THE DESIGN OF

MAN-MACHINE DECISION SYSTEMS

by

THOMAS PATRICK GERRITY, JR.

S.B., Massachusetts Institute of Technology
(1963)
S.M., Massachusetts Institute of Technology
(1964)

SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF

PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF

TECHNOLOGY

June, 1970

Signature redacted

Signature of Author...
Alfred P. Sloan School of Management, May 14, 1970

Signature redacted

Certified by..
Thesis Supervisor

Signature redacted

Accepted by...
Chairman, Departmental Committee on Graduate Students

THE DESIGN OF MAN-MACHINE DECISION SYSTEMS

by

THOMAS P. GERRITY, JR.

Submitted to the Alfred P. Sloan School of Management on June 4, 1970, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Management.

ABSTRACT

The aim of this thesis is to contribute to a theory of man-machine decision systems (MMDS) -- their behavior and design. The thesis develops a general framework for characterizing MMDS and uses this framework as one component in a proposed methodology for MMDS design. The design methodology is then exercised and evaluated through the design and experimental use of a prototype MMDS.

A MMDS is defined as a system composed of human decision maker, computer, and decision task, all interacting in order to make decisions. Although this definition could be taken to encompass decision systems with a very slow and limited interaction between man and machine, as well as decision systems aimed at relatively simple or programmed decision tasks, the emphasis here is upon highly interactive (e.g., conversational) MMDS aimed at relatively complex and nonprogrammed decisions.

A general decision system framework is developed which identifies important characteristics of both human decision systems in general and MMDS in particular. That is, it identifies primary aggregate phases of human decision processes and major characteristics of decision system content, structure, and dynamics. It points out the major classes of design parameters available to the MMDS designer to affect decision system behavior. The framework is used to organize links to research relevant to a theory of MMDS from information systems studies and the behavioral sciences.

A model-based MMDS design methodology is proposed which relies on the general decision system framework throughout. The methodology represents a departure from traditional analysis-oriented methodologies by emphasis on an explicit and early development of detailed normative decision system characteristics. In analysis of the current decision system, the general framework is used as a guide. The methodology emphasizes formal collection of all MMDS design assumptions and expectations, representing an explicit predictive model which is then used as a standard for control on actual MMDS behavior.

The design methodology is applied and evaluated by the design and experimental use of a prototype MMDS in a field situation. The prototype was aimed at aiding in portfolio analysis for a pension fund management section of a major bank. The process of analysis, design, and implementation of the prototype is described within the phases of the proposed design methodology.

The experimental use of the system was monitored and detailed traces were produced for 29 hours of use by investment professionals and 21 hours of use by non-investment subjects. These traces are analyzed and implications are drawn for the validity of original design assumptions and for the design methodology itself.

The general results obtained suggest (1) that MMDS have unique characteristics which argue for a design methodology different from that for traditional information systems, (2) that the MMDS design methodology developed here based upon those unique characteristics can lead to the design of an effective and viable MMDS, and (3) that the MMDS facility itself represents a powerful new vehicle and tool for decision system control, research, and modeling.

Thesis Supervisor: Donald C. Carroll
Title: Professor of Management

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor, Professor Donald Carroll, for his support and encouragement, and for catching my fancy with the notion of man-machine decision systems very early in my graduate studies. I would also like to thank the members of my committee, Professors Marquis, Scott Morton, and Zannetos: Professor Marquis contributed important insights to the problems of experimental design in a field situation; I benefited greatly from Professor Zannetos' careful reading and incisive criticism of my several drafts; special thanks are due to Professor Scott Morton, whose continuing support, interest, and criticism throughout this research project were invaluable.

In addition, I would like to thank Richard Carpenter, without whose assistance in programming, the prototype MMDS might never have existed.

Also, I would like to thank Mr. James P. Baxter, whose unfailing enthusiasm and encouragement made this project possible.

Finally, my thanks go to Diana, for remaining cheerful through a long winter.

TABLE OF CONTENTS

		Page
CHAPTER 1.	INTRODUCTION	11
1.1	Man-Machine Decision Systems (MMDS) -- A Definition	11
1.2	Motivation for Research in MMDS Behavior and Design	14
1.3	Plan of the Research	16
CHAPTER 2.	METHODOLOGY IN DESIGN	19
2.1	Introduction	19
2.2	A Working Definition of Design	19
2.3	Methodology of Design	21
2.4	A Proposed General Design Methodology	25
	Structure and Process	26
	Components of the Model	28
2.5	Toward a Science of Design	39
2.6	Summary	41
CHAPTER 3.	THE SYNTHESIS OF A THEORY OF MAN-MACHINE DECISION SYSTEMS	43
3.1	Introduction	43
3.2	Precursors to a Theory of MMDS	44
3.3	Models for Decision and Decision Making	54
3.4	Synthesis of a General Decision System Framework	63
3.5	The General Decision System Framework	71
3.6	Intelligence	74
3.7	Design	79
3.8	Choice	88
3.9	Implementation and Control	93
3.10	Plans	94
3.11	Memory	97
3.12	Operators	99

TABLE OF CONTENTS (Continued)

Page

CHAPTER 3. (Continued)

3.13 Decision Structure.............................. 103
3.14 Mode of Interaction............................. 105
3.15 Structure of Interaction........................ 111
3.16 Language and Form of Interaction................ 115
3.17 Flexibility of Interaction...................... 122
3.18 Learning and Training........................... 123
3.19 Decision Programming............................ 138
3.20 Summary... 143

CHAPTER 4. THE DESIGN OF MAN-MACHINE DECISION SYSTEMS....... 145

4.1 Introduction..................................... 145
4.2 Issues in the Design of Information Systems...... 146
4.3 Principles of MMDS Design........................ 155
4.4 An MMDS Design Methodology....................... 175
4.5 Summary.. 201

CHAPTER 5. AN APPLICATION OF THE DESIGN METHODOLOGY IN PORTFOLIO MANAGEMENT........... 203

5.1 Introduction..................................... 203
5.2 The Decision Context............................. 205
5.3 The Normative Decision Model..................... 210
5.4 The Descriptive Decision Model................... 214
5.5 Functional Model................................. 229
5.6 Search and Constraining.......................... 238
5.7 Conversational Graph Specification............... 241
5.8 Detailed Design and Implementation............... 246
5.9 The MMDS Control System.......................... 273
5.10 Expected Decision System Behavior............... 282
5.11 Summary... 300

CHAPTER 6. RESULTS OF PROTOTYPE SYSTEM DESIGN AND USE....... 303

6.1 Introduction..................................... 303
6.2 Qualifications on the Results.................... 303
6.3 General Results.................................. 305
6.4 Intelligence..................................... 309
6.5 Design... 318

TABLE OF CONTENTS (Continued)

Page

CHAPTER 6. (Continued)

