BEYOND WHAT-IF: ENHANCING MODEL ANALYSIS IN A DECISION SUPPORT SYSTEM

By

DAVID MICHAEL STEIGER

Bachelor of Science
The University of Texas
Austin, Texas
1972

Master of Business Administration
The University of Texas
Austin, Texas
1976

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 1993

OKLAHOMA STATE UNIVERSITY

BEYOND WHAT-IF: ENHANCING MODEL

ANALYSIS IN A DECISION

SUPPORT SYSTEM

Thesis Approved:

Ramesh Sharda

Thesis Adviser

Thomas C. Collins

Dean of the Graduate College

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to Dr. Ramesh Sharda for his encouragement and advice throughout my graduate program, and especially through the dissertation phase. I also thank Dr. Nikunj Dalal, Dr. Rick Wilson and Dr. Woody Hedrick for serving on my dissertation committee. Their suggestions and support were helpful throughout the study.

I would like to express my deepest thanks to Natalie Steiger, my bride of fourteen years. To her I now pass the graduate school baton, and hope that I can provide some small portion of the support during her Ph.D. program that she has provided during mine.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Once a model is built, validated and run for an initial set of assumptions and instantiating values the decision maker's job has just begun. That is, practically no decision is made on a single model run. A manager develops his understanding of a problem and its solution as he works on it; i.e., as he iteratively analyzes and understands the interrelationships between the model variables/parameters and the model solution [Little, 1970]. There follows an extensive set of what-if questions which explore the workings and tradeoffs of the business system represented by the model. These include such questions as:

1) How does the model behave under small changes in its assumptions? Are certain model solutions and corresponding decisions particularly robust? Which what-if instances should be run next to test the underlying model assumptions or determine alternative solutions which might be better than those already under consideration? What paradigm-specific model settings might be changed (and in what direction) to provide a better model solution?

2) Why did a specific variable increase from one model run to another or from one model year to another? Why did modeled expenses remain flat in 1989 instead of increasing?

3) What are the key variables which can most affect the model solution and corresponding decision? Are several decision variables, parameters and/or intermediate variables interrelated and does some combination of changes in these variables adversely affect my decision? How are these variables interrelated; i.e., is there a simple, deterministic relation between them and the model solution?

4) What is the best solution, given a set of assumed or specified evaluation criteria?

5) Which model solution and corresponding decision could be implemented with the least offensive set of changes (to the current modus operandi or operational status quo) and still realize some large percentage (say 80%) of the benefits of the heretofore 'best' solution? Can we rank order the best ten solutions in terms of difficulty to implement or required changes from the status quo? Which change(s) to the status quo should be implemented first, before other changes, to realize the most benefits early in the implementation process?

6) When this model was used two years ago to help make a decision, did actual results closely correlate to those predicted by the model? Did the actual benefits materialize as forecasted by the model? Has the environment changed significantly since then to require an update or rework of the model?

These questions are grouped into six types of analysis which should be enhanced by the model analysis environment.

Some of the questions in group 1 are standard sensitivity analysis questions which are addressed in most DSS, and particularly well in linear programming-based models. Such sensitivity analysis explores effects of changes in the data on the model solution to determine how robust the model's implications are [Hillier and Lieberman, 1988; Geoffrion, 1975]. Other group 1 questions involve what we term 'planning' issues; i.e., helping the decision maker plan which what-if instance to run next. For example, if several variables interact to influence profits, and fifteen model instances have been run to test the effects of these interactions, which what-if instance should be run next to find a high level of profits? Some researchers refer to such questions as 'guidance' questions; i.e., those which "provide the user with clues as to interesting or important changes to the model structure or parameters" [Brennan and Elam, 1986].

Questions in group 2 are what we term 'comparison' questions and consist of explaining reasons for surprising or unexpected model behavior. Some researchers refer to these as 'explanation' type questions [Kosy and Wise, 1986; Brennan and Elam, 1986; Elam and Konsynski, 1987]. Mathes [1969] suggests that managerial decision makers analyze problems on the basis of differences between model output and similar historical experiences. Little [1970] suggests that a manager often compares model output with his intuition; substantial differences in the two initiate an iterative process of determining the causes of the differences and many times lead the manager to learning something new about the interaction of a number of factors, and updating his intuition.

Questions in group 3 are termed 'causation' questions and consist of identifying and quantifying causal relationships between model variables, intermediate variables and parameters and the model solution. Other researchers refer to these as 'exploration' questions [Brennan and Elam, 1986].

Questions in group 4 are termed 'recommendation' questions and are well documented in the DSS literature.

Questions in group 5 are termed 'implementation' questions and consist of comparing the current modus operandi with the recommended solution to suggest a decision with a minimal, or least offensive, set of changes which would result in a substantial fraction of the total improvement promised by the recommended solution. Also included in this group are the questions which address the order in which the changes are implemented. This corresponds to the 'priority analysis' of Geoffrion and Graves [1974].

Questions in group 6 are termed 'post audit' questions and consist of determining how well the model predicted the actual outcome of the decision and, where discrepancies exist, where the model and reality differed and why. Post audit actually forms the basis for model validation of future models. Such questions should be included as part of the overall DSS analysis [Sprague and Carlson, 1982] and are suggested as research issues (as budget variance analysis questions) by Kosy and Wise [1986].

This research is aimed at answering some of these questions. We propose an extension of the model management system in a decision support systems which uses insight-generating technologies to analyze the multiple model instances created by the decision maker. That is, the purpose of this dissertation is fourfold:

1) to define and justify the need for model insight generator systems (MIGS),

2) to propose an architecture for model insight generator systems and discuss a set of software tools which could be used in such a system,

3) to provide a mathematical statement of the basic analysis functions of a model, and

4) to implement one subsystem of the MIGS architecture, called INSIGHT, which identifies the key factors and their deterministic relations, as part of an integrated decision support system.

We shall focus on the uses of models and, more specifically the identification of key factors and their relations, in the context of spreadsheet modeling. We focus on spreadsheets primarily because of their broad applicability, acceptability and use in industry for a wide variety of problems. However, the concepts developed herein are equally applicable to a wide variety of other modeling paradigms such as linear programming, nonlinear programming, simulation, statistical regression, forecasting, and neural networks.

This dissertation is organized into seven chapters as follows. Chapter 2 describes the basic definition, purpose, benefits and requirements of model analysis and reviews the relevant analysis technologies. This section provides a justification for our work. Chapter 3 provides a mathematical statement of model analysis requirements, including the seven insight generating analysis tasks. Chapter 4 describes our proposed system architecture for the Model Insight Generator Systems (MIGS), a component of model management systems which help the user generate insights into his decision making environment by providing high tech tools which address the primary model analysis tasks developed in Chapter 3. Chapter 5 describes our implementation of the prototype of one MIGS component, named INSIGHT, which addresses the identification of key model variables and the key relationships between these key variables. Also included in this chapter is a sample session for using INSIGHT. Chapter 6 provides the future directions and research areas for model insight generator systems. Finally, Chapter 7 provides the summary and conclusions of our research.

CHAPTER II

MODEL ANALYSIS: STATE-OF-THE-ART AND RELEVANT STUDIES

Introduction

The following literature review of the state-of-the-art in model analysis is divided into five sections. The first section presents our underlying assumption that the real purpose of modeling is to aid the decision maker in his development of 'insight' into his problem environment. This is followed by a section which defines and discusses insight from a theory of learning viewpoint, and draws from that discussion the primary characteristics of an insightful model. The third section reviews the current analysis tools in three popular modeling paradigms and how they are restricted in insight generating capabilities. The fourth section then introduces several technologies which appear to have potential for enhancing insight generation in models. Finally, the fifth section draws several conclusions from this literature review.

Purpose of Model Analysis

Mathematical models, in the context of business decision making, are idealized representations of a business problem expressed as a system of equations and related mathematical expressions that describe the essence of the problem [Hillier and Lieberman, 1986, p. 19]. That is, a mathematical model explicitly states the mathematical relationships between decision variables, intermediate variables and outcomes (or attributes) [Bodily, 1985].

Geoffrion states that "the true purpose . . . (of modeling) . . . is to develop insights into system behavior which in turn can be used to guide the development of effective plans and decisions. Such insights are seldom evident from the output of a single (model) run. One must know not only what the optimal solution is for a given set of input data, but also why" [Geoffrion, 1976a]. That is, the true purpose of modeling, both during model building and during output analysis, is the process of understanding the system being modeled, which controllable variables are important, and how these controllable variables interact with each other to impact the performance of the system.

To this, Jones adds that "developing insight into model behavior is ultimately a process of discovery, of finding trends, surprising behaviors, and comparing the behavior of the model to what is expected or observed in the real system. How well does it match? Where does it differ? Do those changes correspond to what is expected? If yes, why? If not, why not? In one sense, developing insight involves recognizing patterns: How does the model respond to changes in parameters? What trends can be detected? How do the trends compare" [Jones, 1988]?

The common element in each of these statements is the development of insight; i.e., perceiving the inner nature of the tradeoffs inherent in any complex business situation. Yet, understanding the system is not always easily accomplished, especially when several, if not many, components of the situation interact and cause a combinatorial explosion of what-if possibilities, each of which by itself is difficult to analyze.

If insight is so highly important in modeling, perhaps we should review the concept of insight from its roots in the theory of learning.

Insight: From the Theory of Learning Viewpoint

The study of insight was popularized by the Gestalt psychologists in the early 1900's and forms one of the bases of their theory of learning [Burton and Burton, 1978]. Thus, we turn to this literature to identify the meaning and characteristics of insight.

Definitions of Insight. Insight is defined by Webster as "a clear understanding of the inner nature of some specific thing" [Webster's Dictionary, 1966]. In defining insight, Hilgard included problem solving in his definition, stating that insight is "the understanding of the essential relationships of the problem" [Hilgard, 1956]. Logan and Ferraro [1978] included some indications of the sources of insight in their definition: "insight is the sudden appearance of adaptive behavior that frequently involves the combination of previously learned responses into novel combinations". And Kohler [1969] linked insight with perception, defining insight as "a set of processes supplying the most appropriate final link in a situation presented as incomplete"; that is, insight is actually a quality of perception [Lee, 1965]. In fact, Webster's dictionary actually uses the two terms, perception and insight, synonymously.

Examples of Insight. Kohler, a German psychologist, formed the foundation for experimental research on insight in his 1918 book which was later translated into English [Kohler, 1925]. While stranded on the Canary Islands by the outbreak of the 1914-1918 War, Kohler studied the behavior of chimpanzees in solving problems which they had never met before. He reported two classic experiments. In the first, Kohler placed his smartest chimp, Sultan, in a large barred cage and gave him two short bamboo sticks which could be joined by inserting the top of one stick into the hollow end of the other. Outside the cage, he placed a banana just out of reach of either of the sticks. After repeatedly trying without success to reach the banana with each individual stick (as he had done on numerous occasions previously) Sultan abandoned his efforts, sat down and started playing with the two sticks. While doing this, he happened to insert one stick into the hollow end of the other and found himself holding one long stick. Immediately, he jumped up, ran to the bars and drew the banana towards him with the double stick.

In the second classic experiment, a banana was suspended from the top of a chimpanzee cage just beyond reach of six chimps; a box was placed nearby which could be used as a platform to reach the banana if dragged under it. All six chimps tried vainly to reach the fruit by leaping up from the ground. Sultan, the smartest chimp, soon gave up these attempts and paced restlessly up and down. He suddenly stood still in front of the box, seized it, dragged it hastily under the banana, and using the box as a platform jumped up and grabbed the banana. About five minutes had elapsed from the fastening of the banana to the ceiling. However, from the momentary pause in front of the box to the grasping of the banana only a few seconds elapsed, a perfectly continuous action after the first hesitation. The pause in front of the box was important because it could have been the moment when Sultan discovered the relationship between the box and the banana. In analogous circumstances, we might imagine a human being thinking, "Aha, now I see!"

That Sultan quickly proceeded to grab the fruit strengthens the impression that the pause was a period of discovery. Further, Kohler found that animals that attained a solution through such insight characteristically repeated the solution quickly on subsequent presentations of the problem. This was in marked contrast to the "stupidity of stimulus-response learning which required many trials to reach perfect performance" [Kohler, 1925].

<u>Characteristics of Insight</u> Both of these experiments demonstrate certain similarities or characteristics of insight. Lonergan [1958] listed five such characteristics: i.e., insight 1) comes as a release to the tension of inquiry, 2) comes suddenly and unexpectedly, 3) is a function of not only outer circumstances, but also of inner conditions, 4) pivots between the concrete and the abstract, and 5) passes into the habitual texture of one's mind.

Hilgard [1956 p. 238-239] lists three defining criteria which clearly differentiate insight from other problem solutions: 1) the interruption of movement for a period for survey, inspection and attention, followed by the critical solution, 2) the ready repetition of the solution after a single critical solution, and 3) the generalization of one insightful solution to new situations that require mediation by common principles, or awareness of common relationships.

Even more interesting (from our viewpoint) is a list of five characteristics of perception (i.e., insight) as provided by Lee [1965, pp. 48-49]. Recall that Kohler defines insight as a quality of perception and Webster lists insight and perception as synonyms. Lee's characteristics of perception include the following. 1) The perceiver himself organizes his field into 'figure' and 'ground'. The figure then commands the whole attention of the perceiver, and the ground receives little or no further consideration. 2) The perceiver tends to look for simplicity, regularity and completeness. Simpler perceptions take precedence over more complex ones. 3) There is a tendency to perceive incomplete material as complete. 4) It follows that the perceiver organizes his own pattern from the perceptual field presented to him. This may or may not be the pattern intended for him. 5) The perceiver works within a frame of reference mainly constituted by his former experience, and this gives final interpretation and meaning to his acknowledged response.

Requirements of Insightful Models. These five characteristics of insight listed by Lee suggest five basic characteristics for insightful models shown in Figure 1.

The first characteristic of perception, organizing the field into 'figure' and 'ground', maps into the model development of insightful models. That is, modeling consists of specifying the primary variables and their relationships (the figure) so as to represent the actual business situation, at the same time excluding

THEORY OF LEARNING	CHARACTERISTICS OF INSIGHT-GENERATING MODELS & ANALYSIS TOOLS
PERCEIVER ORGANIZES FIELD FIGURE & GROUND	MODEL SHOULD INCLUDE KEY CHARACTERISTICS, IGNORING INSIGNIFICANT DETAILS
PERCEIVER LOOKS FOR SIMPLICITY,REGULARITY	MODEL SHOULD DETERMINE LINEAR & NONLINEAR RELATIONSHIPS BETWEEN KEY VARIABLES
PERCEIVER ORGANIZES HIS OWN PATTERN	MODEL SHOULD INCLUDE PATTERN RECOGNITION, PATTERN GENERATION, PATTERN COMPARISON CAPABILITIES
TENDENCY TO PERCEIVE INCOMPLETE MATERIAL AS COMPLETE	MODEL PATTERNS SHOULD BE GENERALIZABLE
FORMER EXPERIENCE FORMS FRAME OF REFERENCE	COMPREHENSIVE DATABASE OF PATTERNS & WHAT-IF INSTANCES

Figure 1. Characteristics of Insight and Insight-Generating Models

inconsequential and irrelevant detail (the ground). This is the same logic which dictates that a road map should include cities, towns, large lakes and highways; if more detail were included, the map would become too cluttered to be useful.

The perceiver looking for simplicity equates to the model requirements of identifying key variables, and specifying and developing relationships among them. Identification of key variables obviously simplifies the problem, but the specification and development of the relations of these key variables is equally important. Kohler states that "we have to recognize that all problems with which we may be confronted, and also the solutions of such problems are matters of relations [Kohler, 1965, p. 143]. He further states that "not only does our understanding of the problem demand our awareness of certain relations, we can also not solve the problem without discovering certain new relations" [Kohler, 1965, p. 144]. Thus, not only must an insightful model reflect the decision maker's known relations, it must also help him develop and explore new relations.

The organization of patterns from the perceptual field maps into the pattern recognition requirement of insightful models. In decision making, this consists of recognizing patterns of events and key factors which occurred in the decision maker's past, the decision maker's reaction (or decision) to that pattern and the eventual outcome of that reaction or decision [Hayes, 1989]. The importance of pattern development and recognition has been demonstrated by researchers in showing why chess masters are better than weaker players. De Groot [1965] found that the primary difference was that chess masters recognized chess positions much better than weaker players. Simon and Gilmartin [1973] estimated that chess masters recognized board position better because they can recognize a board pattern and compare it to between 10,000 and 100,000 chess patterns stored in their memory. Such patterns are developed from at least a decade of intense preoccupation of the game.

The tendency to perceive incomplete material as complete maps into the generalization requirement for insightful models. That is, insightful models must be able to generalize on the tasks for which they have been designed, enabling the model to provide the correct answer when presented with a new input pattern that is different from those on which it was based [Dayhoff, 1990, p. 13]. Further, such models should, to a degree, be insensitive to minor changes in the input patterns; i.e., they should be able to see through some distortion and noise to the true pattern that must be recognized and evaluated [Wasserman, 1989, p. 2].

The importance of the perceiver's former experience maps into the model requirement of a comprehensive and integrated database of relevant historical patterns and what-if instances specified by the decision maker. The historical patterns form a basis for both model validation and pattern recognition. The what-if instances form the basis of investigative patterns which the decision maker has recently explored. Geoffrion [1975] states that some business analyses have included as many as 215 separate instances. Both historical and what-if instances can be used to develop generalizations of patterns for enhancing decision making and predicting the impacts of those decisions.

Current Analysis Tools

Classical Analysis Tools. Three stages of model use (formulation,solution and analysis) have historically received unequal treatment by researchers. Most research has been devoted to algorithm development; i.e., model solutions. More recently, with the advent of model management systems, model formulation has been receiving more emphasis. However, proportionately little time has been devoted to the development of analysis tools.

This section reviews model analysis tools. We begin by discussing classical analysis tools such as graphs and regression analysis. We then review the analysis tools currently available in each of the three most popular modeling paradigms (optimization, simulation and spreadsheets), discuss some of the potential analysis technologies currently being developed, and draw several conclusions based on this literature review.

As mentioned in the previous section, one of the key requirements of model analysis is the recognition and development of patterns of interactions between model variables and parameters and the model solution or output. The classical techniques for recognizing and testing such patterns are graphs and regression analysis.

Graphs are useful in visual recognition of patterns; i.e., the human eye has long been recognized as a productive tool in integrating data and determining various patterns. However, graphs and visual pattern recognition are limited to 3-dimensional patterns and have no ability to determine quantitative measures concerning goodness of fit.

