructor/subject interactions. Also, to isolate the effects of the training methods, no exercises were provided. Because hands-on experience often is used to promote learning, exercises were excluded from the training sessions to isolate the unique effects of the training methods on performance. Finally, all training methods required the same amount of time. The behavior modeling and lecture-based training sessions were videotaped and lasted eighteen minutes. The CAI sessions were designed as on-line tutorials in which users read through the materials at their own pace. For these sessions, however, the same amount of time was allotted as used in the videotaped sessions (i.e. eighteen minutes). To further equate the training methods, identical visual aids (i.e. computer screens, examples, etc.) were used in all of the training sessions. For examples of the information that was presented in the sessions, see Appendix A.

5.3 MEASUREMENTS AND MANIPULATIONS

In causality models, two types of variables exist that provide a basis for depicting and analyzing relationships among constructs: exogenous and endogenous variables. Pedhauzer and Smelkin (1991) provide explanations of each of these types of variables:

> An exogenous variable is a variable whose variability is assumed to be determined by causes outside the model under consideration. Stated differently, no attempt is made to explain the variability of an exogenous variable or its relations with other exogenous variables. In contrast, an endogenous variable is one whose variation is to be explained by exogenous and other endogenous variables in the causal model. (p. 311)

Given these definitions, below are operational definitions of various exogenous and endogenous constructs used in the model being tested in this dissertation.

5.3.1 Exogenous Variable--Prior Performance

Basic spreadsheet abilities served as proxies for prior performance. They were determined by administering a test on general Excel commands (e.g. entering data, constructing simple formulas, copying data and formulas, etc.) and were performed on a computer. The sum of scores for all ten Excel tasks served as a manifest indicator of prior performance.

Subjects were given eight minutes to complete the ten Excel tasks. Tasks were ordered from least difficult to most difficult, and all responses were independent of answers to previous questions. Each item on the test was given a score ranging from 0 to 1. A score of 0 was given for questions that were completely incorrect, and a score of 1 was given for questions that were completely correct. Scores between 0 and 1 represented partial credit for each question. Awarding partial credit was appropriate because each question required the execution of various subtasks. For example, question #8 instructed users to center a label across a set of columns. This question required two types of skills: (1) the knowledge of *how* to center a label across columns and (2) the motor skills to physically highlight the columns across which the data was to be centered. In cases where a user highlighted too many or too few columns, partial was granted. Clearly, the user possessed the knowledge to perform the task, but failed to perform the manual operation correctly. Similarly, in questions #5 and #9, where the user was required to enter a formula in a cell, partial credit was given if (i) the formula was entered correctly but in the incorrect cell, (ii) the formula was not entered correctly in terms of Excel syntax (e.g. omitted equals sign indicating the presence of a formula), or (iii) the formula contained errors in logic (e.g. misplaced/omitted parentheses) but was otherwise entered correctly. The list of Excel tasks is provided in Appendix B.

In addition to the Excel tasks, two other measures were used to assess prior performance: a self-reported measure of Excel expertise and a self-reported cumulative grade point average. The measure of Excel expertise ranged from 1 to 3, with 1 representing a novice user and 3 indicating an experienced user. The grade point average was reported on a continuous 4.0 scale. Both of these questions were presented in the general demographic questionnaire at the beginning of the experimental session.

5.3.2 Exogenous Variable --Training Type

The treatment manipulated in this study was the type of training given to subjects. Behavior modeling was compared to traditional lecture-based training and computer-aided instruction. For behavior modeling, an eighteen minute video displaying a model performing the steps required to execute both simple and complex tasks using Excel's Solver was presented to the subjects. As in the C&H study, the model periodically made and recovered from mistakes. The act of displaying imperfect performance in behavior modeling was necessitated by a "perceived similarity between the observer and the model [to] facilitate imitation" and enhance vicarious learning of coping skills. (Meichenbaum , 1971) For trainees to learn techniques for error recovery, the model had to demonstrate possible mistakes in order to show the procedures for correcting them.

The second type of training was a lecture-based approach. Unlike the C&H study, the instructor providing the training was videotaped rather than "live". The purpose of videotaping both behavior modeling and lecture training sessions was to eliminate any potential confounds caused by the interaction between the instructor and subjects. The video lasted for eighteen minutes and displayed the instructor describing the procedures needed to execute both simple and complex tasks using Excel's Solver. A picture of the screen was displayed periodically in order to illustrate the commands and data involved in performing the various tasks. No demonstration of executing these commands, however, was given. Error recovery was taught by verbally communicating possible errors and techniques for recovering from them. For example, in one part of the video, the instructor said, "If you accidentally misspecify a constraint, you can delete it and begin again by highlighting the constraint and clicking the delete key on the Solver dialog box." The same individual was featured in both the behavior modeling tape and in the lecture-based training tape to eliminate any potential effects caused by different instructors.