6.6 Choice.... 322
6.7 Plans.... 323
6.8 Memory.... 324
6.9 Decision Structure.... 326
6.10 Mode of Interaction.... 328
6.11 Structure of Interaction.... 332
6.12 Form and Language of Interaction.... 333
6.13 Flexibility of Interaction.... 335
6.14 Learning and Training.... 341
6.15 Decision Programming.... 344
6.16 Other Observations on System Usage.... 345
6.17 Summary.... 354

CHAPTER 7. CONCLUSIONS AND DIRECTIONS FOR FURTHER RESEARCH.... 358

BIBLIOGRAPHY.... 369

BIOGRAPHICAL NOTE ON THE AUTHOR.... 384

LIST OF TABLES

TABLE		Page
6.1	Number of References to Operand Types by Prototype Users	310
6.2	Frequency of Application of Prototype Functions by Users	312
6.3	Distribution of Type of User Input	336
6.4	Distribution of Degree of Command Stacking	336
6.5	Frequency of Use of Command Forms for Long-Term Versus Short-Term Users	338
6.6	Frequency of Function Usage Versus Function Complexity	350
6.7	Frequency of Use of Prototype Functions for Long-Term Versus Short-Term Users	352

LIST OF FIGURES

FIGURE		Page
1.1	The Range of Man-Computer Interaction for Decision Making.................	12
2.1	A Methodology for Design...........................	27
3.1	Various MMDS Structures of Interaction.............	114
3.2	Cognitive Mode Vs. Problem Type for Problem Solvers of Varying Ages and Degree of Task Experience..............	128
4.1	A MMDS Design Methodology..........................	176
4.2	Sample Operator-Operand Matrix Representation......	190
4.3	Sample Conversational Graph Segment................	196
5.1	Major Phases in Prototype MMDS Development and Use........................	206
5.2	The Portfolio Management Information Environment...	208
5.3	Initial Operator-Operand Matrix for the Prototype MMDS..................	239
5.4	Top Node in the Prototype MMDS Conversational Graph.............	243
5.5	Sample Segment of Detailed Prototype Conversational Graph.................	244
5.6	Initial Prototype "Sign On" Display................	250
5.7	The Top "Menu" of Alternative Functions............	251
5.8	The Full Directory of Portfolios and Stock Lists..................	252
5.9	The Status Report on a Portfolio (Page 1)........................	254

LIST OF FIGURES (Continued)

FIGURE		Page
5.10	The Status Report on a Portfolio (Page 2)........................	255
5.11	Dialogue to Specify Creation of a New List.........	257
5.12	Digression to Check the Data Items Available.......	258
5.13	Continuing to Create a List by Filtering...........	259
5.14	The New List Resulting from Filtering.............	260
5.15	Dialogue to Specify a Histogram Display............	262
5.16	Histogram Display on One Portfolio.................	263
5.17	Histogram Display on Two Portfolios................	264
5.18	Dialogue to Specify a Table Display................	266
5.19	Four-Variable Table Display on One Portfolio (Page 1)..............	267
5.20	Four-Variable Table Display on One Portfolio (Page 2)..............	268
5.21	Scatter Plot on One Portfolio.....................	270
5.22	Stock Run Report on One Stock.....................	271
6.1	Sample of Prototype Man-Machine Monitor Trace......	307

CHAPTER 1

INTRODUCTION

1.1 Man-Machine Decision Systems (MMDS) - A Definition

A man-machine decision system (MMDS) is defined here as involving the interaction of three main components:

1. man - the decision maker (one or more)
2. machine - meaning a computer, plus associated information technology necessary to support man-computer interaction
3. decision task - the problem, plus related environment and information sources.

Thus a MMDS involves the interaction of these main components to make decisions, to solve problems.

Note that this definition does not necessarily mean what have often been called simply "man-machine systems," which typically involve a man using a machine to control a physical process (e.g., a lathe) except insofar as these systems involve complex or unstructured decision making. In a sense, of course, even completely programmed man-machine activities in control of a physical process could be labelled decision making; hence the system could conceivably be called a MMDS. However MMDS here generally connotes a system aimed at complex or unstructured decision tasks. For example, one view of the range of system types suggested by the phrase "man-computer decision making" is displayed in Figure 1.1. The connotation of MMDS adopted here focuses on the lower

DEGREE OF MAN-COMPUTER INTERACTION	DECISION TYPE: Structured	DECISION TYPE: Unstructured
Low	Most Computer Applications (Batch Processing)	Some Management Science or Research Applications
High	Most On-Line Computer Applications (e.g., Information Retrieval)	MMDS

FIGURE 1.1

The Range of Man-Computer
Interaction for Decision Making

right hand corner of this range upon highly interactive or "conversational" computer systems for aiding a man in making unstructured decisions. The upper left hand corner of this space, by way of contrast, focuses upon use of relatively non-interactive (e.g., batch processing) computer systems for relatively structured functions. Batch computer output is clearly used by humans in unstructured decision making, but it is often a relatively small part of the decision "system." Typically, the decision maker facing an unstructured problem who has only a batch computer for machine support will rely much more heavily on other sources of information (e.g., other people) and more heuristic and approximate processing methods. In almost all cases, it would take a stretch of the imagination to view such a decision maker as directly and actively using the computer as an intimate decision aid--i.e., it is a relatively small component in his "decision system," if it is included within the system's bounds at all.

The field of MMDS studies has only recently, over the past ten years, begun to assume much distinct form. Because the MMDS involves quite different major components as noted above, it is not surprising that the necessary boundaries of such a field cut across a number of disciplines such as the following: computer science, management information systems, behavioral science, planning and control, design methodology, etc. Some of these are so new as to have little distinct definition themselves. Nonetheless, this new field of MMDS studies does have a central focus in its attention to the decision making process. Part of the object of this dissertation is to emphasize that focus by developing a MMDS design methodology based upon explicit models of human decision making processes.

1.2 Motivation for Research in MMDS Behavior and Design

The total investment and rate of growth of investment in computers and information systems over the past two decades has been phenomenal, and it promises to continue. A more recent surge of investment in time-shared or on-line computer systems has been even more spectacular. For example, only six years after the first successful demonstration of a general time-shared system at M.I.T. in 1963, the commercial time-sharing service market had grown (in summer of 1969) to encompass over 100 independent firms with over 275 computer installations. With commercial time-sharing services alone, industry forecasts call for a growth in sales from $150 million in 1969 to $2 billion by 1976.

Although the primary impact of this new information technology to date has been upon the lower levels and more structured tasks of the organization, there has been a growing pressure over the past ten years to bring the power of the computer directly to the aid of the decision maker facing complex and unstructured decision tasks. Despite this pressure, however, there has been little substantive progress in this area. Recently, Little (1970) has described the limited direct impact of management science models, computerized or not, on management decision making. In other words, although the necessary technology seems to exist to support MMDS, and the need and promise has been asserted by many over the past ten years, little progress toward effective design or even understanding of MMDS has been made.