Regression analysis is a statistical analysis technique which finds the best fitting curve which minimizes the sum of the squares of the deviations of the observed values of the dependent variable, or variable of interest, from those predicted values based on a set of one or more independent variables [Mendenhall and Beaver, 1992]. For example, in simple linear regression between one independent variable and the dependent variable, y, regression analysis uses the method of least squares to find values for the constants, b_0 and b_1, in the equation $y = b_0 + b_1 x + e$, where e is assumed to be a normally distributed random error term with mean = 0. Regression analysis is equally applicable to multiple independent variables and to nonlinear functions. Correlation analysis, usually done in conjunction with regression analysis, provides both correlation coefficients for individual independent variables and a multiple coefficient of determination for the regression model as a whole. This latter is defined as the percentage of total deviation in the dependent variable which is explained by the model; i.e., it provides a goodness-of-fit measure for the regression model.

Regression analysis has several limitations in practice. One limitation is that the functions in a nonlinear model must be known a priori; e.g., one must know in advance that the dependent variable depends on the square or cube of some independent variable, the cross product of two or more independent variables, etc. This places a high premium on pre-analysis of the problem and its relationships. In addition, regression assumes that the error terms are normally distributed and random for every value of the independent variable(s).

Optimization-based Model Analysis Tools. Optimization-based modeling has been one of the most active research areas since the Simplex method was introduced by Dantzig in the mid-1940's. By far, most of the research emphasis has historically been devoted to the development of specialized algorithms designed to solve a myriad of problem classes. For example, Sharda [1992] reports 49 PC-based packages available commercially which solve linear programming (LP) models, some capable of solving large problems with as many as 367,000 non-zero variables and 16,000 rows or 69,000 columns on a desktop computer.

Recently, more research effort has been invested in mathematical modeling languages to improve the efficiency of model formulation and specification. [Steiger and Sharda, 1991a] Research in modeling languages started in the 1960's with the development of matrix generator/report writing (MG/RW) systems such as MaGen [Haverly Systems, 1977], GAMMA [Bonner & Moore, 1989] and DATAFORM [Ketron, 1970]. (MG/RW's are procedural languages which integrate the input matrix generation and report writing steps of the modeling process.)

Modeling languages (ML), which extend this automation of translations, are declarative rather than procedural languages; i.e., they express what should be computed rather than how to compute it. One group of ML, the algebraic languages, are "declarative languages which accept, as input, the modeler's (algebraic) form of an LP in a notation that a computer can interpret, and generate the algorithm's (matrix) form as output" [Fourer, 1983]. These algebraic languages support the constraint, or row, form of a model and are characterized by 1) extensive use of domains over sets, 2) subscripting capabilities, 3) indexed sum capabilities, 4) simple and straight forward arithmetic expressions, 5) symbolic descriptions of set and parameter data (preferably with a natural separation of model and data), 6) data instantiation capabilities, and 7) the ability to impose simple restrictions on parameters and variables [Greenberg and Murphy, forthcoming; Steiger and Sharda, 1993b]. Examples of these algebraic modeling languages include GAMS [Bisschop and Meeraus, 1983; Brooks et al., 1899], LINGO [Cunningham and Schrage, 1990], LPL [Huerleman, 1989] and MPL [Maximal Software, 1989].

The most recent modeling languages support views of a model other than the classic algebraic view. For example, PAM [Welch, 1987], MIMI [Chesapeake Decision Sciences, 1988], and MathPro [MathPro, 1989] support the block schema view. Further, current research efforts are developing MLs which support additional views to both conceptualize models [Baldwin, 1990] and to specify and analyze them; e.g., the process network view [Chinneck, 1990], the fundamental graph view of NETFORMS [Glover et al., 1977, 1990], interactive NETFORM views [Jones, 1990, 1991; Kendrick, 1990; Steiger et al., 1991 and 1993a], fisheye views [Mitta, 1989; Furnas, 1986], frame views [Krishnnan, 1988] and interactive block/algebraic views [Ma et al., 1989; Murphy and Stohr, 1986]. Structured Modeling [Geoffrion, 1987, 1989, 1992a 1992b] epitomizes multiple view modeling by offering many views while representing all in a single internal schema. However, much less research and development effort appears to have been devoted to model analysis, even though "comprehension is the present bottleneck in using large-scale models -- in particular, linear programs" [Greenberg, 1983]. Model analysis includes validating the model and building confidence that it properly reflects the real world situation, determining the impacts of changes to input parameters on the model solution, explaining model results, comparing and explaining differences created by multiple model cases, and translating the "best" solution into implementable and auditable decisions.

Currently, there are only three software packages available for the analysis of optimization-based models. One is PERUSE [Kurator and O'Niell, 1980], a FORTRAN analysis program which provides an interactive capability to query the LP matrix and solution values. PERUSE is designed to provide computer assistance to modelers and analysts for debugging, verifying and analyzing models and model instances.

Another analysis package for LPs is ANALYZE [Greenberg, 1983, 1988, 1990]. ANALYZE is also a FORTRAN program which extends the analysis capabilities of PERUSE by including routines to aid in the 1) determination of why the model results occurred (i.e., trace causation), 2) documentation and verification of the model, and 3) simplification of the model (e.g., search and identification of embedded structures such as NETFORMS [Glover et al., 1977]). ANALYZE includes a wide variety of commands which allow the analyst to, among other things, display the row rim/solution information, find embedded cycles, find null and singleton rows and columns, determine implied bounds on primal quantities and dual prices, report summary statistics, trace a complete flow path to a designated output and report value statistics.

The primary limitation of ANALYZE is that it is limited, by design, to analyze only one model or a model instance; i.e., it has no capabilities to compare two or more instances from various what-if cases so as to determine how and why they differ. Further, ANALYZE is limited to LP models and thus cannot be used in other modeling paradigms. Finally, the model itself must be generated with ANALYZE in mind so that the appropriate 'hooks' for ANALYZE can be included in the model formulation and specification.

The third analysis technique for LPs is called 'candle-lighting' and was developed as part of the Coast Guard's KSS project [Kimbrough et al., 1990, 1992]. This is a Prolog-based system implemented on a Macintosh platform. It is used for the interactive analysis of LPs, although the authors state that "the ideas apply to management science modeling of all sorts, including mathematical programming in general, queuing modeling, multiple attribute utility models as well as LP" [Kimbrough et al., 1992, p. 4].

Candle-lighting focuses on searching for submodels or surrogate models in an LP. Submodels consist of an equation (i.e., model) that "when executed determines the value for the parameter of the main model" [Kimbrough, 1992, p. 126]; e.g., an objective function cost coefficient may really be a function of labor hours, cost of labor and an inflation factor. A surrogate model represents a rule of thumb about some underlying relationship between the main model parameter and other, more elementary, components. Candle-lighting uses these submodels and surrogate models to expand normal sensitivity analysis to answer such questions as: Do any submodels have common variables, and if so, how much can these common variables change before the basis changes? What is the cost of changing the results of the model given that we can act so as to alter assumptions of the models?

The primary limitation of candle-lighting analysis, at least as currently implemented is that it also deals with a single model or model instance instead of a set of instances; i.e., it has no capabilities to compare multiple instances for differences and explain those differences. Further, while the concepts are applicable to other modeling paradigms, the software modifications required for use in other paradigms would be extensive.

Simulation-based Analysis Tools. Like optimization-based modeling, simulation-based modeling research has centered around the development of generalized modeling capabilities and, more recently, around special purpose software development (e.g., for manufacturing, logistics, communication networks, etc.) and integration between simulation software and other programs (e.g., spreadsheet models). This research and development has produced 56 commercially available discrete event simulation packages for microcomputers or workstations [Swain, 1991]. The primary output created by these packages to aid human analysis has historically been statistical summaries and, more recently, statistical graphs. Several years ago, some vendors introduced animation to help analyze the simulation results. Such animations are unrivalled sales tools, in that they allow the user to 'see' what the system acts like under various assumptions.

In general, the tools with which to analyze simulation output have been almost nonexistent. Swain [1991] lists the use of artificial intelligence/expert systems (AI/ES) as a natural future research and development area to make comparisons between simulated alternatives and to interpret output or infer operating strategies based on the simulation results.

In fact, there are only two research systems we know of that are designed to help analyze a simulation model. One is GODDESS [Rangaswamy and Fedorowicz, 1983], a system equipped to perform some of the same analysis functions as ANALYZE does for LP-based models. The other is I-KBS [Reddy, 1985], an intelligent system which helps managers conceive and develop simulation models as well as learn and verify interactions between model entities. I-KBS provides run-time graphics to aid decision makers in their understanding of model variables and detect errors in spatial relationships.

Spreadsheet-based Model Analysis Tools. Computerized spreadsheet are, in general, a newer technology than either optimization or simulation. However, within the past decade spreadsheets have become very widely used for modeling in industry. The software development has been very rapid and currently there are over 50 commercially available spreadsheet systems for the microcomputer and mainframes.

However, there are only four spreadsheet tools available for enhancing the analysis capabilities: Lotus 1-2-3 (with 3-dimensional spreadsheets) and compatible, Lotus 1-2-3 (with the @RISK add-in), EXCEL (with Scenario Manager) and IFPS (with explanation capabilities). Each of these is discussed below.

Lotus 1-2-3 (with 3-dimensional spreadsheets). Lotus 1-2-3 is arguably the most widely used PC-based spreadsheet modeling language in industry today. It offers a wide range of data manipulation and calculation options, statistical functions, graphical capabilities, etc. It has, in recent years, included a linear programming optimization add-in and most recently added a 3-dimensional spreadsheet capability. 3-dimensional spreadsheet allow the decision maker to store multiple instances of a model, with each instance corresponding to a specific what-if case; i.e., with what-if instances stored along the third dimension, each instance varying from others by one (or a few) values specified for the critical variable(s). Once the instances are specified, the decision maker can manually analyze the results via graphs or summary printouts. One can also build a master spreadsheet and use instance spreadsheets to generate various statistical summaries across instances.

The primary limitation of the Lotus 3-D capability is that it provides a mechanism to keep related model instances, generated from what-if questions, tied tightly to the base model where they can be referenced, manipulated and analyzed either individually or as a group. This is especially advantageous when one is analyzing the effects of multiple changes made to a base model.

The primary limitations of the Lotus 3-D capability is that it requires most comparative analysis, with the exception of certain statistics, to be done manually, without providing software routines to aid planning or analysis of instances. In addition, it provides no efficient way to systematically analyze risk reflected in uncertain input variables.

<u>Lotus 1-2-3 (with @RISK)</u>. @RISK [Palisade, 1991] is an add-in to Lotus 1-2-3 which allows the decision maker to explicitly model and analyze risky business situations using appropriate probability distributions for random variables instead of the expected values which would normally be used. More specifically, @RISK allows one or more cell values to be defined using @functions representing one of over thirty different probability distributions; e.g., sales might be

defined as being normally distributed with a mean of 500 and a standard deviation of 50 via @NORMAL(500,50). @RISK then uses simulation (with either Monte Carlo or Latin Hypercube sampling technique) to produce distributions of possible results for each appropriate output cell. These graphic displays include relative frequency distributions, cumulative probability curves, graphic overlays for comparing distributions, statistical summary reports, and probabilities of occurrence for any target value in the distribution.

To use @RISK, the decision maker creates a Lotus 1-2-3 spreadsheet as one normally would. In such a model there may be one or more uncertain, or random variables. A probability function is specified for each random variable using a Lotus @function in conjunction with an @RISK distribution and its specification format; e.g., @TRIANG<30,5,80>.

The primary advantages of the @RISK add-in is that it adds the ability to include risk in spreadsheet models and determine its effect on the primary output parameters. In addition, @RISK provides a file of the most recently requested simulation instances which could perhaps be used by other add-in functions as the basis for further analysis.

The primary limitation is that @RISK provides no indication to the decision maker of which variable(s), random or deterministic, are the most important in influencing the output results. Neither does @RISK provide any direct tools for sensitivity analysis, what-if planning, results interpretation, or explanation facilities.

Microsoft EXCEL's Scenario Manager. Microsoft Excel (version 4.0) is a PC-based spreadsheet modeling language with business graphics, an optimization solver, and a database capability [Microsoft, 1992]. One of its latest additions is the Scenario Manager which helps the decision maker create and save instances of a model representing multiple what-if scenarios. This add-in allows the decision maker to create summary reports which include both input values and results for a specified set of scenarios.

With Scenario Manager, the decision maker first creates a base case spreadsheet model. Then, if one wishes to run several what-if scenarios reflecting various values of, say, gross revenue (GR) and cost of goods sold (COGS), one can specify the cell names, references or cell ranges of the 'changing cells' and input values for each changing cell and for each scenario in a table format. One can also enter a unique name for each set of changing values (i.e., for each scenario). To run the spreadsheet model using a specific scenario, one chooses the Scenario Manager function and selects the appropriate scenario name. To create a summary report of several different scenarios, one first chooses the Scenario Manager menu, selects the Summary Button within the menu, selects the appropriate scenario name(s) to be included in the report, and specifies the references or names of the desired 'Report Cells'. Scenarios can be changed, added and/or deleted interactively by using a combination of menus, buttons, pop-up boxes and prompts. In addition, spreadsheet 'changing cells' can be added or deleted from existing scenarios.

The primary advantage of Excel's Scenario Manager is that it provides an efficient database of what-if scenarios without unnecessary duplication of input and intermediate variables and parameters. In addition, it provides a way to create and view (side-by-side) the results of multiple what-if scenarios for manual analysis.

The primary limitation of Scenario Manager is that is provides no tools for automated sensitivity analysis, what-if instance planning, results interpretation, user interrogation, explanation facilities, etc. In addition, it provides no summary measures over scenarios.

<u>IFPS/Plus</u>. IFPS/Plus [EXECUCOM, 1992] is another popular spreadsheet modeling system which is available on the mainframe or microcomputer. Like Lotus 1-2-3, IFPS has a full range of modeling capabilities, including the ability to save model instances for later analysis. However, the most interesting analysis feature is its explanation commands which provide tools for interpreting and explaining the difference between a specified variable in two different instances or in two different periods of the same instance.

This capability is based on the ROME/ERGO/ROMULUS research systems developed at Carnegie-Mellon University [Kosy and Wise, 1984, 1986; Wise and Kosy, 1986].

The explanation tools "trace the path of influences in a model or consolidated structure so you can see not only which variables are most important from a definition viewpoint, but also which ones had the most influence on the change in values" [EXECUCOM, 1992]. For example, the WHY command lets the user ask questions about why a value changes , providing the answer in terms of the smallest subset of variables that accounts for most, say 80%, of the change. Variables in this subset are further categorized according to whether they have a positive or negative (counteractive) effect on the variable in question.

The primary advantage of the IFPS software is that it represents virtually the only capability in any modeling paradigm which has the capability to analyze more than one model instance at a time and explain why they differ. This is a powerful capability, especially in view of the heavy real-world use of what-if analysis and the user's need for automated tools to help analyze those multiple what-if instances.

The primary limitations of the IFPS/PLUS explanation facilities is that the comparison capabilities are limited to at most two model instances and to those variables which are computed by the same formulas. In addition, as noted as research areas by Kosy and Wise [1986], the methodology needs to expand the explanations to more than one context at a time (e.g., more than one column at a time), and to find the causes of differences between actual performance and planned/budgeted performance modeled instances.

Potential Analysis Technologies

As noted in the preceding section, the development of analysis tools is currently in its infancy when compared to the formulation and solution phases in any model paradigm. However, the need for such analysis tools is critical especially with the recent widespread explosion of desk top computing and the associated growth in end-user computing; i.e., the decision makers building and running models and analyzing their results.

Researchers have seen this growing need for analysis tools and have proposed various technologies which could be applied. These technologies include artificial intelligence/expert systems (AI/ES), influence diagrams, case-based reasoning and neural networks. Each of these technologies, along with its potential application to model instance analysis, is discussed briefly below.

AI/ES. Several researchers have suggested the use of AI/ES in model instance analysis. Brennan and Elam [1986] suggest six such areas: 1) detection -- what is important in the model, 2) validation -- do results from the model make sense, 3) natural language discourse, 4) guidance --

which what-if instance should be tried next, 5) explanation -- why is a solution recommended and 6) exploration -- how can the model be used as a predictor of the system behavior being modeled.

Other researchers also recommend several specific areas of analysis in which AI/ES could be used. Elam and Konsynski [1987] suggest six analysis tasks and several interpretation tasks which could be enhanced by using AI. These include 1) matching -- identifying and testing the applicability of a model for a particular problem, 2) expecting -- detecting and explaining abnormal model behavior, 3) planning -- determining ways to perform analyses to reach a specified goal, 4) Causation -- identifying causal relationships between model variables, 5) recommending -- choosing an alternative, 6) synthesizing -- generating new models from model fragments, and 7) explaining -- generating explanatory models.

As mentioned in section 2.3.1.2 above, Swain [1991] suggested expert systems to be used to analyze simulation runs, both to make comparisons between model instances and for optimization of the system. He also suggested that neural networks might someday be used to infer operating strategies based on multiple simulation runs or even as a surrogate for simulation. These uses of AI/ES "should significantly strengthen the utility of simulation by making simulation easier to use and by increasing the complexity of the system that could be modeled" [Swain, 1991, p. 92].

Influence Diagrams. Influence diagrams are decision-analysis tools which provide a simple graphical representation of the relationships of the decisions and uncertainties in a decision problem [Howard and Matheson, 1984]. These diagrams consist of decision nodes, chance nodes, deterministic nodes value nodes and two types of directed arcs which identify all of a model's essential elements and relationships [Howard, 1990; Clemen, 1990]. Each node type is represented in an influence diagram as a specific shape, with specific rules used for evaluating the decisions; e.g., a dashed arrow from a chance note to a decision node means that the outcome of the chance event is known when the decision is made. It should be noted that influence diagram are cyclical; i.e., there is no path which leads back to any given starting point.

In addition, influence diagrams provide a snapshot of the decision making environment at a given point of time, including the details such as outcomes choices and payoff at each node. These details are usually suppressed to simplify the representation, but are utilized in the evaluation of the influence diagram.

The primary potential use of influence diagram in the analysis of models and model instances is in the intelligence of the model structure which they contain. That is, the influence diagram is, in effect, a pictorial or graphical knowledge base of relevant dependencies and influences present in the model they represent. Given that their graphical nature makes them understandable be decision makers, influence diagrams represent a potential communication device for both model specification/validation and model analysis.

Case-Based Reasoning. Case-based reasoning (CBR) is characterized by the decision maker making his inferences and decisions based directly on previous cases recalled from memory rather than on general knowledge [Kolodner, 1987]. That is, the decision maker tries to avoid or reduce the potential for failure by recalling previous failures and avoiding the associated pitfalls or changing key factors in those previous failures. He also can speed the decision making process by not having to generate and evaluate all alternatives from scratch. Finally, he can generalize from the attributes of recalled cases to improve decision making in the future [Hammond, 1987].

CBR involves, in the simplest case, the following set of steps: 1) previous case recall, 2) focus on the relevant parts of the recalled case (i.e., the decision maker's current reasoning goals if the recalled case was successful or the recalled case's reasons for failure if it failed), and 3) making a case-based inference or decision based on these parts of the previous case which are appropriate for the current decision [Kolodner, 1987].