The third type of training, Computer Aided Instruction, was an on-line tutorial that allowed subjects to read through the training materials at their own pace. The amount and type of information to be presented in the CAI training emulated that of the other training methods. The tutorial were written in Excel 5.0 and used the same computer screens and examples that were used in the other training methods. The primary difference between this approach and the videotaped sessions was the absence of an instructor. To avoid any confounds related to hands-on experience, no exercises were included in the CAI training. Appendix A contains excerpts from the scripts that were used in the videotaped sessions and the screens that were displayed for all three training methods.

5.3.3 Endogenous Variable--Outcome Expectations

A six-item questionnaire was used to assess performance outcome expectations. This instrument was a modified version of an eleven-item questionnaire that was developed and used by Compeau and Higgins in 1991 and 1995, respectively. In their studies, Compeau and Higgins factor analyzed the instrument and found that only six of the eleven items loaded on performance outcome expectations. The other five items represented personal outcome expectations.   In this study, only those items tapping performance outcome expectations were used. Responses were scored on a 5-point Likert type scale. Each item on the instrument represented a separate manifest indicator of outcome expectations. Factor analysis using a varimax rotation was performed to ensure adequate construct validity, and Cronbach's alpha was computed to test the instrument's reliability. These measures are presented in Chapter 6. Appendix-B illustrates the questions that were included on the outcome expectations instrument.

In this study, the measurement of outcome expectations followed the training but preceded the measurement of the dependent variable, performance. In causal modeling, a necessary condition for causality is that the causal variable *must* precede the variable it is hypothesized to affect. (Pedhazur and Smelkin, 1991, p. 699; Keith, 1988) Therefore, the measurement of outcome expectations was taken after the manipulation of the treatment. Additionally, to avoid any effects that performance might have had on outcome expectations, this questionnaire was administered before the final task.

5.3.4 Endogenous Variable -- Computer Self-efficacy

A ten-item questionnaire, developed by Compeau and Higgins (1995b), was administered to determine subjects' perceptions about their abilities to use computers. Again, like outcome expectations, this instrument was administered after the performance measure but before the final task. Each item on this questionnaire represented a separate manifest indicator of computer self-efficacy. A factor analysis using a varimax rotation was performed to ensure the construct validity of this instrument. Cronbach’s alpha also was computed to assess the reliability of this questionnaire. These results are presented in Chapter 6. Appendix-B illustrates the questions on this questionnaire.

5.3.5 Endogenous Variable -- Task Performance

Subjects were given one of two tasks as the final performance measure in order to assess their understanding of the information presented in the training sessions. Both types of tasks required the use of Excel's Solver feature. The tasks differed, however, in the number of constraints that were included in the problem. The task representing high complexity, involved five constraints. The other task, representing low complexity, had no constraints. Subjects were allowed fifteen minutes to complete their assigned task.

The computer program in this experiment recorded all keystrokes from the moment the user opened the first task screen. When the user completed his task, a dialog box appeared that asked the user to indicate, on a scale of one to ten, his perceived difficulty of the task. This task difficulty questions was intended as a check on the strength of the complexity manipulation.

Scoring of the final performance measure was based on the number of possible actions (subtasks) included in each task. Because the tasks involved a different number of constraints, direct comparison of the results was not possible. Instead, ratios of the number of correct actions taken to the total possible number of actions required were used as the final performance measurements. The high complexity task required eighteen distinct actions, whereas the low complexity task needed only four actions. From a grading perspective, the low complexity task score gave credit to subjects for not only performing the required subtasks, but also knowing when *not* to include any constraints. For example, because the low complexity task did not require specification of any constraints, one of the four possible points represented the absence of any constraints in the solution. For subjects who performed additional, unnecessary tasks, points were deducted from the performance score. For the high complexity task, subjects were not prone to add more constraints than were necessary; therefore, no additional action was built into the score. The actual performance tasks are presented in Appendix-B.