Part of the problem seems to lie in the fragmented nature of research relevant to MMDS, derived as it is from many diverse disciplines. Another problem may be that traditional information systems design methodologies may be inappropriate for design of MMDS. This dissertation attacks both of these potential problems: first, by developing a general decision system framework which allows for an integrated view of research and theory relevant to MMDS; second, by developing a MMDS design methodology based upon the unique characteristics of such systems.

Such research into MMDS behavior and design is of significant and growing importance to practice. For with the advent of widely available time-shared computers and attractive conversational terminal devices over the past several years, the stage has been set for a massive investment in the development of operational MMDS. Unfortunately, given our current state of understanding of such systems, this rapid growth of MMDS application ahead of research may proceed in a dangerously unguided and ultimately very costly manner. The point is that this massive MMDS development effort will occur regardless of whether or not practitioners and academic researchers have found adequate formal models and design methodologies to support such an effort.

Thus the opportunity costs of our limited understanding of MMDS may be very great. The need is therefore correspondingly large for more MMDS research--research both in the behavior of MMDS and also into methodologies for their design. The research should have two fundamental threads: one, a broad integrating and structuring effort aimed at drawing together relevant results to date from the widely diverse fields that relate to MMDS; two, a focused and experimental effort aimed at elaborating and refining models, methodologies, and hypotheses. This dissertation pursues both of these thrusts by developing an integrated and general decision system framework and a MMDS design methodology and then exercising and evaluating them via the design and experimental use of a prototype MMDS.

1.3 Plan of the Research

The primary objective of this dissertation is to contribute to the understanding of MMDS design and behavior by developing and exercising a model-based MMDS design methodology. Chapter 2 lays the foundation for this development by surveying some of the literature on general processes of design and then proposing a new general design methodology which alleviates some of the problems noted in other methods. The proposed methodology relies heavily on the explicit use of models throughout the design process. Thus, in order to elaborate and refine this methodology for specific application to the design of MMDS, there is a clear need to develop a base of models of MMDS structure and behavior.

Chapter 3 synthesizes a general decision system framework from relevant research in the behavioral sciences and information systems. The chapter begins by surveying the evolution of various points of view toward information systems, culminating in the development of a decision centered view. Next, various representations and models of decision systems and decision processes are introduced. Finally, the general decision system framework is proposed, containing some elements common to all decision systems and others specific to MMDS. Each element of this framework is elaborated and related to research foundations, as well as to MMDS behavior.

Chapter 4 focuses on design of information systems and MMDS. It begins by noting the bias of most of the literature away from the early, less structured phases of the design process, and away from a decision-centered view of MMDS design. Considering the special characteristics of MMDS developed in Chapter 3, several principles of MMDS design are proposed and elaborated. Finally a MMDS design methodology is outlined, based on these principles as well as the general methodology of Chapter 2.

Chapter 5 elaborates the description of the MMDS design methodology of Chapter 4 in the course of applying it to a specific decision task, that of portfolio management. A field context was chosen, involving the pension fund management section of the trust department of a large bank. The design of a prototype MMDS for experimental use in the bank is described. Finally, this designer's expectations as to ultimate MMDS behavior are outlined for comparison with actual results.

Chapter 6 describes the experimental use of the prototype MMDS and compares it with expectations. The primary instrument for MMDS behavior analysis is a programmed decision monitor system which is part of the prototype. The prototype MMDS proved to be largely a success in the sense of being attractive to its users, and the majority of the designer's behavioral expectations appeared to be borne out in the relatively short period of system use. The chapter concludes with general observations on both MMDS behavior and the application of the design methodology.

Finally, chapter 7 discusses some of the salient results of this research and proposes some specific directions for further work.

CHAPTER 2

METHODOLOGY IN DESIGN

2.1 Introduction

This chapter introduces a working definition of design, surveys some of the methodological issues in design, and proposes a general design methodology.

In a survey of the literature, one soon finds that the field of design is fragmented into highly task-dependent bodies of technique. It is not the purpose of this dissertation to synthesize this diversity into a general theory of design. It is, however, its aim to develop a specific methodology for the design of a new kind of system--a man-machine decision system (MMDS). It was only when faced with this specific task that the author fully realized the total absence of a general theory of design. Hence, in order to develop some general guidelines, however rudimentary, for a specific design methodology, the general design methodology of this chapter was developed.

2.2 A Working Definition of Design

Herbert Simon sees design as a pervasive process in human activity:

> Everyone designs who devises courses of action aimed at changing existing situations into preferred ones. The intellectual activity that produces material artifacts is no different fundamentally from the one that prescribes remedies for a sick patient or the one that

> devises a new sales plan for a company or a social welfare policy for a state. Design, so construed, is the core of all professional training; it is the principal mark that distinguishes the professions from the sciences.
>
> (Simon, 1969, p. 55)

Design is an active, creative process aimed at change. It is also goal-oriented in that it seeks change for the better; it seeks to build things to perform a function.

Simon and others distinguish design from natural science, pointing out that a fundamental objective in natural science is to describe or to model an existing process or system, rather than to create a new one. To emphasize this distinction, Simon refers to the body of design as a "science of the artificial," as opposed to a science of the natural. He says that it is a science of "how things ought to be...in order to attain goals, and to function" (ibid, p. 5).

This is not to say that the models and theories of natural science are irrelevant to design. On the contrary, such models are important "building blocks" which the designer often requires as components or guidelines in the design process (e.g., electromagnetic wave theory can be useful to the electronics engineer).

Both the designer and the scientist often work with models, but their purposes are different. The scientist is searching for representations of what is, the designer is building representations of what ought to be. For example, a cognitive psychologist attempts to build a descriptive model of how a human makes decisions in the game of chess;

an operations researcher attempts to build an optimizing model for scheduling work flow in a manufacturing plant. In fact, the designer is concerned not only with what should be, but also with what could be; in other words, he must balance ultimate desired system characteristics against realistic notions of system feasibility.

With the above discussion in mind, the following working definition of design is proposed:

> Design is the on-going process of search for and construction of a desirable model of a goal-oriented process, system, or artifact.

The goals may be financial, social, esthetic, etc. The models may be physical, mental, mathematical, verbal, etc. Note that this definition of design is essentially "open-loop" in the sense of encompassing no preliminary or following stages to the process of design. The design methodology which is proposed in this chapter is, in fact, a closed-loop model, and it includes steps of problem recognition, design initiation, implementation, and control (which leads back to further problem recognition and redesign).