The primary advantages of CBR is that it generates knowledge from stored cases (or model instances, in our terms). Assuming that the decision maker's specification of instances has some directed, though unspecified goal (e.g., to learn more about the interaction of variables and their effect on the system being modeled), the set of instances generated during what-if sessions should have some bit(s) of knowledge, buried (perhaps perspicaciously) within the instances themselves. Thus, it makes perfect sense to analyze such instances and derive as much knowledge from them as possible. CBR works on this principle.

The primary limitations of CBR are: 1) its initial screening of cases and corresponding loss of potential knowledge contained in them, 2) its lack of ability to generate multiple variable relationships, explicitly, to help the decision maker better understand the system, 3) its algorithm limitation associated with changing one or more of the variables in a recalled case to change its outcome, and 4) its simulation of changed cases to test the efficacy of any attribute changes.

Neural Networks. Neural networks are biologically inspired by the architecture of nerve cells in the human brain; i.e., they are massively parallel networks consisting of neurons (or nodes) interconnecting synapses (or arcs) arranged in multiple layers with a large number of interconnections. Neural networks are not programmed; they learn by example. That is, they accept, as input, a training set consisting of a group of examples from which the network can learn. Each example, in turn, consists of values for each input variable and a correct value from the output variable. Neural networks use these training examples to adjust parameters associated with the interconnections between neurons; the rate of learning is dependent on the rate of interconnection updates.

Neural networks excel at problems involving pattern mapping, pattern completion and pattern classification. They are especially adept at completing noisy and incomplete patterns (i.e., those with segments missing), translating financial time series data into financial predictions, and analyzing and recognizing patterns in visual and acoustic data. Another area in which they excel is in "generalizing on the tasks for which they are trained, enabling the network to provide the correct answer when presented with a new input pattern that is significantly different from the inputs in the training set" [Dayhoff, 1990].

The primary advantages of neural networks include the following: 1) neural networks are self-organizing and learn by example so there is not need to program them to decipher patterns, 2) once trained they are very fast in classifying patterns, 3) they can generalize on the task from which they have been trained, and 4) they are able to see through some distortion and noise to the true patterns that must be recognized [Wasserman, 1989].

The primary disadvantages of neural networks include: 1) the current training algorithms are slow, sometimes taking days to train on complicated training patterns, 2) they require the researcher to set certain sensitive parameters (e.g., the learning rate) which can significantly affect the final solution, and 3) neural network architecture requires significant expertise in some problem types.

Taxonomy of Instance Analysis Tools

Figure 2 provides a taxonomy of instance analysis tools. This taxonomy is based on 1) whether the tools analyze a single model instance, two instances or more than two instances and 2) whether the tool addresses the question "How much does the solution change?" or the more provocative question "Why does the solution change?". This latter question is the foundation for generating insights into the decision making environment [Geoffrion, 1976].

As can be seen from the figure, we found only six analysis tools (over all modeling paradigms) which address the question "Why does the solution change?", and none of these six tools analyze more than two instances simultaneously. If insight is the product of the simultaneous processing of many inputs, as is stated in the theory of learning literature [Lee, 1965; Lonergan, 1958; Logan and Ferraro, 1987] and is suggested in the theory of hemispheric specificity of human consciousness [Ornstein,

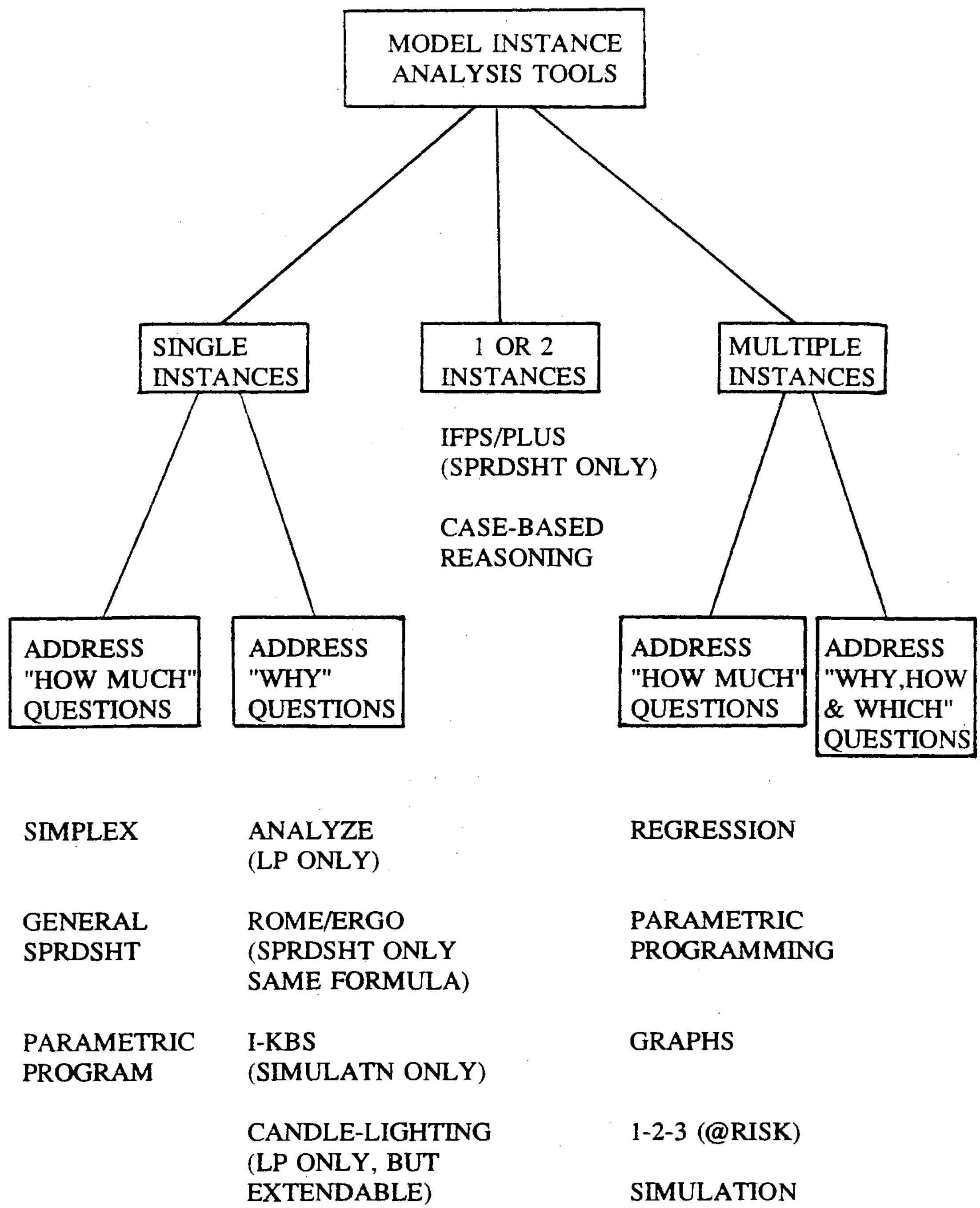

Figure 2. Taxonomy of Instance Analysis Tools

1976, 1985; Lee, 1965], then model analysis must deal with many model instances simultaneously if it is to enhance the decision maker's insight generating capability. Further, pattern recognition, classification and/or completion capabilities should be a part of the insight-generating tools.

Conclusions From Literature Review

There are three conclusions which can be drawn from the above literature review concerning currently available analysis tools and potentially applicable technologies. These conclusions include the following:

1) little research appears to have been done on analysis, especially in comparison to model formulation and solution.

2) little research, with the notable exception of ROME [Kosy and Wise, 1986] and case-based reasoning, has been done on the analysis of multiple instances as a source of data and intelligence, and

3) several current technologies provide excellent potential in the generation of insightful analysis in a MMS environment.

CHAPTER III

MODEL ANALYSIS TASKS AND THEIR MATHEMATICAL STATEMENTS

Introduction

Decision making is described by Simon [1966] as a three stage process consisting of intelligence, design and choice. Several authors have suggested that human analysis, amplified and enhanced by currently available technologies in a decision support system environment, can be used to develop helpful insights in each of Simon's three stages; i.e., insights which improve the understanding of the problem and its environment, insights which enhance both the quantity and quality of the alternatives developed for consideration, and insights which improve the selection of the best alternative. Brennan and Elam [1986] and Elam and Konsynski [1987] suggest several DSS/MMS analysis and interpretation tasks which could be improved by use of various technologies. We combine these into seven specific analysis tasks to improve insights into the decision problem: validation, sensitivity/planning, comparison, causation, recommendation, implementation and post audit. Each analysis task is discussed individually below, illustrated with example questions based on a spreadsheet model depicting the following business situation.

Hunt-Wesson Foods, Inc., produces tomato products, cooking oil, matches, puddings, shortening and many other products at 14 locations (Wesson refineries, Hunt canneries, and copackers). At the start of the study, it distributed nationally through 12 distribution centers. Annual sales were in the vicinity of $450 million and growing fairly steadily. Transportation was by common rail and by both common and contract truck carriers, with extensive use of the storage-in-transit privilege for large rail-supplied customers. The company's policy was to service each customer from a single distribution center for all products. A few years ago, the company decided to undertake a planning study because it faced pressing distribution-center expansion and relocation issues. Management recognized that these issues, while seemingly regional in character, were in fact interwoven with the company's entire national distribution system design. It decided to employ a computer-based method that would not only resolve the immediate questions in their proper national perspective, but would also comprehensively re-balance all distribution center locations, assignment of customers to distribution centers and aggregate annual product flows through the system [Geoffrion, 1979].

Validation

Validation consists of making comparisons between model results and actual, real world situations. As suggested in Hillier and Lieberman [1986, p. 20 and 23], we divide model validation into three parts: 1) a model is 'correct' if, when the model is set to reflect some group of historical situations, its solutions differ from the actual historical outcomes by come constant amount, 2) a model is 'accurate' if a change in input parameter and/or variable value produces a reasonable change in the solution, and 3) a model is 'consistent' if the model solutions are highly and positively correlated with corresponding historical situations; i.e., if the model explains a high percentage of variation from average of a group of historical situations.

For example, if the model is constrained to supply the same products from the same warehouses actually used in several previous years, does the model solution vary from the actual costs by some constant amount each year? Is the model output highly correlated with changes in the actual costs during those years? Does adding a warehouse or changing a customer zone from one warehouse to another in the model result in a reasonable change in fixed costs, variable costs and total costs?

Mathematically, validation is specified as follows. Let $m \in \{1, 2, \ldots M\}$ = a set of subscripts to identify historical situations.

$i \in \{1, 2, \ldots I\}$ = a set of subscripts to identify independent variables.

H_m = one in a set of M historical situations, each of which is represented by a set of I independent variables, x_i.

$x_{i,m}$ = a model variable $i \in I$ evaluated at the value occurring at $m \in M$ time in history.

S_m = one in a set of M historical outcomes based on the then-current actual values of each $x_{i,m} \in \{x_{i,m}\}$ corresponding to the historical situation.

O_m = one in a set of M dependent model output solutions evaluated at $\{x_{i,m}\}$ corresponding to some historical situation.

O'_m = one in a set of M optimal dependent output solutions evaluated at optimal values $x'_{i,m}$ corresponding to the historical situation.

Then, the model is <u>correct</u> iff:

$\forall m \in M$: $O_m = a*S_m + b$ where a and b are constants.

The model is <u>accurate</u> iff:

$\forall m \in M$: $O'_m = a*S_m+b + \sum_i c_i(x'_{i,m} - x_{i,m})$ where $x_i \neq x'_i$ for one (or a few) x_i, and c_i is the effect of x_i on the solution.

The model is <u>consistent</u> if:

$\mathrm{Corr}[S_m,O_m] \doteq 1.0$

The results of validation may be model modification, model validation, decision maker confidence in the model and/or newly developed insights into the decision making environment.

Sensitivity Analysis/Planning

Sensitivity analysis consists of "identifying the relatively sensitive parameters (i.e., those which cannot be changed without changing the solution), to try to estimate those parameters more closely, and then to select a solution which remains a good one over the range of likely values of the sensitive parameters" [Hillier and Lieberman, 1986, p. 90]. Planning addresses the questions concerning which what-if case should be tried next. It can be viewed as an extended sensitivity analysis. For example, if incoming shipping costs interact with volume shipped and warehouse location, and 15 model cases have been run to test the effects of these interactions on total costs, which what-if case should be run next to find a lower cost level?

The planning function also extends to determining the best settings for certain solution technique; e.g., the time-step in simulation models, the learning rate in neural network models, the step-size in gradient search models, etc. For example, in a neural network model, if the solution repeatedly converges rapidly to a local minimum and cycles thereafter, is there a better (than random) set of initial weights which tends to result in better model solutions?

Given a set of solved model instances or what-if cases, the planning function can be viewed as a form of multiple, nonlinear regression analysis in which one wants to determine which parameters are most important in causing the desired change in the variable of interest. This analysis is complicated by the fact that the decision maker does not know which subset of parameters should be included in the regression, which power(s) these parameters should be raised to, or what degrees of interaction between parameters is most appropriate.

Sensitivity analysis is well defined and discussed in the optimization modeling literature; e.g., in Hillier and Lieberman [1986]. However, for other modeling paradigms (e.g., spreadsheets) it is less well defined and mathematically can be represented as follows.

Let tgt^- = the lowest solution which management is able to accept.

tgt^+ = the highest solution which management is able to accept.

And:

$\forall\ i \in I$: $p_{i,min} = \min_{all\ m} \{p_{i,m}\}$

$\forall\ i \in I$: $p_{i,max} = \max_{all\ m} \{p_{i,m}\}$

$\forall\ i \in I$: $p_{i,base}$ = base case value where p_i = parameter or variable value in the model.

Then:

$\forall\ i \in I$: $f(p_{1,base}, p_{2,base}, \ldots (p_{i,min}), p_{i+1,base} \ldots p_{I,base})$

$< tgt^- \Rightarrow p_i$ is a sensitive parameter or variable.

$\forall\ i \in I$: $f(p_{1,base}, p_{2,base}, \ldots (p_{i,max}), p_{i+1,base} \ldots p_{I,base})$

$> tgt^+ \Rightarrow p_i$ is a sensitive parameter or variable.

Planning can be represented mathematically as follows. Let some forecasting technique (FT) train on a set of M model instances using a subset of the sensitive variables as the training values and the insensitive variables, the remainder of the sensitive variables and the model objective function values (i.e., dependent, or solution, variables), y_j, be the forecasting model attributes. That is,

Let the sensitive variables = $\{s_k\} \in \{x_i\}$

training variables = $\{t_l\} \in \{s_k\}$

attributes of FT = $\{\{x_i\} - \{t_l\} \cup \{y_j\}\}$

$O(t_l)$ = forecasted value; i.e., the output of the forecasting technique

Then

$O(t_l) = f(\{x_i\} - \{t_l\} \cup \{y_j\})$

To plan the next instance, change one or more components of the model output solution, y_j, by some amount, δ, and forecast the appropriate values of the key variables, t_l, based on the modified value of y_j; i.e.,

$O(t_l) = f(\{x_i\} - \{t_l\} \cup \{y_1, y_2, \ldots (1 + \delta) y_j, y_{j+1}, \ldots y_J\}$

$= t_1, t_2, \ldots$

Let $t'_{l,r}$ be a user-specified reasonable upper bound on x_i

If $t_l < t'_{l,r}$ $\forall$ l then

Planning can be represented mathematically as follows. Let some forecasting technique (FT) train on a set of M model instances using a subset of the sensitive variables as the training values and the insensitive variables, the remainder of the sensitive variables and the model objective function values (i.e., dependent, or solution, variables), y_j, be the forecasting model attributes. That is,

Let the sensitive variables = $\{s_k\} \in \{x_i\}$

training variables = $\{t_l\} \in \{s_k\}$

attributes of FT = $\{\{x_i\} - \{t_l\} \cup \{y_j\}\}$

$O\ (t_l)$ = forecasted value; i.e., the output of the forecasting technique

Then

$$O\ (t_l) = f(\{x_i\} - \{t_l\} \cup \{y_j\})$$

To plan the next instance, change one or more components of the model output solution, y_j, by some amount, δ , and forecast the appropriate values of the key variables, t_l, based on the modified value of y_j; i.e.,

$$O\ (t_l)=f(\{x_i\}-\{t_l\} \cup \{y_1, y_2, \ldots (1 + \delta\ )y_j, \quad y_{j+1}, \ldots y_J\}$$
$$= t_1,\ t_2,\ \ldots$$

Let $t'_{l,r}$ be a user-specified reasonable upper bound on x_i

If $t_l < t'_{l,r}$ $\forall$ l then

$O(t_l)$ represents a good estimate of an instance which may provide a better solution in y_j and should be run in the model.

If $t_l > t'_{l,r}$ then set $t_l = t'_{l,r}$ and remove it from the set of $\{t_l\}$ and add it to $\{x_i\}$

$O(t_l)$ represents a good estimate of an instance which may provide a better solution in y_j and should be run in the model.

If $t_l > t'_{l,r}$ then set $t_l = t'_{l,r}$ and remove it from the set of $\{t_l\}$ and add it to $\{x_i\}$

If $t_l = \{\}$, then retrain the FT with a different set of key variables; else rerun the trained FT with a different set of key variables, $\{t_l\}$.

The primary result of good sensitivity/planning analysis should be a better understanding on the part of the decision maker of the sensitivity of the solution to changes in model parameters or solution settings and greater insight into the critical factors of the decision making environment.

Comparison

Comparison consists of detecting, explaining and suggesting reasons for surprising or unexpected model behavior; e.g., Why does the addition of one warehouse change the total costs so much? or Why does increasing a customer's demand cause its source warehouse to change? These comparisons may concern different parameters or decision variables in a single model case (e.g., costs in different years of a multi-period model) or different variables in different model cases (e.g., cost differences when demand for a given customer increased by 20%). Mathematically, comparison is expressed by Kosy and Wise

[1984, 1986] in terms of the focus context versus the referent context; e.g., for the question, "Why do profits go up in 1973 from 1972 levels?", the focus context is 1973 profits and the referent context is 1972 profit. "To be comparable, the derivations must involve the same formula, say f, so that the difference, Δy, comes from evaluating f in the focus context versus the referent context [Kosy and Wise, 1986, p. 27].

$$\Delta y = f(a_f, b_f, c_f \ldots) - f(a_r, b_r, c_r, \ldots)$$

$$y = (f(a,b,c, \ldots) = \text{a model solution.}$$

Let $S = \{a,b,c, \ldots\}$ denote the set of model variables, and

$X \subseteq S$ denote a minimum subset of variables which can account for a substantial fraction (say 80%) of Δy.

and $\mathscr{E}(X,y)$ = a measure of significance which measures the effect of variables in X on y in the focus context relative to the referent context. Then

$$\mathscr{E}(X,y) = y_f - g(Z)$$

where Z contains values of variables in X evaluated in the referent context and values for all other variables in S evaluated in the focus context. By choosing values of X to be singles, doubles, triples, etc. of variables in S (much as in the Group Method of Data Handling [Prager, 1988]), **$\mathscr{E}(X,y)$ can explain a substantial fraction, say 80%, of Δy.** The variables in X are then said to "explain" 80% of Δy.