5.3.6 Moderating Variable -- Task Complexity

Wood (1986) proposes a theoretical model of task complexity in which three types of complexity are described: component, coordinative, and dynamic. In this dissertation, only component complexity is relevant to the type of task that was presented. Component complexity is a "direct function of the number of distinct acts that need to be executed in the performance of [a] task and the number of distinct information cues that must be processed in the performance of those acts." (Wood, 1986)

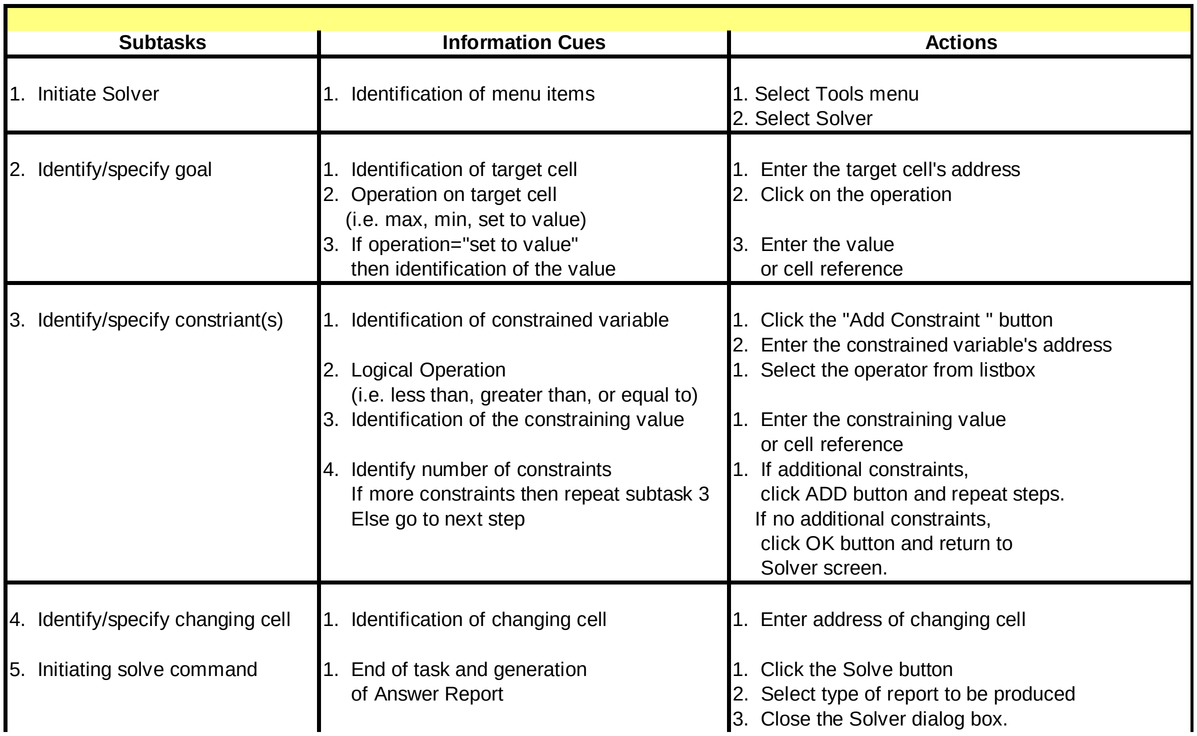
A formula for computing component task complexity that incorporates actions and information cues is as follows:

$$TC_1 = \sum_{j=0}^{j=p} \sum_{I=1}^{i=n} W_{ij}$$

"where *n* = number of distinct acts in subtask *j*, W_{ij} = number of information cues to be processed in the performance of the i^{th} act of the j^{th} subtask, *p* = number of subtasks in the task, and TC_1 component complexity." (Wood, 1986)

The purpose of computing component complexity for the two tasks in this study is to provide an objective means of quantifying the strength of the manipulations. Further, the results of future studies that attempt to replicate or extend this experiment using different tasks can be compared to the results obtained in this study.

To compute the complexity of the tasks presented in this study, the distinct subtasks and information cues were first determined. As illustrated in Figure 5.2, using Solver required five distinct subtasks: (1) initiating Solver, (2) identifying and specifying a *goal*, (3) identifying and specifying a *constraint(s*), (4) specifying the *changing cell*, and (5) initiating the command to solve the problem.



Subtasks	Information Cues	Actions
1. Initiate Solver	1. Identification of menu items	1. Select Tools menu 2. Select Solver
2. Identify/specify goal	1. Identification of target cell 2. Operation on target cell (i.e. max, min, set to value) 3. If operation="set to value" then identification of the value	1. Enter the target cell's address 2. Click on the operation 3. Enter the value or cell reference
3. Identify/specify constriant(s)	1. Identification of constrained variable 2. Logical Operation (i.e. less than, greater than, or equal to) 3. Identification of the constraining value 4. Identify number of constraints If more constraints then repeat subtask 3 Else go to next step	1. Click the "Add Constraint " button 2. Enter the constrained variable's address 1. Select the operator from listbox 1. Enter the constraining value or cell reference 1. If additional constraints, click ADD button and repeat steps. If no additional constraints, click OK button and return to Solver screen.
4. Identify/specify changing cell	1. Identification of changing cell	1. Enter address of changing cell
5. Initiating solve command	1. End of task and generation of Answer Report	1. Click the Solve button 2. Select type of report to be produced 3. Close the Solver dialog box.