2.3 Methodologies of Design

A survey of the literature reveals very little work on general methodologies for design. Nadler (1967a) surveys this literature and finds that most of it deals with methods for scientific research or analysis rather than design. Simon also describes an emphasis on techniques of analysis rather than design:

> "In view of the key role of design in professional activity, it is ironic that in this century the natural sciences have almost driven the sciences of the artificial from professional school curricula. Engineering schools have become schools of physics and mathematics; medical schools have become schools of biological science; business schools have become schools of finite mathematics. The use of adjectives like "applied" conceals, but does not change, the fact. It simply means that in the professional schools those topics are selected from mathematics and the natural sciences for emphasis which are thought to be most nearly relevant to professional practice. It does not mean that design is taught, as distinguished from analysis.
>
> (Simon, 1969, p. 56)

Certainly the methodology advocated by many in the design of information systems places heavy and early emphasis upon detailed analysis of the existing system (e.g., Optner, 1968; I.B.M., 1963). Although the methodology is described with various terminology and steps by different writers, the fundamental process they advocate has three steps:

1. Define and model the current system.
2. Determine system requirements.
3. Design a new system.

In fact, these three steps are often described as strictly sequential; i.e., step two does not proceed until step one is completed and approved (I.B.M., 1963). The greatest emphasis in the literature appears to be upon step one, descriptive modeling, perhaps because that is the most structured and programmed of the three. Despite calls for creativity and original logic in step two, there is a tendency to emphasize "analyzing each activity to determine required inputs, operations, outputs and resources" based upon the structure and activity definitions of step one (ibid). Clearly the primary focus of this process is upon the detailed structure of the current process, and requirements are developed by an even tighter focus upon individual activities within the full system. This suggests a hypothesis that this analysis-based approach biases the designer toward very conservative and incremental modifications in the current system. In other words, this approach of attending to the "trees rather than the forest" will direct the designer's attention away from a more global view of the design problem and hence from more unique and revolutionary design alternatives. On the other hand, the cost-benefit characteristics of this traditional methodology obviously depend upon the nature of the problem and of the design resources available--conceivably it could be a nearly optimal method where the costs of substantial change are uncertain and potentially very high. But the obvious implicit bias toward traditional structure and conservative solutions has its own potential costs.

Other design methodologies have been proposed. Nadler (1967b) suggests the following ten step process:

1. Function determination
2. Ideal system development
3. Information gathering

4. Alternative design generation
5. Selection of solution
6. Detailed design
7. Design review
8. Test
9. Installation
10. Performance measures established

He notes that some limited field research has shown this ten step approach to produce better design results (e.g., more effective design at less design cost) than the research-oriented process.

Nadler's methodology has two primary characteristics of interest. One, it places an explicit, goal-derived "ideal system" modeling effort (steps 1 and 2) first in the sequence, before descriptive modeling (step 3), second, a formal control mechanism imbedded in the process of design is implied by the tenth step.

Rockart (1969) also decries the traditional design methodology as applied to information systems of merely finding out what the current system is and then computerizing it. He proposes also that normative thinking occur early in the process and that analysis in fact be guided by a normative model of the system.

A general methodology for design is proposed in the next section which incorporates some of the ideas of Nadler, Rockart and others regarding desirable alternatives to analysis-based methodologies.

2.4 A Proposed General Design Methodology

In the recent literature on design and throughout the previous sections, there is the general implication that the design process is simply one manifestation of problem solving or decision making. Design has been distinguished from the natural sciences by the greater degree of emphasis of the latter upon analysis. However, common threads of problem solving pervade both of these processes.

Once it is recognized that the process of design is fundamentally related to the process of problem solving, then the door is opened to many opportunities for synthesis of a "science of design" from the current base of research in cognitive psychology, management science, artificial intelligence, computer science, and other fields dealing with descriptive and normative models for information processing and decision making. Note that what is proposed here goes beyond the use of management science models, for example, within the design process and implies use of such models at the meta-level of the model of the design process itself. This follows from the assertion that the process of design itself is usefully viewed as an information processing and decision making activity.

The methodology proposed here is viewed as only one limited step in the process of this synthesis of a science of design. It is based upon some of the design ideas of Nadler, Rockart and others, as well as some of the literature of problem solving referred to above.

The general structure of the methodology is shown in Figure 2.1. Various aspects of the methodology will be discussed individually below.

Structure and Process

The basic structure as shown is meant to imply an iterative process of problem definition, design, implementation, control, problem recognition, etc. The model is not meant to suggest a sequential, uni-directional flow from one function to the next. On the contrary, the phases will blur into one another in practice, and all steps will be highly iterative. The flows shown in the figure are only meant to be suggestive of the stronger iterations between phases and are in no sense an exhaustive representation of possible interactions. Typically there would be several iterations through normative modeling, system bounding, and descriptive modeling before the general locus of design attention shifts toward design and implementation. It would also be typical that each iteration would produce a more and more detailed set of models. There is, in fact, no rigid organization implied in this process, so that there may be parallelism as well, with different people pursuing different functions at the same time.

Another structural point to make is that the design process may involve different organizations and resources for different iterations through parts of the process. For example, the first pass through the