The results of good comparisons include better insight into the model behavior and decision making environment.

Causation

Causation consists of identifying and quantifying causal relationships between model parameters. This may take the form of generating simplified auxiliary models that help develop insights into system behavior [Geoffrion, 1976]. For example, the classic warehouse location problem modeled using mixed integer linear programming can be reduced (using simplifying assumptions suggested by experienced operations research consultants) to several simple mathematical equations which highlight the key factors and their interrelationships in determining the optimal number and locations of the warehouses. Other researchers have suggested similar auxiliary models, also developed by experienced human experts, for simulation-based models.

Causation can be expressed mathematically as follows.

Let $Z_j = \{ \prod (x_i) \mid \text{order} [\prod (x_i)] <= p \}$

$= $ all possible terms in x_i which are of order p or less; e.g., x^p_m, $x^{p-3}_1 x^3_2$, x_2, etc.

Let

$$O = f(Z_j)$$

represent the output of some self-organizing program which relates independent variables and their cross products to a specified dependent variables, y_j, via some pth order polynomial based on the analysis of M model instances each of which is represented by a set of I x_i values and a set of J y_j values.

Let the set of key terms $\{k_l\} \subseteq \{z_j\}$ be the minimum subset of terms in O which account for at least 80% of the total variation between M model instance outputs $\{O_m\}$. Then

$$O = f\{k_l\}$$

is called the simplified auxiliary model; i.e., O is the function which relates all key terms.

The result of good causation analysis includes 1) identifying the critical factors, 2) defining their interrelationships, in simple, mathematical, deterministic terms, and 3) using these relationships to formulate and test hypotheses about the decision making environment.

Recommendation

Recommendation consists of identifying, evaluating and choosing the most appropriate, or satisfactory, solution to a given problem based on a decision maker-specified objective. In some situations, this is a function of trying several (or perhaps many) what-if cases until one is found which satisfies all objectives. In other situations, optimization techniques may be employed to provide the best solution, given the assumed parameter values and objective.

Mathematically, recommendation can be expressed as follows. Let y_j represent the set of J dependent output variables and let the solutions of each of $m \in M$ model instances be represented by

$$O_m(y_j) = f(x_i)$$

Assuming we want to maximize the utility, y_j, over all model instances, then if

$$y_{j,k} > y_{j,l} \quad \forall\ j,\ k \neq l,\ k \in M,\ l \in M$$

then $y_{j,k}$ dominates $y_{j,l}$, and the latter model instance can never be preferable to the former. If weights, w_j, can be assigned to the utility of factor y_j, then the overall utility for any instance is represented by:

$$\forall\ j \in M: \quad U_m = \Sigma_j\ w_j y_{j,m}$$

Then, the instance offering the maximum utility is

$$O_{max} = max_M\ [\ \Sigma_m\ (w_j y_{j,m})]$$

Implementation

Implementation consists of comparing the current modus operandi with the recommended solution to suggest a minimal, or least offensive, set of changes which would result in a substantial fraction (e.g., 80%) of the total improvement promised by the recommended solution. This might be viewed as a step-wise, multiple, nonlinear regression of the recommended operational changes against levels of the objective, assuming multicollinearity since some of the changes may be interrelated; i.e., the implementation of one operation change may reduce the impact of a different recommended change.

For example, the primary outcome of the study of facility location in our example was that five changes were recommended in distribution center locations (the movement of existing facilities to different cities and the opening of new facilities). The company implemented the three most urgent changes at the earliest opportunity, as well as improvements in assigning customers to distribution centers. Another distribution center addition was delayed until a later date. And a final (marginal) distribution center was dropped from further consideration. [Geoffrion, 1976, 1979]

Mathematically, implementation can be expressed as follows. Let

C = status quo or current modus operandi translated into model terms for every variable, $x_{i,c}$

= $\{x_{1,c}, x_{2,c}, \ldots x_{I,c}\}$

O_c = model solution which occurs at the status quo values for every $x_{i,c}$

O_{max} = model solution which maximizes the utility of the decision maker (see above)

U_{max} = utility generated at O_{max}

U_C = utility generated at O_C

v_i = weight assigned to each x_i which represents the difficulty or pain incurred in changing the variable x_i from its status quo value to its recommended (maximum utility) value.

Then, define a measure (X,O) to indicate the effect of the subset of variables in X on the model solution which maximizes utility relative to the solution associated with the status quo; i.e.,

$$\mathscr{E}(X,O) = O_{max} - f(Z)$$

where Z contains the values of variables in X evaluated at the status quo and values of variables in S evaluated at the maximum utility values $S \subseteq X$. Then,we find the subset of variables, x_i S which satisfies Eq. 1 (i.e., minimizes the total implementation difficulty or pain)

$$\min[\ \Sigma_{i \in S}\ v_i\ x_i\ ] \qquad \text{Eq. 1}$$

while at the same time generates a large percentage of the incremental utility, $(U_{max} - U_{sq})$, say 80%; i.e.,

$$1/\ \lambda < (U_C\ /\ U_{max}) < \lambda \quad \text{where} \quad \lambda = 80\% \qquad \text{Eq 2}$$

The subsets of variables which satisfy Eq 1 & 2 are satisfy Eq 2 and are "close" to satisfying Eq 1 can be listed in order of total implementation difficulty or pain to provide an indication of sensitivity.

Then, the recommended solution, based on the above criteria, is given by

$$O_{REC} = f(Z)$$

The output of an implementation analysis consists of an ordered set of changes with the corresponding cumulative and incremental benefits generated by each. For example, if the optimal solution of a warehouse location model recommends building five warehouses in specified locations, perhaps building three new warehouses in the most opportune locations would result in achieving 80% of the target profits with much less capital risk. Thus, the implementation analysis would consist of a list of warehouses and their locations based on the best order of their construction and the resulting cumulative and incremental projected profit for the building of each.

Post Audit

Post audit consists of comparing the presumed or forecasted impact of an implemented, model-recommended decision against what actually happens. Differences of actual versus forecasted results might be caused by several factors, including poor implementation, bad forecasts, and/or erroneous model values, assumptions or relationships. The challenge is to determine what happened, why it happened and how to compensate for such occurrences in the future to improve decision making.

For example, we might compare the actual building and transportation expenses associated the three new warehouses against those forecasted when the decision was made. Forecasts which are consistently optimistic or pessimistic can be thus be detected, and tempered with reality in future decisions. In general, the output of such an analysis is better future decision making through better forecasts, improved decision making, and more controlled decision implementation.

Mathematically, post audit can be expressed as follows. Let

X_R = set of variable values associated with the recommended solution; i.e., the implementation solution in the previous section

$= \{x_{1,R},\ x_{2,R},\ \ldots.\ x_{I,R}\}$

X_A = set of variable values associated with the actual status quo at the time the implementation decision was made; i.e., at time t'

$= \{x_{1,t}.\ x_{2,t},\ \ldots\ x_{I,t}\}$

X_{PA} = set of variable values associated with the actual status quo at the time of the post audit; i.e., at time $t'' > t'$

$= \{x_{1,t},\ x_{2,t},\ \ldots\ x_{I,t}\}$

Then there are three questions concerning discrepancies in the post audit:

1) What variables changed from time t' to t'' ?

2) What percentage of the variation between O_{REC} and O_{ACTUAL} do they explain? and

3) Is the model still valid?

The first question is answered by a straightforward comparison of actual values. The second is a special case of comparison (see discussion of Comparison above). And the third is a reevaluation of Validation (see above).

CHAPTER IV

MODEL INSIGHT GENERATOR SYSTEM (MIGS):
SYSTEM ARCHITECTURE

The architecture which we propose to address these fundamental analysis tasks consists of four major components: 1) an instance database containing a) model instances, including input parameters, relationships, decision variable values, solutions and algorithms settings, b) historical situations, including input parameters and the implemented decisions (for validity checking), and c) the current modus operandi, converted into parameters of a model instance, 2) an analysis toolkit consisting of, but not limited to, a) Kohonen networks, neural networks, abductory induction and statistics for model analysis, b) neural networks and expert systems for the analysis and interpretations of model insights, and c) case-based reasoning, set influence diagrams, data/model directories and/or ANALYZE-type sensitivity analysis information for model intelligence and analysis enhancement, 3) support modules, including an expert system-based interpreter, a neural net input controller and an automatic instance generator, 4) A user interface module. Each of these four components is described briefly below. They are depicted in Figure 3.

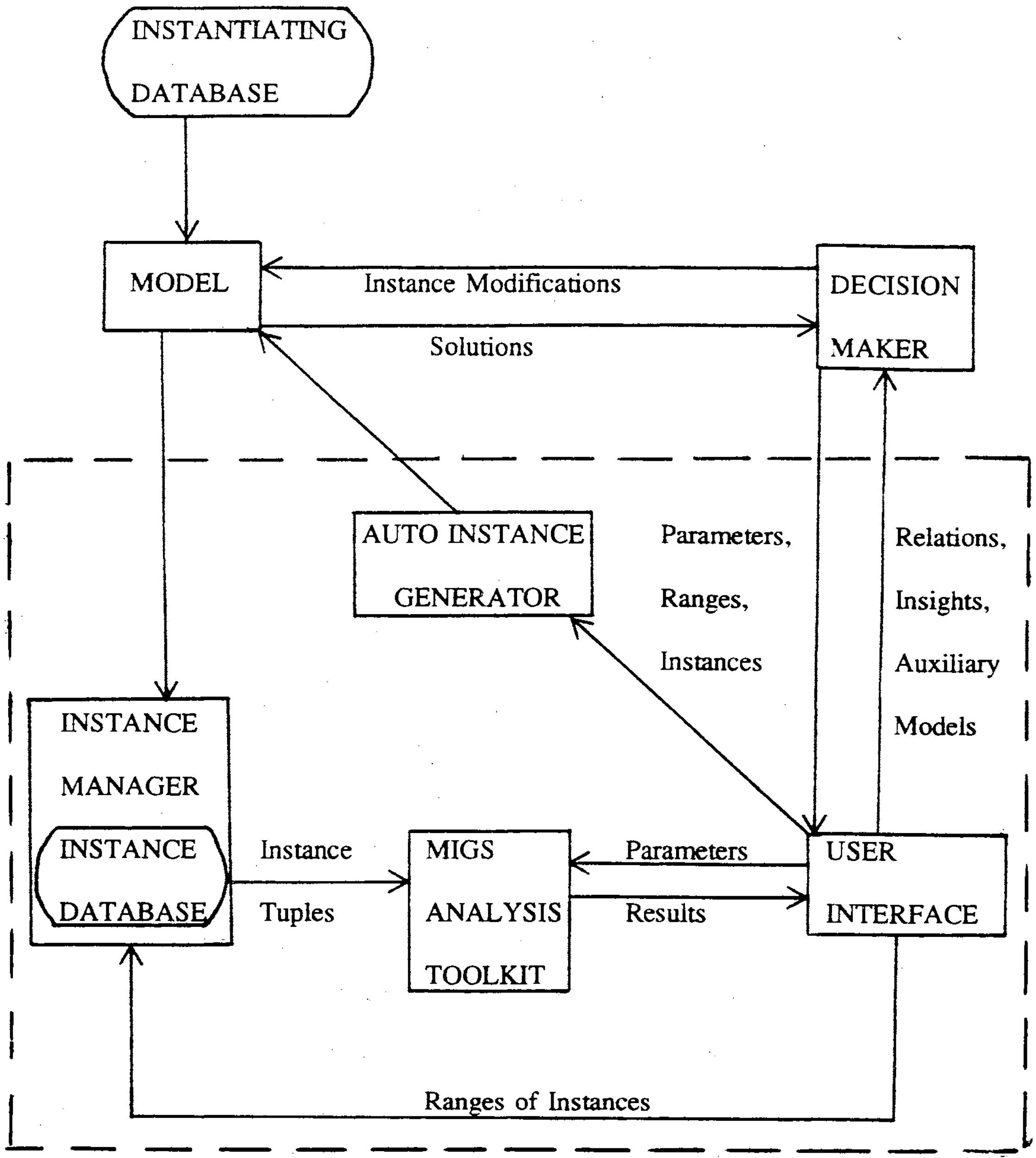

Figure 3. System Architecture for a Model Insight Generator System (MIGS)

Instance Database

The instance database can be thought of as a relational database which has one tuple for each model instance (each what-if case), for each historical situation, and for the current modus operandi. Each tuple has multiple attributes, one attribute for the identification, one for each model input parameter value, one for each (solved) decision variable value, one for the overall solution level (e.g., objective function value), and one for each solver parameter value. That is, each tuple contains all the information required to identify and recreate the solution to a specific model instance.

MIGS Technologies and Toolkit

The MIGS toolkit consists of several tools or modules which help the decision maker generate the insights into the complex situation being modeled. In general, these tools address the seven fundamental model analysis tasks described in Chapter 3. Specifically, the MIGS toolkit applies classical and "high-tech" technologies to the analysis of multiple model instances in generating insights into the basic "why" questions posed by the decision maker during modeling.

One such tool consists of a Kohonen network whose purpose is to group, or cluster, related model instances together. That is, as a decision maker traces the causation of different model characteristics or recommendations, he may generate several instances by varying one parameter, several more by varying another parameter, and several others by varying the two parameters together. To further analyze these instances, the Kohonen net clustering module groups similar instances together so that other analysis tools (see below) can derive whatever insights that exist from each group of instances. Thus, the Kohonen Net-based tool is an instance preprocessor which accepts, as input, a set of model instances and produces, as output, groupings or clusters of related instances.

A second tool in the MIGS toolkit is a multi-layered neural network model which predicts the impacts of parameter changes on the model solution based on its analysis of multiple, solved model instances and helps determine which what-if to try next. In this context, the neural net is acting as a nonparametric, nonlinear multiple regression pattern recognition device or classification technique. Neural nets are recognized as a superior (to discriminant analysis) technology for this task; i.e., neural nets are nonparametric, require no a priori knowledge of the model equation, are very fast predictors (once training is completed), and are less subject to statistical assumptions [Sharda, 1991]. However, neural nets have several disadvantages, including very slow training times, a lack of correlation coefficients and equation generation, and an obtrusive set of required training parameters which can effect the final neural net solution.

A third MIGS tool generates simplified auxiliary models [Geoffrion, 1976] which are then used to identify critical parameters and determine their interactions. This tool uses pattern recognition technologies (e.g., statistical regression, neural networks, and/or the self-organizing technology of Group Method of Data Handling or GMDH, etc.) to analyze multiple, solved, model instances and generate multiple relations in the independent variables. The multiple relations generated are then passed to an interpreter which analyzes and simplifies them to produce the simplified auxiliary model. This simplified auxiliary model is also passed through a statistical analysis package to determine its explanatory power (i.e., its coefficient of determination). The resulting simplified auxiliary model represents an automated approximation of those generated by human expert model analyzers; e.g., it compares to the auxiliary models generated by Geoffrion [1976] to the warehouse location problem based on simplifying assumptions as follows:

$$n^* = (A/3.05) * (pt/f)^{2/3}$$

$$A^* = 3.05\ (pt/f)^{-2/3}$$

where n^* = the optimum number of warehouses

A = area served by each warehouse

p = sales density

t = outbound transportation cost

f = fixed cost for every warehouse

This auxiliary model provides the insightful tradeoff between fixed costs and outbound freight, as well as the insensitivity of total costs to small departures from the optimal number of warehouses.

The GMDH technology has several advantages over other neural net technologies, including much faster training times, and output equations versus only connections weights. Its primary disadvantage is shared by neural nets in that it requires several training parameters whose settings can have severe impacts on the solution.

User Interface Module

The user interface is a loosely-coupled system which has several functions, including: 1) providing an interface between the user and the rest of the MIGS toolkit, 2) providing a driver for the MIGS tools, 3) generating, upon request, a reasonable set of new model instances for subsequent analysis and 4) querying and passing information from one MIGS tool to another and/or to the user.

The overall MIGS architecture is shown, within the dashed box, as a part of the DSS in Figure 3. In general, the inputs to MIGS consist of DSS solved model instances and various control information (e.g., ranges of instances, training parameters, etc.). The toolkit consists of neural network-, GMDH-, and ANALYZE-based tools along with various analysis enhancement technologies such as influence diagrams, case-based reasoning tools and data/model directories. MIGS output consists of clusters of similar instances, predicted results of changes in parameters, simplified auxiliary models, critical success factors, and goal seeking factors.

It should be noted that MIGS is applicable to any model-based DSS; i.e., it is equally applicable to spreadsheet models, simulation models, optimization models, statistical models, etc. This, along with the fact that it operates on multiple, solved model instances, distinguishes it from all currently available model analysis tools.

CHAPTER V

INSIGHT: PROTOTYPE IMPLEMENTATION OF A MIGS COMPONENT

Introduction

The implementation phase of this dissertation concentrates on one part of the overall MIGS architecture, specifically the relation generation tool, named INSIGHT. INSIGHT, which is one of several tools in the MIGS Analysis Toolkit (Figure 3), analyzes multiple model instances to: 1) identify the critical model parameters, and 2) generate one or more simplified auxiliary models. This tool, using self-organizing group methods of data handling (GMDH), accepts as input the tuples representing solved model instances stored in EXCEL's Scenario Manager. The tool then analyzes the instances, incrementally adding other intelligence, as needed. INSIGHT output is a set of key factors, as well as the simplified auxiliary model which explains a high proportion (say 80%) of the total variance in model output across instances.

INSIGHT System Description

The general functional characteristics of the INSIGHT system are shown in Figure 4.