Figure 5.2 Breakdown of Component Complexity for Solver

Initiating Solver involved simple actions. Only one information cue and two distinct actions were needed to initiate Solver: Identification of menu items, Selection of the Tools Menu from the Menu Bar, and Selection of Solver from the Tools menu, respectively.

Identifying and specifying a *goal* involved both subtasks and information cues. A minimum of two information cues were needed in the task description to identify a goal: (1) the variable to be maximized, minimized, or set equal to a specified value, (also known as the “target cell”), and (2) whether the variable was to be maximized, minimized , or set equal to a specific value.   If the variable was to be set equal to a specific value, then an additional information cue, the specified value, was needed. An example of this is a break-even analysis in which the goal is to set profit equal to zero.

To implement a goal in Solver, a minimum of two distinct actions had to be performed that correspond to the information cues: (1) selecting the cell that contained the variable and (2) indicating whether the variable was to be maximized, minimized, or set equal to a specific value. If the variable was to be set equal to a value, then this value or cell reference had to be entered as a third distinct action.

Identifying and specifying *constraints* was the most complex item of Solver's four subtasks. A constraint involved four information cues: (1) the variable that was constrained, (2) the type of constraint (i.e. less than, greater than, equal to, etc.), (3) the constraining value, and (4) the number of constraints in the problem. An example of this is a decision regarding how much inventory to purchase.   A constraint in this example might be that floor space is limited to 6000 square feet.

The distinct actions involved in specifying a single constraint included (1) clicking the Add Constraint button, (2) identifying the cell that contained the constrained variable, (3) specifying the relationship, (4) entering the constraining value, and (5) clicking the Add or OK button.

For each additional constraint, only distinct information cues were added. Because the **actions** to be performed were the same for every constraint, they were not distinct and, therefore, contributed nothing to the formula. Thus, for every new constraint that was added, complexity was increased by only the 4 points related to **information cues**.

Identifying and specifying the changing cell involved knowledge of the formulas that were used in the task. For example, a hypothetical problem may require a person to determine the number of units to be sold in order to achieve a total profit of $100, given that each unit produces a profit of $20. Cell A1 contains the profit per unit, and cell A2 contains the formula that multiplies profit per unit by quantity sold. Cell A3 represents the cell that contains the actual quantity when the solution is reached. Thus, it is this cell, (A3), that is used in the formula to determine total profit. This cell is known as the *changing cell*.

In this study, identifying the changing cell required one information cue: determining the location of the cell that contained the solution. Specifying the changing cell required one distinct action: entering the address of the changing cell.

Finally, initiating the command to solve the problem required only one information cue: determining when the problem was completely specified and, thus, ready for processing. The action required to perform this subtask required clicking the OK button on the Solver screen.

As an additional feature, Solver produces reports that are designed to assist decision makers in the interpretation of their results. Three types of reports are available: (1) Answer, (2) Sensitivity, and (3) Limits. In this study, subjects were instructed only on the use of the Answer report. The generation of this report, however, was not graded. Because Excel does not produce this report if a solution cannot be found, only those subjects who correctly solved the task were able to generate this report. Thus, it was not possible to determine whether subjects did not elect to produce this report due to lack of understanding or compliance with the instructions or whether they were unable to produce it due to an infeasible solution. In terms of component complexity, the generation of this report was considered in the computation of overall task complexity. In the final performance measure, however, it was not be included as part of the total possible points awarded for a given task.

As stated previously, two tasks were presented in this study. The low complexity task involved no constraints and had a goal of setting a variable equal to a specific value. This task had a component complexity score of 24. The high complexity task, in contrast, involved five constraints, and its goal was to maximize a given variable. The component complexity score for this task was 40.

5.4 MANIPULATION CHECKS

Manipulation checks also were performed during the experiment. After completing his/ her assigned task, each subject was asked to indicate, on a scale from one to ten, the perceived difficulty of the task. This final task difficulty question served as a check on the strength of the complexity manipulation. A One-Way Analysis of Variance was computed for the total number of subjects who participated in the training sessions. These results are presented in Chapter 6.