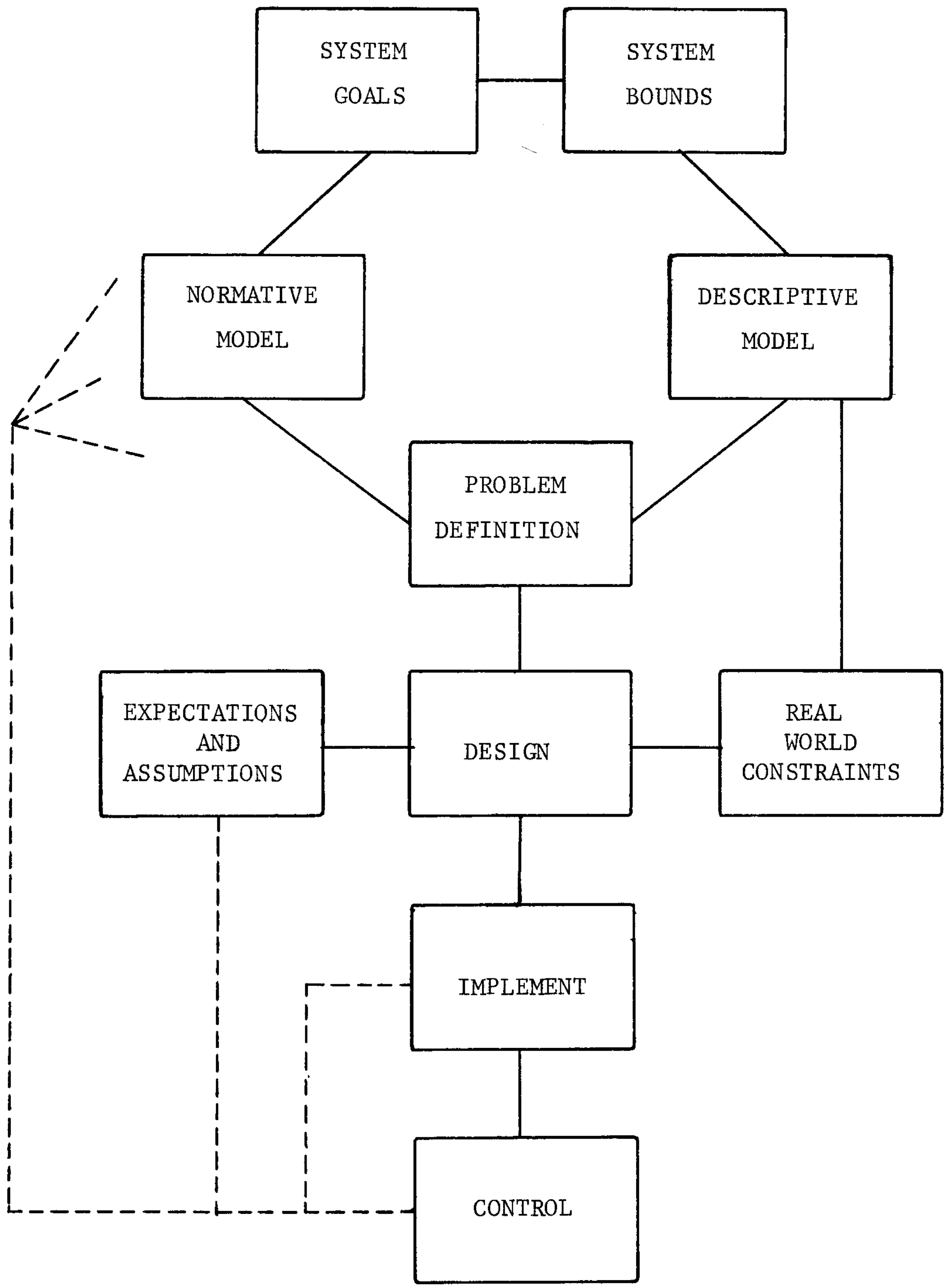


FIGURE 2.1

A Methodology for Design

design problem definition phase of the process may be almost subconscious, one person comparing a rough mental normative model with a perception of a current or expected situation and recognizing a problem. This is problem finding at its most informal. The next iteration may involve more conscious rigor and detail and yet still remain completely within the control of one person. If this iteratively defined problem and the tentative designed solutions that begin to emerge seem significant, then a multiperson design effort may be launched. It is at the point where the process becomes more than a quick, subjective recognition of a problem that a formal and logical methodology begin to be useful. As the design effort becomes larger and more cumbersome, then the decomposition, sequencing, and interaction of the phases becomes increasingly important.

Components of the Model

The identification of system goals is a fundamental function in the model, analogous to the first step in the general "systems analysis" approach (e.g., Enthoven, 1965; Quade, 1966). This step, at one extreme, could be a simple review of existing goal statement. At the other extreme, however, where goals previously have been totally implicit and ill-structured, it could mean an intellectual (and political) exercise of considerable magnitude. Some of the literature of goal definition in human decision making relevant here is reviewed in Chapter 3.

The bounding of the "system" is a key step involving a considerable degree of judgment. The trade-off that must be made is between (1) defining too large a system, over-complicating the problem and causing an excessive expenditure of modeling and design resources, and (2) defining too narrow a system so that the design is sub-optimized and may have disfunctional consequences for broad system performance. It is asserted that a great deal of design effort in computer-based information systems lies at the latter extreme; they focus too tightly on the technological system being designed to the detriment of the larger system in which it must be imbedded. The danger of this narrow view is even greater in the design of conversational man-machine decision systems, where the computer system is not so loosely coupled to decision makers in the total organization as it may be with non-interactive, batch processing computer systems.

Construction of or search for normative models is aimed at uncovering desirable standards for comparison with the current system. It involves the specification of characteristics and behavior of an ideal system. These characteristics may be derived from a number of sources: direct elaboration of system goals, search of the literature, similar systems elsewhere, abstract optimizing models, people in the system, etc. For example, Rockart (1969) discusses a case study where he adapted a model for a job shop to a medical clinic scheduling problem, thus making some of the normative techniques of job shop scheduling relevant to the medical situation.

Although the aim in the methodology is to define problems by comparing normative and descriptive models, it is proposed that the normative model be developed early in the process relative to the descriptive model. The hypothesis here is that one will be more effective and efficient in descriptive modeling if one begins with some normative constructs to guide analytic attention. A related hypothesis is that, on balance, one will arrive at a more creative and effective normative model if one builds it <u>before</u> engaging in the bulk of the descriptive modeling effort; i.e., one will be overly biased toward suggesting suboptimal, incremental modifications of the current system after having modeled it in some detail. On the other hand, it must be recognized that the danger of normative modeling with <u>too much</u> insulation from the real system is that the result will prove naive and infeasible in light of characteristics of the real process. In the extreme, of course, normative modeling cannot begin without some minimal notion of the descriptive structure and goals of the real process. Hence, descriptive and normative modeling must be parallel activities to some extent.

In any case, the explicit, distinct actions of finding or building normative models and comparing them with descriptive models or perceptions are of fundamental importance to the proposed methodology for design. Rockart, in his paper on model-based systems analysis, summarized concisely the potential advantages of use of explicit normative models:

> The problem-finding aspect of model-based studies, it is suggested, tends to identify problems not before recognized--or given full credence--and to reveal those areas

> which need additional operations research. Through the use of this technique, insights are gained into further existing operational control problems and therefore into the information system necessary to support operational decision-making. The technique helps to ensure that (1) no important areas of the system are overlooked, (2) that deficiencies in the current process are identified, and (3) that the information system is designed to be able to take advantage of improvements in the basic process as they are developed. In addition, there are subsidiary benefits from the process in terms of providing the analysts prior to the study with a better understanding of the area, better communication potential with people employed in the area, and a better base from which to plan the study itself.
>
> (Rockart, 1969, pp. 43-44)

Implicit in these advantages of early normative modeling is the notion that descriptive modeling can lock an analyst into a certain set which is focused too strongly on the detailed processes that make up "what is," thus inhibiting his ability to think freely about "what should be." This is not to say that the specific constraints and processes of the real world system should not be introduced into the process of design, but only that they should not be too much with us in the early and most creative phases. These problems of set, or "functional fixedness" in problem solving and design are surveyed and analyzed by Allen and Marquis (1964). They also show evidence of improved problem solving performance as search for alternative designs is broadened.

Beyond just encouraging wider search, an explicit normative modeling exercise can be useful for extending the designer's planning horizon. In other words, even if the "ideal" system characterized by the normative model is not immediately feasible, it at least serves as a long run standard to guide more short-term modifications in the current system. It is asserted that this represents a more intelligent approach to design than allowing design decisions to be guided primarily by short run constraints and current status.