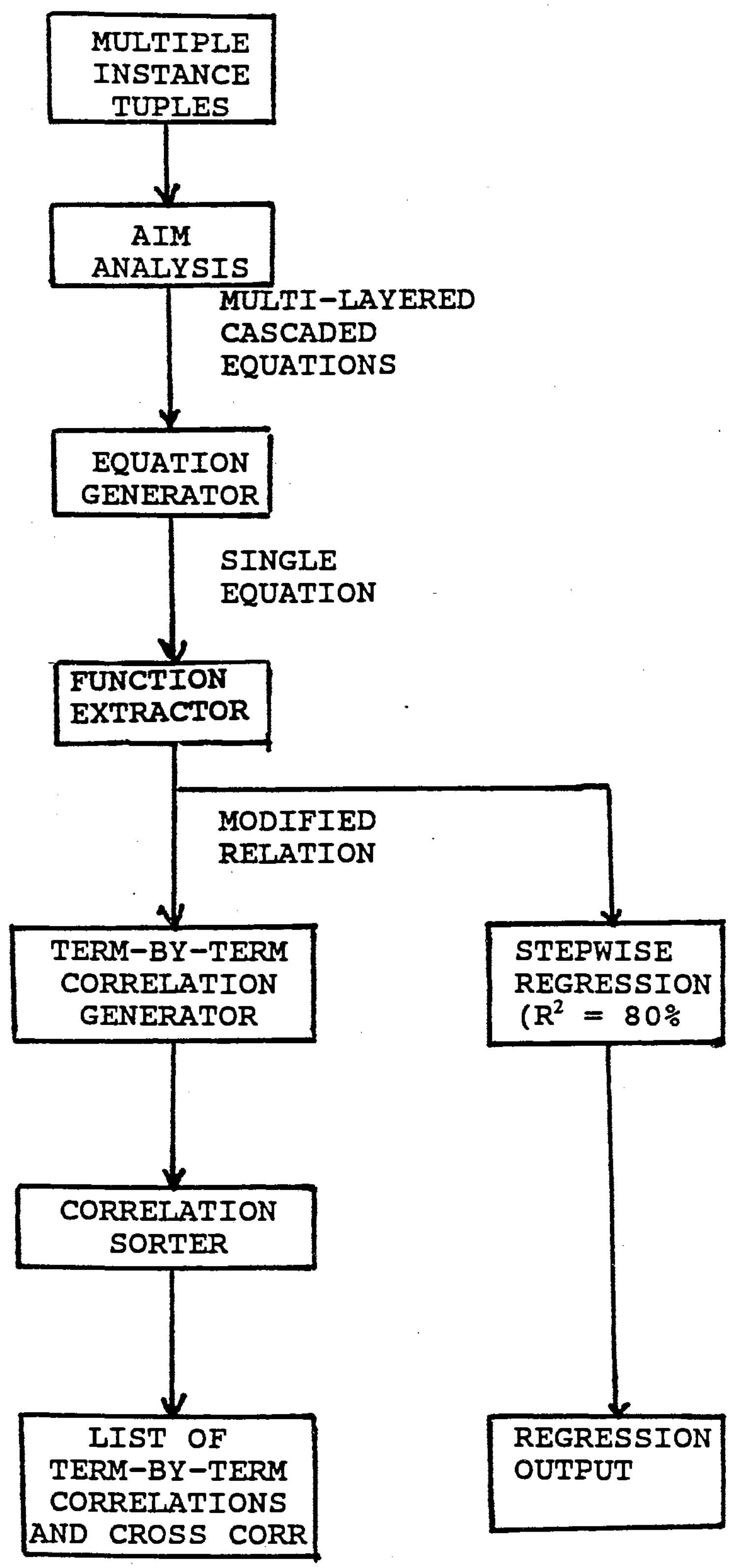

Figure 4. The INSIGHT Subsystem of a MIGS

The flowchart consists of four primary modules: 1) a model generation and storage facility, 2) a pattern/relation generator, 3) an equation generator, 4) a simplified auxiliary model generator. Each of these modules is discussed individually below.

Model Generation and Storage Facility. EXCEL is used as the model generator for the INSIGHT system, whereas EXCEL's Scenario Manager function is used to store the model instances. This package was chosen over other spreadsheet-based packages for two primary reasons: 1) it provides an add-in capability through which a mixed integer linear programming model solver (What'sBest! [Savage, 1992]) could be used to solve the facility location model in our experiment case, and 2) it provides an instance storage capability (Scenario Manager) which allows the decision maker to specify which variables and parameters to save for each model instance, and thus reduces instance processing required in other parts of the system. In addition, EXCEL is a popular spreadsheet package which is known by many decision makers in the business world.

Pattern/Relation Generator. The Pattern/Relation Generator for the INSIGHT system is AIM, a GMDH-based package which generates a multi-layered, cascading network with each layer expressed as a third-degree polynomial equation whose parameters are optimized to minimize the error between the proposed model and the training data

[AbTech, 1990]. Input to this package is a set of model instances stored as scenarios in EXCEL's Scenario Manager. The output of AIM is a data file containing the best-fit relations between the independent variables and their cross products and the dependent variable. Such a relation is generated at each layer of the cascaded network with the output of one layer treated as the input of the succeeding layer.

AIM was chosen over other GMDH-based packages and other technologies (e.g., statistical regression, neural networks, etc.) for the following reasons: 1) AIM is self-organizing and requires no a priori knowledge of, or assumptions concerning, the model form (i.e., whether the best model is linear, quadratic, etc.) as statistical regression does, and 2) AIM provides an explicit equation relating the independent variables to the instance solutions, as opposed to the matrix of interconnecting weights provided by neural networks or the correlations and influence factors of other self-organizing packages (e.g., ModelWare [TERANET, 1992]).

In the INSIGHT software, AIM is called and controlled by a keyboard macro program, Automate Anytime [Complementary Solutions Inc, 1992]. Using this keyboard macro eliminates the need for the INSIGHT user to know, and interface with, the AIM software.

Equation Generator. The equation generator is a C language program which accepts, as input, the output file from AIM (filename.NET) which contains node type indicators, node input variables and component coefficients. The equation generator then computes the single, overall equation for each primary node; i.e., each node using only original variables as inputs.

The output of this module is a list of those AIM terms having non-zero coefficients, some subset of which (hopefully) explains a high percentage (say 80%) of the total variation from average in the model output across instances.

The INSIGHT equation generator is called and controlled by a keyboard macro.

Simplified Auxiliary Model Generator. The Simplified Auxiliary Model Generator accepts, as input, the list of terms produced by the Equation Generator and produces, as output, the best simplified auxiliary model. This program uses EXCEL's correlation and regression software routines. The procedure for generating the simplified Auxiliary model consists of a two-step iterative process which finds and linearizes the most highly nonlinear term(s) and then implements a step-wise linear regression to find the best overall statistical model; i.e., the model which explains some threshold of explanatory power as determined by the coefficient of determination, say $R^2 > .80$.

EXCEL's statistical routines were chosen to implement this module since they are already integrated into EXCEL's software and they were sufficiently fast and accurate to work with.

INSIGHT System Validity

To test the validity of the INSIGHT software, we generated a facility location model for a test case, called the Ajax case, described in Appendix A. Geoffrion [1976] used a similar model to illustrate the development of insight-generating simplified auxiliary models. The general facility location model, formulated using mixed integer linear programming, is as follows:

$$\text{Min} \ \Sigma_i \ \Sigma_j \ t_{ij} * x_{ij} + \Sigma_i \ f_i * y_i$$

$$\text{S.T.} \ \Sigma_i \ x_{ij} = p_j \text{ for every } j$$

$$\Sigma_j \ x_{ij} - M * y_i \leq 0 \text{ for every } i$$

$$y_i = 0,1 \text{ for every } i$$

$$x_{ij} \geq 0 \text{ for every } i, \text{ for every } j$$

To this formulation, Geoffrion [1976] added seven simplifying assumptions, (namely:

1. Demand is uniformly distributed on the plane with a density of $p(CWT/\text{mi}^2)$.

2. All warehouses are identical and arbitrarily relocatable.

3. The supply cost for each warehouse is s ($/*CWT*) regardless of its location.

4. The fixed cost of each warehouse is f ($).

5. The variable throughput cost of each warehouse is v ($/*CWT*).

6. The outbound freight rate for each warehouse is t (\$/CWT-mi).

7. There are no throughput limits for the warehouse.

He then used human expertise and mathematical manipulation to generate the following insight-generating simplified auxiliary model for the optimal number of warehouses, n^*, for an area having A square miles of area:

$$n^* = A/3.05 * (p * t / f)^{2/3}$$

Our test case incorporates approximately the same set of four assumptions as Geoffrion used, within the limitation of a real-world setting. However, in our test case demand, instead of being uniformly distributed throughout the plane, is evenly distributed among the thirteen potential cities. Thus, the case represents lumpy demand located in thirteen fairly centrally located (but not exactly equidistant) cities, where distances between cities are actual mileages. Our test case depicts the cities in Central Texas, an arbitrary locale chosen simply because a map showing city-to-city driving distances was handy at the time.

The Ajax facility location model was formulated as a 13 x 13 city mixed integer linear programming model using the What's Best [Savage, 1992]! software package to solve specific instances. As a test of the INSIGHT tool, we generated and solved a set of 24 model instances, each with a different value of one or more of the following variables: total demand, p, warehouse-to-customer transportation rate, t, and/or warehouse fixed costs, f. Required solution time was approximately two hours for each model instance on a 80386SX microcomputer with a math coprocessor, the machine on which we did our testing.

Any instance which resulted in an optimal number of warehouses greater than one and less than thirteen was accepted as one of the instances to be analyzed by INSIGHT. In addition, we ensured that there were at least two model instances which depicted different values for each of the three model variables mentioned above. Both of these restrictions, concerning the selection of model instances to be used in the analysis, could easily be implemented in an expert system module in MIGS.

AIM cannot generate <u>simple</u> models for common terms such as 1/x, sqrt(x) and cos(x); e.g., AIM generates an eighteenth degree polynomial to approximate cos(x), patently unsuitable for our simplified auxiliary model. To address this potential problem, we included in the AIM preprocessor a routine to automatically add 1/x and sqrt(x) for each of the independent input variables specified in Scenario Manager since these are common terms which might be potential components of any simplified auxiliary model; a future enhancement to INSIGHT will include the capability for the user to specify such terms at his discretion.

Based on these 24 model instances, the INSIGHT tool generated the following simplified auxiliary model:

$$n^{*} = 700 * (p * t / f)$$

using the same three key variables as used in the Geoffrion model in an even more simplified form. This single term explained 92% of the total variation from average of the optimal number of warehouses; i.e., $R^2 = .92$. A side-by-side comparison of Geoffrion's results and those of INSIGHT is shown in Table 1.

Also shown in Table 1 is an indication of the insensitivity of the INSIGHT results to the number of instances analyzed. In this model, at least, the INSIGHT results degrade gracefully (with respect to R^2) down to the 10 - 15 instance range. This indicates that only a modest number of intelligently selected instances are required to generate the insightful results shown. Actual results in other models would depend on the model, modeling paradigm, specific instances, complexity of relationship, etc.

Thus, the INSIGHT tool, using concepts of artificial intelligence as applied to the analysis of multiple model instances, was able to duplicate the insight-generating simplified auxiliary model produced by Geoffrion without using human expertise or mathematical manipulations based on simplifying assumptions. This provides a basic validity test of the concepts of MIGS and the concepts and implementation of the MIGS INSIGHT tool.

TABLE 1

TEST PROBLEM RESULTS AND COMPARISON

GEOFFRION	INSIGHT
METHOD	METHOD
HUMAN EXPERTISE	ARTIFICIAL INTELLIGENCE
MATHEMATICAL KNOWLEDGE	STATISTICAL SOFTWARE
MATHEMATICAL SIMPLIFICATION	MORE REALISTIC ASSUMPTNS
MATHEMATICAL MANIPULATION	MULTIPLE INSTANCES
RELATION GENERATED	RELATION GENERATED
$n^* \approx (d * t / f)^{2/3}$	$n^* \approx d * t / f$

RESULTS FROM INSIGHT BY NUMBER OF INSTANCES

# INSTANCES	R^2 FOR RELATION
15	.88
24	.92
48	.94

INSIGHT Simple Experiment

An experiment was conducted to test the face validity of the INSIGHT concepts; i.e., to test directionally, whether the INSIGHT tool aided decision makers in identifying key factors and generating relations between those key factors and the model solutions.

In testing the face validity of a system such as INSIGHT, it must be stated that research of computerized decision analysis tool is currently exploratory in nature [Aldag and Power, 1986; Benbasat and Nault, 1990]. Further, it has been hypothesized and found that decision makers will employ more effortful strategies, which generally lead to higher decision quality, when the DSS reduces the effort needed to use them [Todd and Benbasat, 1990,1991; Johnson and Payne, 1985; Jarvenpaa, 1989; Payne, 1982; Einhorn and Hogarth, 1978].

The primary hypotheses which we wanted to test with respect to INSIGHT were as follows:

Hypothesis #1: Decision makers using the INSIGHT tool will generate a more valid set of key factors than those not using the system.

Hypothesis #2: Decision makers using the INSIGHT tool will generate more accurate relations between key factors and the model solution than those not using the system.

We generated a set of model instances by varying the three parameters over their appropriate ranges, making sure that the final set of instances used as input to INSIGHT contained a representative sample of the variable values. In addition, we limited the instances to those in which the optimal number of warehouses, n*, was strictly greater than 1 and strictly less than 13; i.e., the minimum and maximum possible number of warehouses.

To determine the effects of variations in the number, selection and order of instances, we ran INSIGHT using several different sets of instances and analyzed the results. Specifically, we ran INSIGHT using 15, 24 and 48 instances and found no change in the key variables or key relationship; however, the coefficient of determination did change, from .88 to .92 to .94, respectively. We found that no changes occurred when the selection of instances was varied, except for minor changes in R-squared. And, no changes occurred when the order of instances was varied.

The validation of the value of MIGS and the INSIGHT tool, when used by MS/OR experts, was missed. However, the value of MIGS and INSIGHT was at least partially validated by recreating Geoffrion's [1976a] results artificially without relying on human expertise and mathematically manipulations.

Participants in the study consisted of 65 undergraduate business students enrolled in a management science course. This course covers such normative decision analysis topics as decision theory, simulation, model development, linear programming, goal programming, sensitivity analysis, etc. At the time of the study, the students had a basic understanding of PC use, EXCEL spreadsheet software, what-if analysis, and business decision theory. As a result of course prerequisites, the students were, as a group, homogeneous with respect to modeling experience, computer skills, etc. The case study which forms the basis of our experimental design was counted as part of the students' semester grade, thus providing sufficient student motivation for the experiment.

The subjects analyzed a pre-prepared set of 24 solved model instances depicting a classic facility location problem (see Appendix A). This task was chosen as representative of complex decision problems which occur in industry and which have been shown to significantly improve profits and/or decrease expenses [e.g., Geoffrion, 1974; Klingman et al., 1987].

The multiple solved model instances are required in the evaluation of various deterministic model assumptions. That is, multiple instances are required when the deterministic assumptions in a normative model are relaxed and replaced by stochastic assumptions, and the model is re-solved to determine the robustness of a solutions for real world implementation. For example, the robustness of a model solution assuming deterministic freight rates might be tested in light of partial versus full truckload commercial freight rates, inflated freight rates (caused by normal economic inflation or by an oil shortage) or deflated rates (caused by switching from truck to rail freight or from partial truckload quantities to full truckload rates). When several model parameters are tested, the determination of the "best" solution for a probable range of multiple parameters becomes difficult. Furthermore, when such stochastic parameters interact to significantly affect the solution, the determination of the "best" solution quickly becomes an exercise in combinatorics. In these complex situations, the INSIGHT tool becomes highly advantageous.

The facility location model was presented as a pre-formulated EXCEL spreadsheet model which used mixed integer programming as a solution technique, but returned resulting solutions to the spreadsheet.

Subjects performed the required analysis during a regularly-scheduled class period in a PC laboratory or during general laboratory periods. Each subject was required to work alone on the experiment. Subjects were asked not to discuss the case with other students.

Each subject was randomly assigned to one of two groups, the first group was given access to INSIGHT and the second group was denied such access. Each subject then received 1) a copy of the case, 2) the questions, and 3) an explanation of the EXCEL decision model. Each subject was given access to, and an explanation of, the solved model instances and the computer-assisted tools (i.e., graphs for all subjects and the INSIGHT tool for the first group) to be used to analyze them.

A time constraint of sixty minutes was suggested to the subjects; this time limit was a function of the length of regularly scheduled class periods for those who performed the analysis during class, or the time between classes for those who performed the analysis during a lab period between two classes. This time included both the analysis of the model instances and the answering of the insight questions.

The experiment investigated the effect of a single independent variable on effectiveness at two levels; i.e., the presence or absence of the INSIGHT tool. Subjects were randomly assigned to one of two groups, one group aided by the INSIGHT tool and the other group unaided. Both aided and unaided subjects were provided with the same set of solved model instances. These instances formed the input for the INSIGHT tool. All decision makers were also provided with pre-generated scatter graphs of each independent variable versus the dependent variable, a lower triangular correlation matrix, a linear regression output, and a summary of the 24 scenarios in spreadsheet format.

The dependent variable in this study was the subjects' ability to correctly identify the key factors and the key relations, as determined by their test scores on the eleven-item questionnaire administered as part of the experiment. This questionnaire (see Exhibits A and B) asked the subject to identify the key variables and determine key relationships between these variables and the model solution. The questionnaire score was calculated by totaling the number of correct answers to the appropriate questions.

The questions in the questionnaire are based on the most important insights to be gleaned from this model. Specifically, these insights include answers to the following questions [Geoffrion, 1975, 1976a,b]:

How do the major cost categories change as the number of distribution centers change?

How sensitive is the most appropriate system design to the cost and environmental assumptions in which there is significant uncertainty or likelihood of change?

What is the tradeoff between fixed costs and outbound freight costs?

Results. The questionnaire used in the experiment is divided into three parts. Questions #1 and 2 address the subject's ability to determine the key factors in the case. Questions #3 - 10 address the subject's understanding of the key relationships between the key factors (independent variables) and the best number of warehouses (dependent variable). Question #11 addresses the subject's confidence in correctly answering the previous ten questions. The first part of the questionnaire is used to test Hypothesis #1, while the second part of the questionnaire is used to test Hypothesis #2.

Since the sample sizes are large (i.e., greater than 30) for both samples, a large sample hypothesis test for the difference between two population means is used, incorporating the z-score as the test statistic. Large sample sizes provide for testing hypotheses without having to assume normally distributed populations or equality of population variances. However, a one-way ANOVA (SAS) was also run which used the same data and produced similar results.

Test scores were computed for every subject by totaling the number of correct answers for the applicable part of the questionnaire. For scoring purposes, Question #1 was treated as four questions, one for indicating the relative importance of each of the four factors. Thus, the maximum score for Question #1 was +4, and together with Question #2, provided a maximum score of +5 for Hypothesis #1. Questions 3 - 10 provided a combined maximum score of +8 for testing Hypothesis #2. The resulting scores were then averaged for each sample. Experimental results are summarized in Table 2 below.

As indicated in Table 2, the INSIGHT tool had no significant effect on the subjects' ability to determine which factors were key factors ($p > .125$), although directionally INSIGHT subjects did perform slightly better. A vast majority of subjects in both groups indicated (incorrectly) that plant-to-warehouse transportation cost was a key factor, even though the case clearly states that "due to the company's current zone transportation agreement,

TABLE 2

SUMMARY OF EXPERIMENTAL RESULTS

	Aided by INSIGHT	Unaided by INSIGHT	Statistical Significance
Key Factor Determination			
Sample Size	$n_1 = 33$	$n_2 = 32$	no; $p > 0.10$
Average Score	$\bar{x}_1 = 2.1212$	$\bar{x}_2 = 1.8438$	
Standard Dev.	$s_1 = 0.9924$	$s_2 = 0.9873$	
Key Relation Determination			
Sample Size	$n_1 = 33$	$n_2 = 32$	yes; $p < 0.025$
Average Score	$\bar{x}_1 = 3.9393$	$\bar{x}_2 = 2.9375$	
Standard Dev.	$s_1 = 2.0907$	$s_1 = 1.2936$	

the transportation costs from Ajax's California plant to anycity in Central Texas is the same." In addition, the plant-to-warehouse transportation cost was omitted from the list of key factors in the INSIGHT output.