The descriptive model construction is analogous to the typical analysis effort in computer systems analysis. It is primary input to (and may proceed somewhat parallel with) detailed design of the new system and its components. At a much less detailed level, the descriptive model suggests constraints on the feasibility of the normative model. It may even suggest omissions in the normative model.

The descriptive modeling step can often require a large investment of design resources relative to the other early steps. This is another reason why early normative modeling to guide the main descriptive effort can be worthwhile.

A key to effective descriptive modeling lies in finding appropriate and efficient representational forms to describe the state and process of the system. For example, the classic forms of computer systems analysis are flow charts and decision tables.

The process of problem definition involves a continuing comparison between the normative and descriptive models. The comparison process reveals differences between "what is" and "what should be" that constitute potential problems recognized and eventually diagnosed. The list of problems thus identified should be assigned priorities for design attention.

Pounds (1969) emphasizes the importance of models, either implicit or explicit, in the process of problem recognition. He defines problem finding as the process of defining differences between the actual state of some relevant system and its desired, or expected state. He defines problem solving (with Newell, Shaw, and Simon, 1960) as the process of finding, selecting, and applying operators which will transform or reduce differences. In other words, problem finding corresponds to Simon's Intelligence function in decision making, and problem solving corresponds to the Design and Choice functions (Simon, 1960).

Thus Pounds argues that in problem definition the desired or expected state of a system is represented or produced by a model, either implicit or explicit. Hence he implicitly asserts the importance of normative models or standards to human problem finding. Pounds reviews several classes of models which he found empirically were used in problem finding:

1. Historical models. These are models based upon a simple extrapolation of past system history to the present.
2. Planning Models. These are primarily financial models of future system behavior (e.g., budgets).
3. Other People's Models. These are models from superiors or customers, whose problem finding processes are imposed upon employees. These are the most powerful models of those found.

4. Extra-organizational Models. These are more general trade or theoretical models. These tend to be relatively little attended to as problem finders.

Since design in general is largely a human activity, Pounds' work suggests that human designers will have an implicit normative model in any case. The argument here is that it should be made explicit in many situations.

Pounds also notes that the literature of normative decision making suggests model building as the step in the process which follows problem formulation. His conclusion is that, in fact, model building should (and does in some form) precede problem formulation.

The central process of design involves a compromise--a constraining of the normative model with the realities of limited human, economic, and technological resources, inflexibilities in the current system, and problem priorities. The design is developed typically in a series of iterations, each pass modifying and adding further detail. The initial design specification, however, is developed in response to the particular problems (or "gaps") identified in the comparison of normative and descriptive models. Later design iterations show accommodations to the real world constraints, derived in part from the descriptive model and in part from limitations in design resources.

Note that the design process, at least at the initial very general level, should deal with the entire system of interest, not just with the particular artifact to be designed or modified. For example, the design of an SST, at some level of generality, should also consider social and ecological factors, as well as purely economic and aeronautical considerations, in the system. As more detail is introduced, the greatest focus will be upon the design artifact but the full system environment should not be neglected.

Note that the early stages of design involves association of potentially useful operations on systems with each of the problems identified in the normative-descriptive comparison. Implied in this process of association is an ability to classify problems or gaps with some discrimination, as well as an ability to model and to predict the effects of application of a given operation to a given problem. These processes of classification, association, and prognosis have been treated in depth by Gorry (1967, 1969) and related to a formal Bayesian model in his work on the diagnostic process.

As the design process continues, a continual sequence of design decisions must be made. Each of these choices is based on a set of expectations or assumptions about the performance of the design and the future state of the system. These design expectations and assumptions should be recorded and remembered, for they will provide a basis for control on the behavior of the designed system. In a sense, the full collection of these expectations represent the designer's model of the expected future system and its behavior. This formal predictive modeling is viewed as especially valuable in situations where ultimate design performance is relatively uncertain and a need for continual redesign and modification is seen.

The process of implementation, given detailed design specifications, is rather task-dependent in nature and is generally well covered by the literature of each specific field of design. Hence it will not be discussed here. It should be noted, however, that it is at the detailed stages of design and implementation that most of the formal techniques of cost-benefit analysis and design optimization are applied in practice.

The function of control should operate at several levels as suggested by the several feedback paths of Figure 2.1. The lowest level, that of controlling the implementation process, will be left to formalisms such as PERT and will not be treated here. There are, however, several other levels of control of interest: for example,

1. control on design assumptions,
2. control on normative model,
3. control on descriptive model,
4. control on goals,
5. control on the design process, etc.

The process of control requires a model of expected or desired system state or behavior and measures of actual system status. The proposed design methodology involves explicit models of expected designed system behavior, a normative model, a descriptive model, and a system goal structure. The function of control is to feed back system status measures which are relevant to these models. That is, experience with actual implemented system behavior may imply modifications in all of the above models. Note that provision for control, however, should be made in the process of design--i.e., control mechanisms normally should be designed in as an explicit part of the system, rather than added later as an informal afterthought.

Control on design assumptions or expected system performance is analogous to what Carroll and Zannetos (1966) have called "operating process control." Therefore, their list of requirements for "intelligent" operating process control apply also to designed system control.

Similarly, the higher level of control, that on the design process itself, is analogous to Carroll and Zannetos' "planning process control." Mechanisms for this level of control are less formalizable but no less important than other levels. The general methodology for design proposed here is a very aggregate representation. It has no pretensions of being a completely general methodology, but merely a step toward one, as well as a rough guide for the development of a specific design methodology for MMDS. It requires considerable "fleshing out" and "tuning" for application to any particular area of design. The specific techniques, heuristics, and representational forms involved within each of the functions of the design process should be made explicit and formally evaluated after each design project. A body of literature should develop in the process of design as a step toward what Simon calls a "science of design."

It was noted at the beginning of this section that the interactions between specific functions in this model of design could be many and complex. A few examples are in order:

1. It should be recognized that the process of design sometimes suggests new goals (or a different level of goals); i.e., attributes of a designed alternative system may be sufficiently novel as to suggest new criteria for system evaluation and perhaps new goals.
2. The process of implementation almost inevitably reveals unanticipated constraints in any complex system, thus calling for a modification of the design and the system model. The higher the expectation of such unanticipated consequences (or, the greater the uncertainty in the expected descriptive model), the more attention should be given design control mechanisms and design flexibility.
3. The exercise of building a normative model may suggest that original defined system bounds be modified and that descriptive modeling focus be changed accordingly. This is one reason why the methodology proposes some investment in normative model search and development very early in the design process.

It is proposed that such interactions and feedback between phases of the design process are aided by the use of formal models and representations as suggested. Also, it should be emphasized once again that, despite the rigid appearing structure of Figure 2.1, this design methodology is intended and expected to be highly interactive and iterative in this fashion.

2.5 Toward a Science of Design

The methodology described here is distinct from traditional methodologies primarily in its heavy emphasis (1) upon early, explicit and significant normative modeling, (2) upon maintaining a broad definition of system bounds, even in the more detailed functions of design, and (3) upon design and use of explicit performance measures, expected system behavior assumptions, and control mechanisms. The effects of all of these differences are testable. Although the complexity and expense of an adequate field experiment should be obvious, limited laboratory experiments and field surveys are possible as well as means to investigate this methodology.

Some of the difficulties involved in conducting research on the process of design are described by Allen (1966). He notes that most research in the field has utilized the individual case study method, but he notes the following alternatives:

1. statistical analysis of many cases;
2. matched cases (parallel sets of identical projects);
3. experimental problem situations;
4. computer simulation.

Allen advocates the matched case approach in particular because, in fact, several organizations (e.g., N.A.S.A., the Department of Defense) occasionally do award several contracts for the same design study.

For the case of the design methodology proposed here, the effects of introduction of an early and explicit normative modeling phase could be tested in an experimental situation. Groups of students at a business school doing a written case analysis and solution design might provide a reasonable set of subjects. Another possibility would be a group of student engineers with a design problem. One subgroup could be told to analyze the problem in detail, then design a solution (the traditional method); another could be told to skim the problem description briefly, spend a significant period in developing a normative model, read the description looking for key problems, and design a solution. The resulting solutions could then be compared.

As for field surveys, it is proposed that the area of information systems design may be studied fruitfully. There are a variety of E.D.P. applications which are common to a vast number of installations and which might lend themselves to a comparative survey of the effects of different design methodologies.

Obviously, any of these studies would provide challenging problems in control and design of measures. Nonetheless, they are essential steps toward improving our understanding of methodological issues in design.

2.6 Summary

This chapter has introduced a general methodology for design, which is to serve as a framework for a specific methodology for design of man-machine decision systems that will be introduced in Chapter 4. The chapter began by proposing a general working definition of design as a process of search for or construction of a goal-oriented model of a system. The sparseness of literature on general design processes was noted, and the traditional analysis-based methodology and Nadler's approach were introduced.

Finally, a general design methodology was proposed that attempts to reduce the traditional emphasis on analysis of the current system. The methodology relies heavily on explicit use of models throughout: normative and descriptive models of the process whose comparison leads to definition of specific problems; a model of expected design behavior which is necessary to guide detailed design choices throughout the design process and which can be used for control on design results. The chapter concludes with a statement of need for research into design methodologies, and several suggestions as to directions for such research.

The focus of the remainder of the dissertation is upon man-machine decision systems (MMDS), their behavior and design. Chapter 4 proposes an approach to MMDS design. Given the model-based character of the general design framework of this chapter, however, the need for understanding and models of decision system behavior should be clear.

Therefore, Chapter 3 will lay the groundwork for this MMDS design methodology. It will assimilate current theory and experience of decision systems behavior into a general decision system framework useful for MMDS design.

CHAPTER 3

THE SYNTHESIS OF A THEORY OF MAN-MACHINE DECISION SYSTEMS

3.1 Introduction

A man-machine decision system (MMDS) is defined here as a system which links man, computer, and decision task in an interactive fashion in order to make decisions. Note that this definition does not necessarily exclude less interactive (e.g., batch computer) cooperation of man and machine, and it includes systems aimed at trivial or highly programmed decisions, as well as those for highly unstructured or judgmental decisions. The main focus in this dissertation, however, is upon highly interactive (e.g., conversational) man-machine systems, and upon MMDS aimed at relatively non-programmed decisions.

As stated earlier, a prime objective of the thesis is to develop a methodology for the design of MMDS. The previous chapter has argued that such a methodology should be model-based. Hence, there is a requirement for models or representations of MMDS and their behavior. In particular, the methodology requires means for representing (1) normative characteristics of a MMDS, (2) the structure and behavior of an existing decision system, and (3) expectations as to the behavior of a particular MMDS design. This chapter will propose such a general framework for viewing and representing MMDS structure and behavior, which will be integrated into the design methodology in Chapter 4. This framework in no sense represents a complete theory of man-machine decision systems, yet it does provide a convenient form for classifying and integrating those existing bits of research and theory that are relevant to the field.

The chapter begins with a survey of the historical development of contributions to the field of MMDS. Then several models for decision systems are surveyed, and a hybrid model is proposed for use in the general decision system framework. Next, several key characteristics of MMDS in particular are identified. Finally, results from the literature, both theoretical and empirical, of MMDS and of human decision systems in general are organized and elaborated within this general framework. The result is an organized representation of expectations and hypotheses about the behavior of MMDS, which should be useful to the MMDS designer.

3.2 Precursors to a Theory of MMDS

This section introduces the field of MMDS studies by surveying a sampling of relevant literature. The survey reveals a variety of early views or models of interactive computer systems, some of which appear to be "machine-centered," others "data-centered." The survey traces the development of a third view, a "decision-centered" view, which is the one adopted in this dissertation. The recent rapid growth of technical (if not operational) demonstrations of MMDS capability is noted, and several experimental and field study forays into MMDS research are cited.

Machine-Centered and Data-Centered Studies

A significant number of studies have focused upon the behavior of time-shared computer systems. These have emphasized primarily machine-oriented measures of the dynamics of user-machine interaction with no direct relation to the user's decision process (e.g., see Scherr, 1965 and 1966; Shaw, 1965; Bryan, 1967). The measures of user behavior treated in the above studies were, for example, user "think time" (the time between machine output and next user input), console session time, input lines per session, and length of input line. The results of these user studies are surveyed by Schrage (1969).

With no adequate treatment of user behavior in MMDS, analyses of the economics of such systems, therefore, have dealt primarily with machine costs. Treatment of system benefits have been cursory at best, (e.g., Bauer, 1967) since an adequate analysis would require understanding of the human component of the total system, as well as task-dependent payoff measures.

For example, Erikson (1967) uses a typically simple aggregate model of the man in the system. He represents what he calls "user costs" as the product of two parameters: Cp, the mean cost of a user per hour of work, and Tp, the total time that the user spends in running one job (including waiting time as well as actual use time). Erikson does go further than Schrage, however, in that he uses a simple representation of the user problem, or "job," as well. He characterizes a job by two parameters: N_I, the mean number of interactions per job; T_u, the mean number of minutes of execution time used per interaction. Clearly, Erikson's model is appropriate primarily for decision situations either where the parameters are set statistically across some sizable population of user-system experience, or where the problem is so simple and structured that the parameters may be set reasonably well prior to design of the system. It is thus inappropriate for the design of MMDS where the problem is complex and unstructured, and no significant system experience exists.

There is no question that such machine-centered studies and models are valuable for the challenging task of design of time-sharing systems and scheduling algorithms (e.g., Nielson, 1968). However, they obviously offer little in the way of enlightenment as to the complex behavior of the human user.