Also as shown in Table 2, the INSIGHT tool did have a significant influence on the subjects' ability to determine and understand the key relations between key factors. Specifically, subjects with the INSIGHT aid scored significantly better ($p < 0.025$) than those without the aid.

Limitations. One possible rationale for not finding a significant improvement in key factor determination (Hypothesis #1) is that the INSIGHT output variable, t, did not distinguish plant-to-warehouse transportation from warehouse-to-customer transportation. This may have caused confusion in determining the key transportation factor. Further, since all transportation costs are normally important in business decisions, to exclude the plant-to-warehouse transportation as a key factor may have appeared counterintuitive, especially if the case were quickly scanned instead of carefully read.

On the other hand, many subjects' answers to Questions #1 and 2 were inconsistent; i.e., the key factors identified as having importance ratings of 4 and 5 in Question #1 were not always included as the key factors in Question #2. For example, Nearly half (30 of 65) of all subjects selected answer d) in Question #2 indicating all four factors were key factors, and yet 18 of the 30 did not rate all four factors important in Question #1; i.e., they rated at least one factor less than 3 on the importance rating of 1 to 5.

Although INSIGHT did help the subject determine the key relationship between the key factors (Hypothesis #2), it was surprising that the improvement was not greater. That is, since the INSIGHT output specifies the key relationship in mathematical form ($n^* = 700 * d * t / f$), it seems reasonable that the subjects' average scores should have been closer to 8.0 than to the Table 2 average of 3.9. In search of possible explanation(s) for this difference between expected and actual results, we found several rationale which should impact further developments of the INSIGHT design, as well as the experimental design.

First, upon detailed analysis of the data, we found that almost 40% of the subjects (13 of 33) who had access to the INSIGHT tool never cited its use as an aid to answering a question. The INSIGHT output potentially provides an effective method of both increasing the quality of the key relation (when compared to the linear regression, correlation output and scatter diagrams) and reducing the subject's cognitive effort in answering question (when compared to the analysis of the scenario summary, linear regression and correlation). These two criteria, decision quality and cognitive effort, are the most cited rationale for the use of computer-assisted decision making aides [Todd and Benbasat, 1991; Keen and Scott Morton, 1978; Russo and

Dosher, 1983; Johnson and Payne, 1985; Jarvenpaa, 1989]. This indicates that INSIGHT was perhaps not sufficiently "sold" to the subjects as an appropriate effort saving and analysis enhancing tool. Alternatively, INSIGHT might have been perceived by the subjects as complex to run or understand, as opposed to the manual tools which required no additional steps. Or subjects might have viewed INSIGHT as additional information that must be analyzed in an already constricted time period.

Second, upon analyzing the aids which were used by subjects not having access to INSIGHT, it was found that scatter diagrams (graphical charts), correlation coefficients (the lower triangular matrix) and the scenario summary (in spreadsheet form) were the most frequently used aids in answering questions. In contrast, the INSIGHT output was presented as a mathematical representation of the key relationship. This indicates that the INSIGHT output should be integrated into multiple analysis views; i.e., incorporated into the scenario summary spreadsheet (as an additional column replacing, or appended to the right of, the column for n^*), added to the correlation matrix (as a linearized term) and used to generate scatter diagrams (either as a whole plotted against the dependent variable n^*, or be holding all but one component constant and plotting changes in the one independent variable against changes in the dependent variable). This would correspond to the multiple model views used during model formulation in current executable modeling language research [Krishnan, 1989; Steiger et al., 1993].

Third, the relatively high variance in scores for INSIGHT-aided subjects indicates that, even though INSIGHT-aided subjects scored higher on test questions, at least some subjects had significant difficulty answering questions correctly even while using the INSIGHT output. For example, one subject incorrectly answered questions #7, 8, 9 and 10 while citing the INSIGHT relation as his method of arriving at the answers. Another subject incorrectly answered questions #7, 8 and 9 under the same circumstances, while a third such subject missed questions #6, 8 and 9 while using INSIGHT. It would be interesting to try to find out why this happened by examining their reasoning; unfortunately, this information is unavailable, but could be included in future experimental designs with INSIGHT. However, these inconsistencies indicate that some subjects perhaps bordered on mathematical illiteracy (i.e., were incapable of applying concepts studied in high school algebra) and were unable to take advantage of the mathematical form of INSIGHT. This provides further impetus for integrating multiple views of INSIGHT output into the MIGS architecture. It also implies an additional research area for implementing a natural language interface to restate the INSIGHT output in natural language for those users who are uncomfortable using mathematical relations.

The purpose of this experimental design was to test the effect of the INSIGHT aid on the selection of key factors (Hypotheses #1) and the determination and understanding of the key relations of those key factors (Hypothesis #2). It was not designed to prove or test the development of the Gestaltists' concept of insight. However, the INSIGHT tool could be used in such an experiment; e.g., by using longitudinal studies to test the habitual characteristic of insight and the generalization of insightful solutions.

It should be noted that this experimental design limits the number and level of insights which could be generated by the subjects. Specifically, the management science literature suggests that significant insights are often developed during model specification and building. In addition, our experimental design limits insight developed during the analysis of a logical sequence of instances which explores unusual and/or unexpected occurrences in model behavior. Both of these limitations are by design. That is, the time limitations of the subjects prevent them from building a complex model, running instances and analyzing the results. Further, the variances in such models when developed by each subject would introduce a critical, uncontrolled variable into the experiment. In addition, the individual development of a set of instances would likewise introduce an uncontrolled variable; i.e., different sets of model instances used as input to the INSIGHT tool might produce different results and lead to different key factors and their interrelationships.

One might also expect there to be an additional independent variable in this experiment, namely the presence or absence of a pre-generated set of solved model instances. However, an experimental design based on instances generated by each decision maker was rejected since each decision maker would, in all probability, generate a different number, set and sequence of instances, and each difference could cause a difference in the INSIGHT output and the subsequent questionnaire score. Given an experimental model for explaining differences in the dependent variable, y (the questionnaire score),

$$y = b_1 + b_2x_1 + b_3x_2 + \ldots$$

each different number, set and sequence of instances would have to be included as a different independent variable, x_i, resulting in a sample size, in all probability, equalling the number of independent variables. Such a model would prove very limited in statistical and predictive power.

Another limitation of the study comes from the use of students, especially undergraduate business students, as subjects. Clearly, using subjects with significant industry experience in the application of complex models would strengthen the conclusions of this study. Undergraduate students may lack both the general problem analysis/solving skills and the domain-specific knowledge required for the test problem used in this experiment.

Another limitation was the lack of a test to validate, at least to a limited extent, the value of the MIGS philosophy and the INSIGHT tool when used by MS/OR experts. Our original plan, which included such a test, was canceled due to excessive model solution times (> 2 hours/model instance). This problem could be addressed by selecting a different test problem (perhaps an LP model versus an MIP model) which could be solved in seconds versus hours, and then instituting an experiment using industry experts.

INSIGHT - A Sample Session

The INSIGHT software has been implemented as an add-in command toward the bottom of the FORMULA menu. To execute the INSIGHT software, the user must first generate an appropriate set of instances, storing them in the Scenario Manager after specifying the appropriate set of independent and dependent variables.

After solving and storing the model instances in Scenario Manager, the user simply selects the INSIGHT command in the FORMULA menu (Figure 5). He is then presented a pop-up dialog box requesting him to specify the names of both independent and dependent variables in the same order they are specified within Scenario Manager (Figure 6). Upon completion of this task, the user clicks the OK button.

Next, he is presented with another pop-up dialog box requesting him to specify the dependent variable and, if

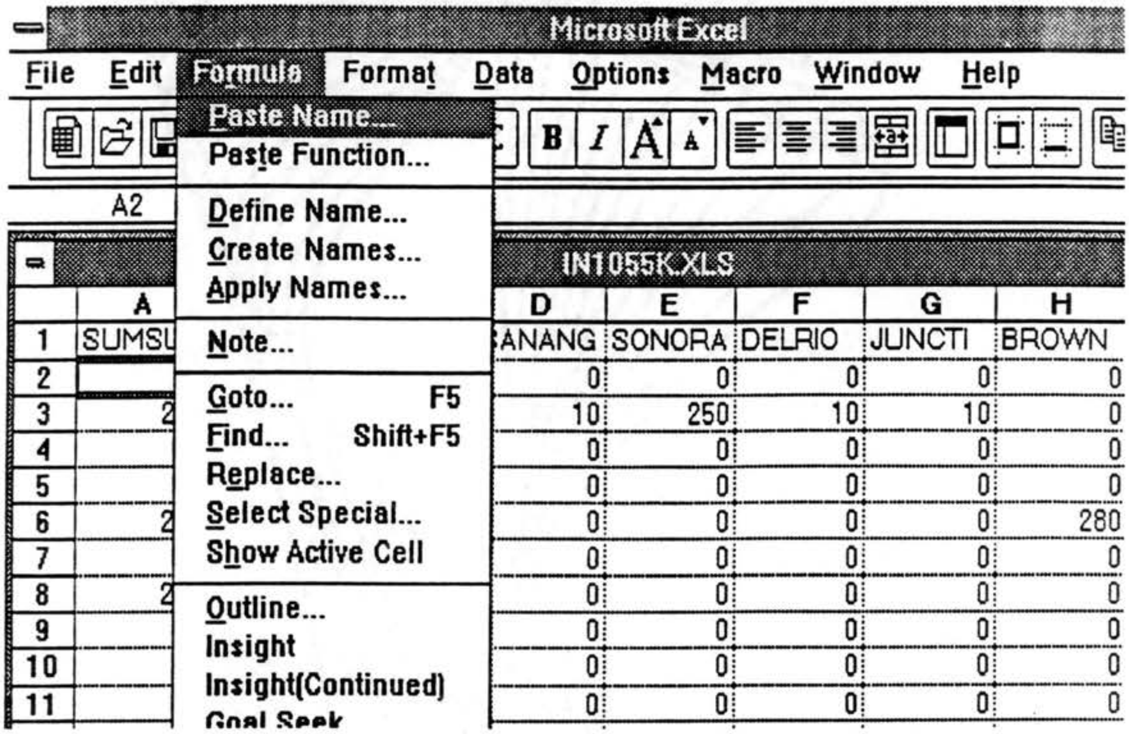

Figure 5. FORMULA Menu Showing INSIGHT Command

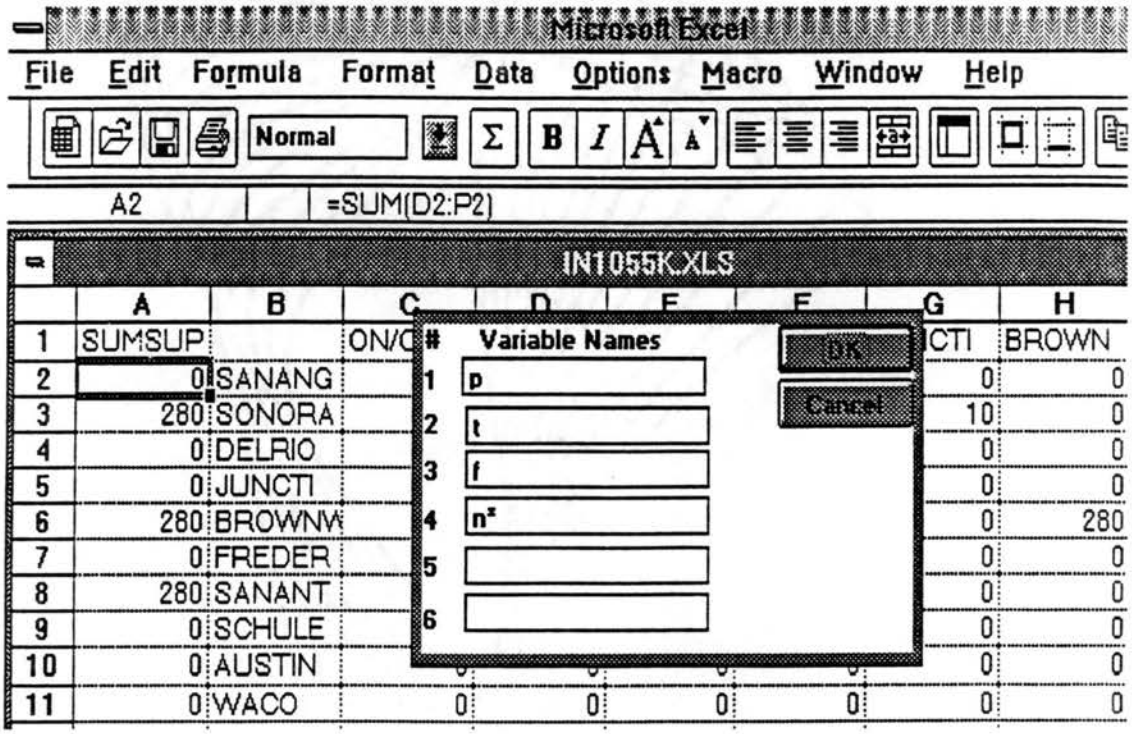

Figure 6. INSIGHT Dialog Box for Specifying Variable Names

applicable, a range of possible values for that variable Figure 7). Again, he clicks on the OK button when finished. This initiates the INSIGHT data processing and eventually (after a couple of minutes on an 80386SX when running the Ajax case) results in a third pop-up dialog box telling the user to exit EXCEL and PROGRAM MANAGER, and key in the word 'PROCESS' (without the quotes) at the DOS prompt. This initiates the keyboard macro which calls AIM, processes the data, calls the equation generator and, eventually returns microprocessor control to the user.

The user must then key in the WIN command to call the Windows PROGRAM MANAGER, and double click on the EXCEL icon. Upon re-entering EXCEL the user must then select the INSIGHT(CONTINUED) command in the FORMULA menu (see Figure 5) to continue processing. After a short time, the INSIGHT software displays a final dialog box, giving the user a list of key factors and one or more key relations relating those key factors (Figure 8). After perusing the information in this dialog box, the user may click on the OK button and continue his EXCEL processing.

Figure 9 provides a summary of user actions required to run INSIGHT. In addition, Figure 10 provides a summary of program and file linkages associated with running INSIGHT.

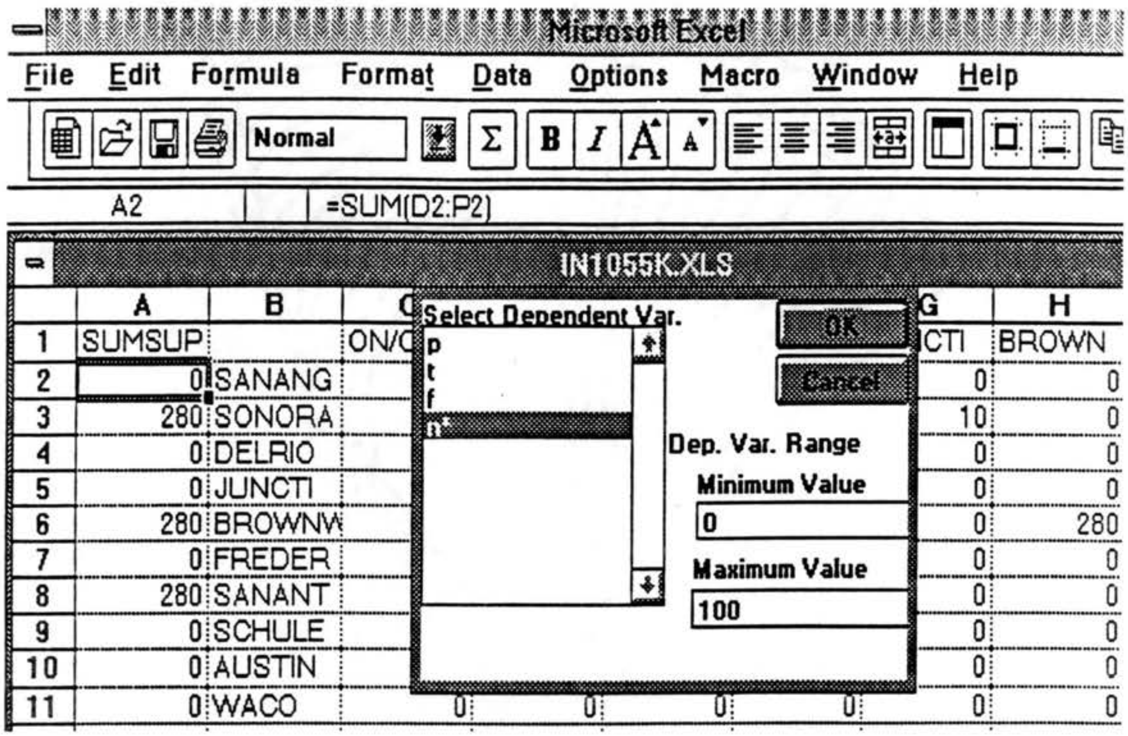

Figure 7. INSIGHT Dialog Box for Specifying Dependent Variable and Its Range

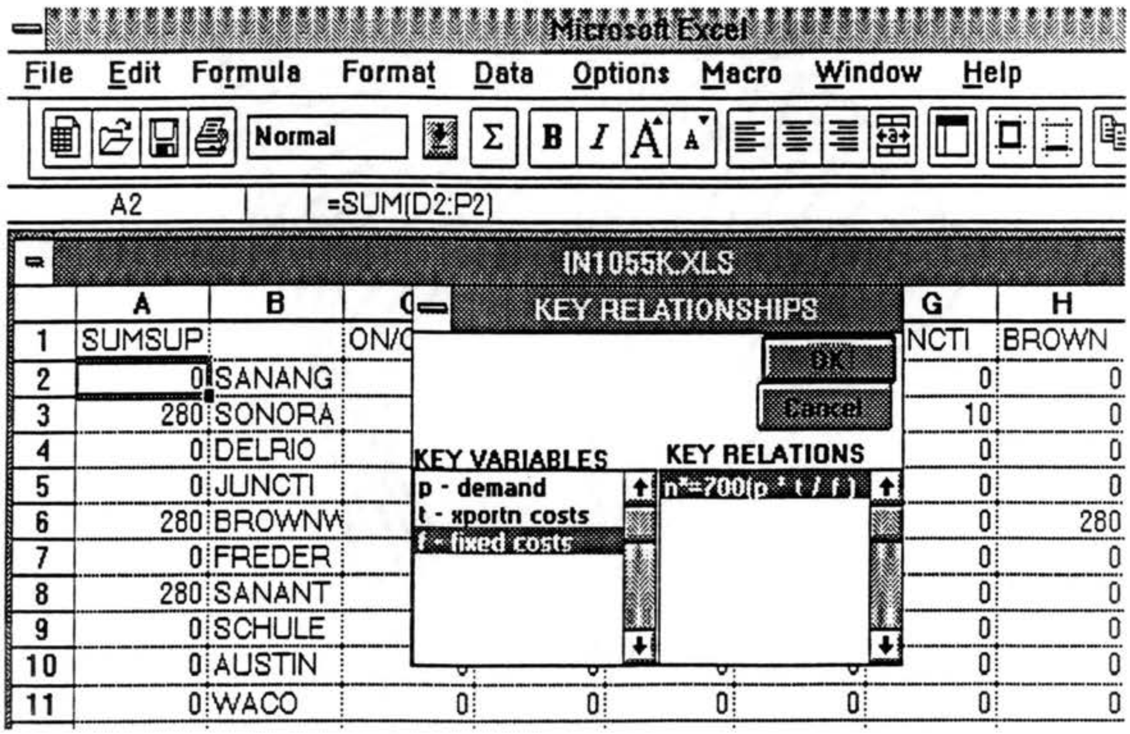

Figure 8. INSIGHT Display of Key Variables and Key Relations

Generate Excel Model Instances

↓

Select INSIGHT Command

in FORMULA Menu

↓

Specify Variables, Identify Dependent

Variable and Value Ranges

↓

Exit Excel

↓

Execute PROCESS

Keyboard Macro

↓

Re-enter Excel

↓

Select INSIGHT(CONTINUED)

Command in the FORMULA Menu

↓

Peruse KEY Variables and Relations

Generated by INSIGHT

Figure 9. Summary Flowchart of User Actions Required to Run INSIGHT

Scenario Manager

Headers for data matrix downloaded to flat file (ZZZAIMIN.HDR)

User specified variable names and ranges

Scenario Values

Data matrix created via Excel Macros & downloaded (ZZZAIMIN.RAW)

AIM analysis; output file with element identifiers, polynomial coefficients (ZZZAIMIN.NET)

Polynomial terms generated by Equation Generator (C program) and uploaded into spreadsheet.

Regression run via macro. Key relations formed by step-wise linear regression via macro.

Produce output screen via macro. End

Figure 10. Flowchart of INSIGHT Linkages of Programs and Files

CHAPTER VI

FUTURE DIRECTIONS AND RESEARCH AREAS

There are several areas for future research suggested by the MIGS architecture and INSIGHT prototype. First we need to explore other pattern recognition technologies which would provide potential key relations. Such technologies might include neural networks using existing or unique special purpose architectures, self-organizing nonlinear regression techniques based on something other than third degree polynomial representations, and/or other self-organizing polynomial techniques which employ quality measures more conducive to filtering out unimportant terms and keeping only key term.

A second research direction might include investigations of potential applications of the INSIGHT methodology, not only in generating key factors and relations for analysis in a decision support system, but also to suggesting relation-based heuristics for solving "tough" problems in management science literature, such as machine layout problems, specific traveling salesman problems, etc. Such problems might be solved and explained more readily and/or efficiently by the simplified auxiliary models produced by the INSIGHT methodology.

A third research direction might investigate the use of neural networks or Kohonan networks to recognize the polynomial representations of the more familiar mathematical functions such as 1/x, sqrt(x), sin(x), etc. This would eliminate the requirement for the user or analyst to know and input these functions a priori as key functions. It would also allow significant simplification of key relations if such key relations contained such terms; i.e., it would allow the reduction of an 18th degree polynomial representation of a cos(x) function produced by an polynomial-based technique to a simple term.

Finally, and most obviously, we need to further refine the mathematical representation of the analysis functions in order to search for additional commonalities in these functions and potentially discover alternative methods of implementing them in an artificial intelligence environment. This, of course, goes hand-in-hand with implementing the rest of the MIGS architecture.

CHAPTER VII

SUMMARY AND CONCLUSIONS

After building and validating a decision support model, the decision maker frequently solves (often many times) a slightly different version of the model. That is, by changing various inputs and parameters and re-running different model instances, the decision maker develops insight(s) into the workings and tradeoffs of the complex system represented by the model. While exploring several aspects of the model, he may develop a (large) set of model instances, some of which are related to one line of exploration (e.g., the cost-benefit tradeoff of additional capital investment), and some related to another line of exploration (e.g., the addition of new product lines complementary to an existing line). As the number of instances grows, the need for a method of storing, accessing and analyzing the instances also grows.

The purpose of this dissertation is to explore and develop the use of current MS/MIS/CS technologies in enhancing the decision maker's analysis of multiple, related model instances in the model management system (MMS) environment of a decision support system (DSS). Such analysis may be in the form of grouping or clustering model instances which are related to the same alternative(s) and/or recognizing and exploring underlying patterns of interaction between specified decision variables or parameters. This analysis may be viewed as developing insight from a database of model instances for an MMS. The objective of this research is to propose an architecture of a model insight generator system, to build a prototype system which implements an appropriate subset of this system, and to test the effectiveness of the prototype system, using business students as subjects.

After building the INSIGHT prototype, we tested the validity of the concepts and design in two ways. First, we recreated an insightful model studied by Geoffrion. The INSIGHT tool, using concepts of artificial intelligence as applied to the analysis of multiple model instances, was able to duplicate the insight-generating simplified auxiliary model produced by Geoffrion without using human expertise or mathematical manipulations based on simplifying assumptions.

Second, we designed and performed an experiment using student subjects to test two hypotheses: 1) that decision makers using the INSIGHT tool would generate a more valid set of key factors than those not using the system and 2) that decision makers using the INSIGHT tool would generate more accurate relations between key factors and the model solution than those not using the systems. We found that the experimental results supported the latter hypothesis, but did not support the former.

REFERENCES

AbTech (1990). Abductory Inductive Models -- User's Manual. Ab-Tech, Inc., Charlottesville, VA

Aldag, R. J. and D. J. Power (1986). An Empirical Assessment of Computer-Assisted Decision Analysis. Decision Sciences, 17: (Fall,1986), 572-588.

Ashton, R. H. and S. S. Kramer (1980). Students As Surrogates In Behavioral Accounting Research: Some Evidence. Journal of Accounting Research,(Spring,1980), 1-15.

Baldwin, D. (1990). The Development and Architecture of DOVE: A Multiple Viewpoint DSS, Proceedings of the 1990 ISDSS Conference,(September 11-14, 1990) Austin,TX, 147-167.

Bell, P. C., D. C. Parker and P. Kirkpatrick (1984). Visual Interactive Problem Solving--A New Look at Management Problems. Business Quarterly. (Spring 1984),24-38.

Benbasat, I. and B. Nault (1990). An Evaluation of Empirical Research in Managerial Support Systems. Decision Support Systems 6, 203-226.

Bisschop, J. and A. Meeraus (1982). On the Development of a General Algebraic Modeling System in a Strategic Planning Environment.Mathematical Programming Study #20. North-Holland Publishing Company, NY. 1-29.

Bodily, S. E. (1985). Modern Decision Making -- A Guide to Modeling with Decision Support Systems. McGraw-Hill Book Company, N.Y.

Bodily, S.E. and L. Weather Ford (1989). @Risk:Lotus Add-in Software. OR/MS Today 16:1, 22-23.

Bonner & Moore Management Science (1988). GAMMA 2000 User's Manual. Version 1.0. Houston, TX.

Brennan, J. J. and J. J. Elam (1986). Understanding and Validating Results in Model-Based Decision Support Systems. Decision Support System. 2, 49-54.

Bos, H. C. (1965). Spatial Dispersion of Economic Activity. Rotterdam University Press, Rotterdam.

Broadbent, D. E. (1961). Behavior. Basic Books, Inc., New York.

Brooke, A., D. Kendrick and A. Meeraus (1988). GAMS A User's Guide. The Scientific Press. Redwood, CA.

Chesapeake Decision Sciences (1988). MIMI/LP User Manual. New Providence, NJ.

Chinneck, J. W. (1990).Formulating Processing Network Models: Viability Theory Naval Research Logistics.37, 245-261.

Clemen, R. T. (1990). Making Hard Decisions: An Introduction to Decision Analysis. PWS-Kent, NY 34-49.

Complementary Solutions Inc (1992). Automate Anytime User Guide. Atlanta, GA.

Cunningham, K. and L. Schrage (1990). LINGO: Language for Interactive General Optimization. LINDO Systems, Inc., Chicago, IL.

Dannenbring, D. G. and M. K. Starr (1981). Management Science: An Introduction. McGraw-Hill Book Company, NY.

Dayhoff, F. (1989). Neural Network Architectures. Van Nostrand Reinhold, NY.

de Groot, A. D. (1965). Thought and Choice in Chess. The Hague, Mouton.

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

Einhorn, H. and R. Hogarth (1981). Behavioral Decision Theory: Processes of Judgement and Choice. Annual Review of Psychology. 32, 53-88.

Elam, J. J. and B. Konsynski (1987). Using Artificial Intelligence Techniques To Enhance the Capabilities of Model Management Systems. Decision Support Systems. 2 (Summer 1987), 487-501.

EXECUCOM (1992). Interactive Financial Planning System: User's Manual. EXECUCOM, Austin, TX.

Fourer, R. (1983). Modeling Languages versus Matrix Generators. ACM Transactions On Mathematical Software. 9:2 143-183.

Furnas, G. W. (1986). Generalized Fisheye Views. ACM Proceedings of CHI '86 Human Factors in Computing Systems. (April, 1986), 16-23.

Gale, W. and D. Pregibon (1982). An Expert System for Regression Analysis. Computer Science and Statistics. Springer-Verlag, NY.

Geoffrion, A. M. (1975). A Guide to Computer-Assisted Methods for Distribution Systems Planning. Sloan Management Review. Winter 1975, 17-38.

Geoffrion, A. M. (1976a). The Purpose of Mathematical Programming Is Insight, Not Numbers. Interfaces. 7:1, 81-92.

Geoffrion, A. M. (1976b). Better Distribution Planning with Computer Models. Harvard Business Review. (July/August, 1976).

Geoffrion, A. M. (1987). An Introduction to Structured Modeling. Management Science. 33:5, 547-588.

Geoffrion, A. M. (1989). The Formal Aspects of Structured Modeling. Operations Research. 5, 30-51.

Geoffrion, A. M. (1992a). The SML Language for Structured Modeling: Levels 1 and 2. Operations Research. 40, 38-57.

Geoffrion, A. M. (1992b). The SML Language for Structured Modeling: Levels 3 and 4. Operations Research.40, 58-77.

Geoffrion, A. M. and G. W. Graves (1974). Multicommodity Distribution System Design by Benders Decomposition. Management Science. 20:5, 822-844.

Glover, R. D. Klingman and C. McMillan (1977). The NETFORM Concept: A More Effective Model Form and Solution Procedure for Large Scale Nonlinear Problems Annual Proceedings of the ACM. (October 16-19, 1977), 283-289.

Glover, F. D. Klingman and N. Phillips (1990). NETFORM Modeling and Applications. Interfaces. 20:4, 7-27.

Greenberg, H. J. (1983). A Functional Description of ANALYZE: A Computer-Assisted Analysis System for Linear Programming Models. ACM Transactions on Mathematical Software. 9:1, 18-56.

Greenberg, H. J. (1988). ANALYZE Rulebase in (G. Mitra, editor). Mathematical Models for Decision Support. Springer-Verlog, Berlin, 229-238.

Greenberg, H. J. (1990). A Primer of ANALYZE, Working Paper, University of Colorado at Denver. (June 1, 1990).

Greenberg, H. J. and F. H. Murphy (forthcoming). A Comparison of Mathematical Programming Modeling Systems. Annals of OR.

Hammond, K. F. (1988). Case-based Planning. Proceedings of a Workshop on Case-Based Reasoning. (May 10-13, 1988), 17-20.

Haverly Systems (1977). MaGen Reference Manual. Denville, NJ.

Hillier, F. S. and G. J. Lieberman (1986). Introduction to Operations Research (4th Ed), Holden-Day, Inc., Oakland, CA.

Hilgard, E. R. (1956). Theories of Learning,(2nd Ed). Appleton-Century-Crofts, Inc., New York.

Howard, R. A. (1990). From Influence to Relevance to Knowledge in (Oliver and Smith, editors) Influence Diagrams, Relief Nets and Decision Analysis, John Wiley & Sons, NY, 3-48.

Howard, R. A. and J. E. Matheson (1984). Influence Diagrams in (Howard and Matheson, editors), Readings on the Principles and Application of Decision Analysis. Vol II, Strategic Decision Group, Menlo Park, CA 719-762.

Hurlimann, T (1989). Reference Manual for the LPL Modeling Language (Version 3.1), Institute for Automation and Operations Research, University of Fribourg, Ch-1700 Fribourg, Switzerland.

Jarvenpaa, S. (1989). The Effect of Task and Graphical Format Congruency on Information Processing Strategies and Decision Making Performance. Management Science. 35:3, 285-303.

Johnson, E. and J. Payne (1985). Effort and Accuracy in Choice. Management Science. 31:4, 395-415.

Jones, C. V. (1988). "User Interfaces," Working Paper, University of Pennsylvania, Philadelphia, PA.

Jones, C. V. (1990). An Introduction to Graph Based Modeling Systems, Part I: Overview. ORSA Journal on Computing. 2:2, 136-151.

Jones. C. V. (1991). An Introduction to Graph Based Modeling Systems, Part II: Graph Grammars and the Implementation. ORSA Journal on Computing. 3:3, 180-206.

Keen, P. G. W. and M. Scott-Morton (1978). Decision Support Systems: An Organization Perspective. Addison-Wesley, Reading, MA.

Kendrick, D. A. (1990). "A Graphical Interface for Production and Transportation Systems Modeling: PTS," CER Working Paper 90-08. Center for Economic Research, Department of Economics, University of Texas, Austin TX.

Ketron Management Science Systems (1970). DATAFORM Mathematical Programming: User's Manual. Ketron, Inc., Arlington, VA.

Kimbrough, S. O., S. A. Moore, C. W. Pritchett and C. A. Sherman (1992). On DSS Support for Candle-Lighting Analysis, Transactions of DSS-92. 118-135.

Kimbrough, S. O., C. W. Pritchett, M. P.l Bieber and H. K. Bhargava (1990). The Coast Guard's KSS Project, Interfaces 20:6, 5-16.

King, D. (1986). The ERGO Project, DSS-86 Transactions, April 1986, 131-150.

King, W. R. and J. I. Rodriguez (1978). Evaluating Management Information Systems, MIS Quarterly 2:(September, 1978), 43-51.

Klingman, D. N. Phillips, D. Steiger, R. Wirth, R. Padman, R. Krishnan (1987). An Optimization Based Integrated Short-Term Refined Petroleum Product Planning System, Management Science, 33:7, 813-830.

Kohler, W. (1925). The Mentality of Apes. (English translation by E. Winter). Harcourt, Brace and Co., New York.

Kohler, W. (1969). The Task of Gestalt Psychology. Princeton University Press, Princeton, NJ.

Kolodner, J. L. (1987). Extended Problem Solver Capabilities Through Case-Based Inference. Proceedings of a Workshop on Case-Based Reasoning. (May 10-13, 1988), 21-30.

Kosy, D. W. and B. P. Wise (1984). Self-explanatory Financial Planning Models. Proceedings of the National Conference of Artificial Intelligence, August 1984, 176-181.

Kosy, D. W. and B. P. Wise (1986). Overview of Rome: A Reason-Oriented Modeling Environment in L.F. Psu, editor). Artificial Intelligence in Economics and Management. Elesvier Science Publishers, North-Holland, 21-30.

Krishnan, D. A. (1988). A Logic Based Approach to Model Construction, SUPA Technical Report. Carnegie-Mellon University, Pittsburg, PA.

Kurator, W. G. and R. P. O'Neill (1980). PERUSE: An Interactive System for Mathematical Programs. ACM Transactions on Mathematical Software. 6:4 489-509.

Lee, D. L. (1965). Perception, Intuition and Insight" in (W. R. Niblett, Editor) How and Why Do We Learn, Faber and Gaber, London.

Little, J. D. C. (1970). Models and Managers: Concept of a Decision Calculus. Management Science. 16:8, B466-B489.

Logan, F. A. and D. P. Ferraro (1978). Systematic Analyses of Learning and Motivation. John Wiley & Sons, New York.

Lonergan, B. J. F (1958). Insight, A Study of Human Understanding. Harper & Row, San Francisco.

Ma, P. C., F. H. Murphy and E. A. Stohr (1989). A Graphics Interface for Linear Programming. Communications of the ACM. 21:8, 996-1012.

Mathes, R. C. (1969). 'D' People and 'S' People (letter in). Science. 164 (9 May 1969),630.

MathPro, Inc. (1989). MathPro Usage Guide: Introduction and Reference. Washington, DC.

Maximal Software (1989). MPL Modeling System (Release 2.0), Maximal Software, Klapparas 11. Reykjavik, Iceland.

McIntyre, S. (1982). An Experimental Study of the Impact of Judgement-based Marketing Models. Management Science. 28:1, 17-23.

Mendenhall, W. and R. J. Beaver (1992). A Course in Business Statistics (3rd Ed.). PWS-Kent, Boston.

Microsoft (1992). Microsoft Excel User's Guide 2 (Version 4.0). Microsoft Corporation.

Mitta, D. A. (1989). Fisheye Representation of Information: IMIS User Interface. Technical Report, Texas A&M University, College Station, TX.

Mowrer, O. H. (1960). Learning Theory and the Symbolic Processes. John Wiley & Sons, New York.

Murphy, F. H. and E. A. Stohr (1986). An Intelligent System for Formulating Linear Programs. Decision Support Systems 3:3.

Palisade Corporation (1991). Risk Analysis and Simulation Add-In For Lotus 1-2-3: Version 2.0 Users Guide, Palisade Corporation, NY.

Payne, J. W. (1982). Contingent Decision Behavior. Psychology Bulletin 92:2, 382-402.

Rangaswamy, A. and J. Fedorowicz (1983). Domain-Independent Dynamic Decision Aids for Managerial Decision Making. Proceedings of the Fourth International Conference on Information Systems. Chicago, IL.

Reddy, Y. V. (1985). The Role of Introspective Simulation in Managerial Decision Making. DSS-85 Transactions, IADSS, University of Texas at Austin, 18-32.

Savage, Sam L. (1992). The ABC's of Optimization Using What's Best!, LINDO Systems Inc, Chicago.

Sharda, R. (1991). Statistical Applications of Neural Networks. Intelligent Systems Report, April 1986, 12-14.

Sharda, R. (1992). Linear Programming Software for Personal Computers: 1992 Survey. OR/MS Today. 19:3, 44-60.

Sharda, R., S. H. Barr and J. C. McDonnell (1988). Decision Support System Effectiveness: A Review and An Empirical Test. Management Science. 34:2, 139-159.

Simon, H. A. (1966). Scientific Discovery and the Psychology of Problem Solving, in Mind and Cosmos: Essays in Contemporary Science and Philosophy. University of Pittsburg Press, Pittsburg, PA.

Simon, H. A. and K. Gilmartin (1973). A Simulation of Memory for Chess Positions. Cognitive Psychology. 5, 29-46.

Sprague, R. H. and E. D. Carlson (1982). Building Effective Decision Support Systems. Prentice-Hall, Englewood Cliffs, NJ.

Steiger, D. M. and R. Sharda (1991). Neural Networks and Event Forecasting for Asset Allocation. Proceedings of the Fifth Oklahoma Symposium on Artificial Intelligence(November 10-22, 1991), 279-284.

Steiger, D. M. and R. Sharda (1992a). Recent Developments in Modeling Languages for Mathematical Programming on the Microcomputer,in. (S. Kumar, editor) Developments in Mathematical Programming 1975 - 1990. Gordon & Breach Science Publishers, 61-78.

Steiger, D. M., R. Sharda and B. LeClaire (1992b). Functional Description of a Graph-Based Interface for Network Modeling (GIN) in.(Balci, Sharda and Zenios, eds.) Computer Science and Operations Research: New Developments in Their Interfaces. Pergammon Press, 213-230.

Steiger, D. M., R. Sharda and B. LeClaire (1993a). Graphical Interfaces for Network Modeling: A Model Management System Perspective. ORSA Journal on Computing (forthcoming).

Steiger, D. M. and R. Sharda (1993b). LP Modeling Languages for Personal Computers: A Comparison. Annals of OR (forthcoming).

Swain, J. (1991). World of Choices: Simulation Software Survey. OR/MS Today. 18:5, 81-102.

TERANET(1992). ModelWare User's Manual. TERANET IA Inc., Idaho Falls, ID.

Todd, P. and I. Benbasat (1991). An Experimental Investigation of the Impact of Computer Based Decision Aids on Decision Making Strategies. Information System Research. 2:2. 87-115.

Van Horn, R. L. (1973). Empirical Studies of Management Information Systems. Data Base. (Winter, 1973), 172-180.

Wasserman, P. D. (1989). Neural Computing: Theory and Practice. Van Nostrand Reinhold, NY.

Webster's New World Dictionary (1966). The World Publishing Co., New York.

Welch, J. S. (1987). PAM -- A Practitioner's Approach to Modeling. Management Science, 33, 610-625.

Wilson, R. L., R. Sharda and M. Odom (1991). Predicting Firm Failure: Neural Networks Versus a Discriminant Analysis Approach. Proceedings of the Fifth Oklahoma Symposium on Artificial Intelligence. (November 20-22, 1991), 285-293.

Wise, B. P. and D. W. Kosy (1986). Model-Based Evaluation of Long-Range Resource Allocation Plans in. (L. F. Pau, editor) Artificial Intelligence in Economics and Management. Elesvier Science Publishers, North-Holland, 93-102.

APPENDIX A

MATERIALS FOR SUBJECTS USING INSIGHT

MATERIALS FOR SUBJECTS

USING INSIGHT

After reading the Ajax, Inc. case study, you should try analyze the 24 what-if cased provided. Since it takes approximately 2 hours so to solve each what-if case, it is impractical to run additional cases.

To aid in you analysis, you have been provided with the output of three classical evaluation tools: multiple linear regression output--in cells AH1:AM20; correlation analysis--in cells AI23:AN29; and 2-dimensional scatter diagrams--in cells Z31:AL62.

In addition, you have provided with the ANALYZE command in the FORMULA menu. This command analyzes the 24 what-if cases and provides you with the key factors and the key linear or nonlinear mathematical relations between these factors and the best number of warehoused, n*. To run this command, follow the steps below:

1) select the ANALYZE command in the FORMULA menu.

2) in the SCENARIOS dialog box, click on the "Use All Scenarios" check box and the OK button.

3) in the Select Dependent Var. dialog box, select the n* variable as the dependent variable, and click on the OK button.

4) in the left side of the Key Relationships dialog box are the key variables: i.e., those which explain most of

the variations in the best number of warehouses, n*. On the right side of the Key Relationships dialog box is the key relation(s) between these factors and n*. When you are finished viewing the dialog box, click on the OK button.

You may use the regression and correlations routines accessed through the ANALYSIS command in the FORMULA menu, and/or your general business knowledge to determine the key factors and the appropriate relations. After doing so, you should answer the then questions in the attached questionnaire.

After completing the questionnaire, please select the EXIT command in the FILE MENU, select the NOT SAVE button the Save Changes dialog box, return your DSCI 3623 DISKETTE to the lab instructor and sign the SUBJECT" NAME LIST for your 10 point credit (no signature, no credit). DO NOT PUT YOUR NAME ON THE QUESTIONNAIRE.

AJAX INC.

ANALYZING WHAT-IF CASES WITH INSIGHT

Ajax, Inc.owns a single plant in California making a range of new consumer product. Theses products are distributed nationally selling it products in Central Texas and, due to a favorable market research study, had decided to expand its operations there. Management has decided to build one or more warehoused to service the are. There are 13 cities in the area, which provide 13 possible warehouse locations, the best subset of which is to be selected. Business policy requires that each marketing city in the area be single-soured; i.e., a city may not receive goods from the 13 cities to build a warehouse in, and which cities to assign to each warehouse so as to minimize the sum of all relevant costs (fixed costs and warehouses-to-customer transportation costs).

The marker research study indicated that the product demand will be approximately the same in all 13 cities, but the total level of demand may vary considerably: i.e., by a factor of 10 or also vary widely depending on various factors, but management feels their regional proximity. Further, while transportation casts warehouse, transportation costs my vary considerably on a per mile cost (depending on partial- or full-truck load deliveries, truck versus rail rates, energy costs, energy taxes, etc.) Due to the company's current zone transportation agreement, the transportation costs from Ajax's California plant to any city in central Texas is the same. Finally, each warehouse will have sufficient capacity to supply all Central Texas demand by itself.

You have hired as a consultant to help Ajax' management determine the best number of warehoused, n*, which would minimize overall costs. Specifically, Ajax had asked you to indicate which of the three variable factors (total demand, warehouse-to-customer significant bearing on the best number

To answer these questions, you have built and run linear programming model (cleverly disguised in an EXCEL spreadsheet) and have run 24 what-if cases and stored them in EXCEL'S Scenario Manger. Now you must determine if, and to what extent, changes in the three factors affect n*. That is, you must analyze these 24 cases using scatter diagrams, statistical regression, intuition, etc. and generate your best answer to the attend attached questions.

AJAX MANAGEMENT QUESTIONS

INSIGHT ANALYSIS

1. Rate the relative importance (in determining the best number of warehouses, n*) of the following factors on a scale of 1 to 5, with 5 being very important.

total demand ____________________
1 2 3 4 5

fixed costs ____________________
1 2 3 4 5

plant-to-warehouse
transportation rates____________________
1 2 3 4 5

warehouse-to-customer
transportation rates____________________
1 2 3 4 5

At what time did you complete this question?____________

2. Which of the following factors interact to affect the best number of warehouses?

a) total demand and warehouse fixed costs only.
b) plant-to-warehouse and warehouse-to-customer transportation rates only.
c) warehouse fixed cost, warehouse-to-customer transportation rates and total demand only.
d) plant-to-warehouse and warehouse-to-customer transportation rates, total demand and warehouse fixed costs.
e) total demand and warehouse-to-customer transportation costs only.
f) insufficient information to determine effect.

At what time did you complete this question?______________

3. What is the trade-off between warehouse fixed costs and total customer demand in determining the best number of warehouses?

Describe how you arrived at your conclusion.

At what time did you complete this question?______________

4. How would doubling the warehouse fixed costs affect the best number of warehouses (holding all other factors constant)?

Describe how you arrives at your conclusion.

At what time did you complete this question?______________

5. How would increasing demand affect the best number of warehouses (holding all other factors constant)?

Describe how you arrived at your conclusion.

At what time did you complete this question? ____________

6. How would decreasing both demand and warehouse-to-customer transportation costs affect the best number of warehouse, n* (holding all other factors constant)?

Describe how you arrived at your conclusion.

At what time did you complete this question?______________

7. How would decreasing the warehouse-to-customer transportation rates affect the best number of warehouses (holding all other factors constant)?

a) increase the best number of warehouses.
b) decrease the best number of warehouses.
c) not affect the best number of warehouses.
d) insufficient information to determine the effect.

How did you arrive at the answer to the above question? (Circle all that apply)

a) by analyzing the scatter diagrams. Which one(s)?_____
b) by analyzing the regression output
c) by analyzing the correlation output
d) by using the key relation given by INSIGHT command
e) by guessing
f) other (please specify)________________________

At what time did you complete this question?___________

8. How would doubling both fixed costs and demand affect the best number of warehouses, n* (holding all other factors constant)?

a) double the best number of warehouses.
b) quadruple best number of warehouses.
c) halve best number of warehouses.
d) quarter best number of warehouses.
e) not affect best number of warehouses.
f) insufficient information to determine effect.

How did you arrive at the answer to the above question? (Circle all that apply)

a) by analyzing the scatter diagrams. Which one(s)____
b) by analyzing the regression output
c) by analyzing the correlation output
d) by using the key relation given by the INSIGHT command
e) by guessing
f) other (please specify)____________________________

At what time did you complete this question?_______________

9. How would doubling warehouse-to-customer transportation costs and halving fixed costs affect the best number of warehouses, n* (holding all other factors constant)?

a) double the best number of warehouses.
b) quadruple best number of warehouses.
c) halve best number of warehouses.
d) quarter best number of warehouses.
e) not affect best number of warehouses.
f) insufficient information to determine effect.

How did you arrive at the answer to the above question? (Circle all that apply)

a) by analyzing the scatter diagrams. Which one(s)___
b) by analyzing the regression output
c) by analyzing the correlation output
d) by using the key relation given by the INSIGHT command
e) by guessing
f) other (please specify)__________________________

At time did you complete this question?__________________

10. How would doubling total demand, fixed costs and warehouse-to-customer transportation rates affect the best number of warehouses, n* (holding all other factors constant)?

a) double the best number of warehouses.
b) quadruple best number of warehouses.
c) halve best number of warehouses.
d) quarter best number of warehouses.
e) not affect best number of warehouses.
f) insufficient information to determine effect.

How did you arrive at the answer to the above question? (Circle all that apply)

a) by analyzing the scatter diagrams. Which one(s)?____
b) by analyzing the regression output
c) by analyzing the correlation output
d) by using the key relation given by the INSIGHT command
e) by guessing
f) other (please specify)______________________

At what time did you complete this question?___________

11. Rate your confidence in the correctness of your answers in this questionnaire on a scale of 1 to 5, with 5 being very confident.

1	2	3	4	5

At what time did you complete this question?________________

APPENDIX B

MATERIALS FOR SUBJECTS NOT USING INSIGHT

AJAX INC. CASE STUDY

STATISTICAL ANALYSIS

After reading the Ajax, Inc. case study, you should try analyze the 24 what-if cased provided. Since it takes approximately 2 hours so to solve each what-if case, it is impractical to run additional cases.

To aid in you analysis, you have been provided with the output of three classical evaluation tools: multiple linear regression output--in cells AH1:AM20; correlation analysis--In cells AI23:AN29; and 2-dimensional scatter diagrams--in cells Z31:AL62.

You may use the regression and correlations routines accessed through the ANALYSIS command in the FORMULA menu, and/or your general business knowledge to determine the key factors and the appropriate relations. After doing so, you should answer the then questions in the attached questionnaire.

After completing the questionnaire, please select the EXIT command in the FILE MENU, select the NOT SAVE button the Save Changes dialog box, return your DSCI 3623 DISKETTE to the lab instructor and sign the SUBJECT" NAME LIST for your 10 point credit (no signature, no credit). DO NOT PUT YOUR NAME ON QUESTIONNAIRE.

AJAX INC.

ANALYZING WHAT-IF CASES

WITH STATISTICS

Ajax, Inc. owns a single plant in California making a range of new consumer product. Theses products are distributed nationally selling it products in Central Texas and, due to a favorable market research study, had decided to expand its operations there. Management has decided to build one or more warehoused to service the are. There are 13 cities in the area, which provide 13 possible warehouse locations, the best subset of which is to be selected. Business policy requires that each marketing city in the area be single-soured; i.e., a city may not receive goods from the 13 cities to build a warehouse in, and which cities to assign to each warehouse so as to minimize the sum of all relevant costs (fixed costs and warehouses-to-customer transportation costs).

The marker research study indicated that the product demand will be approximately the same in all 13 cities, but the total level of demand may vary considerably: i.e., by a factor of 10 or also vary widely depending on various factors, but management feels their regional proximity. Further, while transportation casts warehouse, transportation costs my vary considerably on a per mile cost (depending on partial - or full-truck load deliveries, truck versus rail rates, energy costs, energy taxes, etc.) Due to the company's current zone transportation agreement, the transportation costs from Ajax's California plant to any city in central Texas is the same. Finally, each warehouse will have sufficient capacity to supply all Central Texas demand by itself.

You have hired as a consultant to help Ajax' management determine the best number of warehoused, n*, which would minimize overall costs. Specifically, Ajax had asked you to indicate which of the three variable factors (total demand, warehouse-to-customer significant bearing on the best number of warehouses and what the appropriate relationship is.

To answer these questions, you have built and run linear programming model (cleverly disguised in an EXCEL spreadsheet) and have run 24 what-if cases and stored them in EXCEL'S Scenario Manger. Now you mst determine if, and to what extent, changes in the three factors affect n*. That is, you must analyze these 24 cases using scatter diagrams, statistical regression, intuition, etc. and generate your best answer to the attend attached questions.

AJAX MANAGEMENT QUESTIONS

STATISTICAL ANALYSIS

1. Rate the relative importance (in determining the best number of warehouses, n*) of the following factors on a scale of 1 to 5, with 5 being very important.

total demand ______________________
1 2 3 4 5

fixed costs ______________________
1 2 3 4 5

plant-to-warehouse
transportation rates______________________
1 2 3 4 5

warehouse-to-customer
transportation rates______________________
1 2 3 4 5

At what time did you complete this question?____________

2. Which of the following factors interact to affect the best number of warehouses?

a) total demand and warehouse fixed costs only.
b) plant-to-warehouse and warehouse-to-customer transportation rates only.
c) warehouse fixed cost, warehouse-to-customer transportation rates and total demand only.
d) plant-to-warehouse and warehouse-to-customer transportation rates, total demand and warehouses fixed costs.
e) total demand and warehouse-to-customer transportation costs only.
f) insufficient information to determine effect.

At what time did you complete this question?______________

3. What is the trade-off between warehouse fixed costs and total custom demand in determining the best number of warehouses?

Describe how you arrived at your conclusion.

At what time did you complete this question?______________

4. How would doubling the warehouse fixed costs affect the best number of warehouses (holding all other factors constant)?

Describe how you arrives at your conclusion.

At what time did you complete this question?______________

5. How would increasing demand affect the best number of warehouses (holding all other factors constant)?

Describe how you arrived at your conclusion.

At what time did you complete this question? __________

6. How would decreasing both demand and warehouse-to-customer transportation costs affect the best number of warehouse, n* (holding all other factors constant)?

Describe how you arrived at your conclusion.

At what time did you complete this question?______________

7. How would decreasing the warehouse-to-customer transportation rates affect the best number of warehouse (holding all other factors constant)?

a) increase the best number of warehouses.
b) decrease the best number of warehouses.
c) not affect the best number of warehouses.
d) insufficient information to determine the effect.

How did you arrive at the answer to the above question? (Circle all that apply)

a) by analyzing the scatter diagrams. Which one(s)?_____
b) by analyzing the regression output
c) by analyzing the correlation output
d) by guessing
e) other (please specify)_________________________

At what time did you complete this question?____________

8. How would doubling both fixed costs and demand affect the best number of warehouses, n* (holding all other factors constant)?

a) double the best number of warehouses.
b) quadruple best number of warehouses.
c) halve best number of warehouses.
d) quarter best number of warehouses.
e) not affect best number of warehouses.
f) insufficient information to determine effect.

How did you arrive at the answer to the above question? (Circle all that apply)

a) by analyzing the scatter diagrams. Which one(s)____
b) by analyzing the regression output
c) by analyzing the correlation output
d) by guessing
e) other (please specify)____________________________

At what time did you complete this question?_______________

9. How would doubling warehouse-to-customer transportation costs and halving fixed costs affect the best number of warehouses, n* (holding all other factors constant)?

a) double the best number of warehouses.
b) quadruple best number of warehouses.
c) halve best number of warehouses.
d) quarter best number of warehouses.
e) not affect best number of warehouses.
f) insufficient information to determine effect.

How did you arrive at the answer to the above question? (Circle all that apply)

a) by analyzing the scatter diagrams. Which one(s)___
b) by analyzing the regression output
c) by analyzing the correlation output
d) by guessing
e) other (please specify)_________________________

At time did you complete this question?__________________

10. How would doubling total demand, fixed costs and warehouse-to-customer transportation rates affect the best number of warehouses, n* (holding all other factors constant)?

a) double the best number of warehouses.
b) quadruple best number of warehouses.
c) halve best number of warehouses.
d) quarter best number of warehouses.
e) not affect best number of warehouses.
f) insufficient information to determine effect.

How did you arrive at the answer to the above question? (Circle all that apply)

a) by analyzing the scatter diagrams. Which one(s)?______
b) by analyzing the regression output
c) by analyzing the correlation output
d) by guessing
e) other (please specify)_______________________

At what time did you complete this question?___________

11. Rate your confidence in the correctness of your answers in this questionnaire on a scale of 1 to 5, with 5 being very confident.

____
1 2 3 4 5

At what time did you complete this question?________________

2

VITA

David Michael Steiger

Candidate for the Degree of

Doctor of Philosophy

Thesis: BEYOND WHAT-IF: ENHANCING MODEL ANALYSIS IN A DECISION SUPPORT SYSTEM

Major Field: Business Administration

Biographical:

Personal Data: Born in Bartlesville, Oklahoma, June 27, 1945, the son of John C. and Louise C. Steiger.

Education: Graduated from College High School, Bartlesville, Oklahoma, in June, 1963; received Bachelor of Science Degree in Electrical Engineering from The University of Texas at Austin in May, 1972; received Master of Business Administration from The University of Texas at Austin in May, 1976; completed requirements for the Doctor of Philosophy degree at Oklahoma State University in July, 1993.

Professional Experience: Assistant Professor, Decision Science Department, East Carolina University, August, 1992, to May, 1993; Instructor, Quantitative Methods/Management Information Systems Department, University of Tulsa, August, 1990, to May, 1992.