On the other hand, there have been a significant number of studies of "human factors" in the design of interactive computer systems (see, for example, I.E.E.E. Transactions for Human Factors in Electronics; Journal of the Society of Information Display Systems; Meister and Rabideau, 1965; Shackel, 1969). These studies as conducted, however, contribute only indirectly to an understanding of the total man-machine system. As Baker notes, even these studies tend to be rather hardware-oriented in nature:

> The need is not for increased emphasis on psychological studies of display-surface characteristics (such as brightness, contrast, sharpness, etc.) although additional information in this area is desirable. Nor is improving communication through displays dependent upon

> the development of radically new or different hardware, although continual improvement in display equipment is desirable. What is needed is more research directed toward understanding the structure of standard man-computer tasks and the information requirements these structures impose and relating this knowledge, experimentally, to display parameters.
>
> (Baker, 1964, p. 430)

Beyond these machine-centered and human factors studies, there are a number of studies of data-centered man-machine systems. For example, there is a considerable literature in the study and development of data retrieval systems (Becker and Hayes, 1963; Borko, 1966). In fact, many systems that have been described as on-line "management information systems" are hardly more than rote data retrieval systems with little procedural power (e.g., Stern, 1967). For example, as will be seen later, some writers appear to see on-line, real-time computer systems primarily as data retrieval systems. This "data-centered" view of computer systems, as well as the "machine-centered" view mentioned earlier, both represent prevalent alternatives to the "decision-centered" view adopted in this thesis.

Toward a Decision-Centered View

The general problem with machine or data-centered views has been that they have ignored a central characteristic of MMDS. That is, that the effectiveness or payoff of a MMDS derives from its performance of the decision making function. The characteristics of the machine and the data obviously affect this performance, and they certainly contribute to the overall cost of the system. But their direct impact upon decision effectiveness has not been established by the analyses surveyed.

A number of writers in the field of management information systems (MIS) have tried to escape the data-centered view (e.g., Will, 1968; Baker, 1964; Emery, 1965; etc.) by making elaborate and careful distinctions between "information" (as in MIS) and "data" (as in data retrieval). Their basic notion is that "information" is "data" that has been retrieved, transformed, or displayed so as to be relevant to a decision. Such arguments, however, have little force given the strong popular connotation of data retrieval associated with the word "information." In order to avoid such semantic problems, some researchers in MIS (e.g., Scott Morton) have adopted the phrase "management decision systems" to describe the field.

It must be recognized, of course, that the problems of establishing performance measures or normative models of MMDS for problem solving are formidable. Without some reasonable model of expected behavior of a MMDS, one has little basis a priori for assessing the value of particular bits of data or particular processing functions.

Unfortunately the visible literature of man-machine decision systems is rather small, despite the extraordinary growth in use of interactive computers over the past 5-10 years. Even now, MMDS are not yet widely viewed or studied as systems. Rather, the computer and human components of the system still are often treated separately (with the human component receiving relatively little attention) to the detriment of our understanding of the total system. This situation was lamented by Anshen (1962, pp. 74-75), who argues for the "adoption of a larger frame of reference than most technical specialists are inclined to use, a recognition of the human and functional interplay." Still, as recently as 1965, Carroll pointed out that the growing set of beliefs about the value of MMDS "have been rigorously demonstrated neither in the laboratory nor in the field" (Carroll, 1965, p. 2). If the word "rigorous" is emphasized, the observation may be equally true today. At about the same time, Ruth Davis conducted an excellent comprehensive survey of "man-machine communication" and arrived at a conclusion similar to Carroll's:

> It would appear that although man-machine interaction may eventually revolutionize problem-solving, even the leaders in the field are just taking their first hesitant steps into the deep unknown of joint human-computer problem-solving techniques.
>
> (Davis, 1966, p. 243)

Even in 1967, a comprehensive survey of the literature by Sackman reveals little substantial progress:

> The literature review reveals a large and growing experimental lag between the extension of information services and verified knowledge of user performance. Except for several statistical studies of users (largely for central system cost-accounting), and except for a few experimental investigations comparing online against offline performance of users, there are virtually no empirical studies in the literature...The experimental lag is apparently endemic to the entire field of man-computer communication.
>
> (Sackman, 1967, p. 1)

The survey indicates that what Sackman calls "innovation studies" (reports of a demonstrated man-machine interaction capability in a particular area) were rampant, whereas rigorous evaluations of user effectiveness were practically non-existent.

This is not to say that there is any dearth of assertions of user effectiveness. In the area of on-line programming, recent time-shared computer advertisements claim increases in programming efficiency "up to 40%," while others go so far as to say: "human productivity in a time-sharing environment has clearly risen by a factor at least as large as five, with some optimistic surveys raising the productivity factor as high as a hundred" (Weil, 1965, p. 58). These writers were primarily concerned with on-line programming systems, but Parkhill's enthusiasm is more general:

> Now that interactive processing has become practical for even the largest computers, some exciting prospects are opening up, prospects in which the computer's enormous manipulative and computational powers will be fully melded with the imagination, intuition, and evaluative capabilities of man. Under such circumstances, the computer will become a powerful intelligence amplifier, multiplying by orders of magnitude the capabilities of the man's mind and giving him the freedom to explore in depth the most complex ramifications of his hunches.
>
> (Parkhill, 1966, p. 162)

Such enthusiasm has been tempered by the rebuttals of early critics, such as John Dearden. Dearden, in his article, "Myth of Real-Time Management Information" (1966), suggests that on-line, real-time computer systems may <u>not</u> be the end-all solution to management decision problems. He points out that although such systems may improve decision performance for certain classes of problems (e.g., simulation for strategic planning), they could very well worsen performance in many others. Basically, he argues soundly that a careful cost-effectiveness assessment of such systems must not be forgotten in a headlong rush to put a computer terminal on every manager's desk. On the other hand, Dearden has a tendency to limit his view of "on-line, real-time systems" to cover only data retrieval systems, rather than systems with accessible and flexible processing power as well, and he often fails to see the full implications of such systems for total decision system restructuring.

Nonetheless, the evidence does suggest very little effect on management decision making by interactive computer systems thus far. Although there have been a number of field surveys of the significance and impact of the computer in industry (Garrity, 1963; McKinsey and Company, 1968; Dow Jones and Company, 1969), only Brady's study of 100 top executives directly faces the question of computer impact on management decision making (Brady, 1967). His research yielded no instance of direct use of the computer by top management for decision making. In fact, he only rarely found a manager who used computer results in the same format as produced by the machine--there was in most cases intermediate processing by staff and middle management before the executive saw the result. He does, however, predict that this condition will not continue: "I believe that by 1975 the computer will have had a substantial impact on top executives' decision making..." (Brady, 1967, p. 74). Beyond expected advances in hardware, Brady attributes his prediction to anticipated breakthroughs in "decision analysis."

Outside of the management area, however, significant capabilities for interactive problem solving have been demonstrated. The classic early example of such systems is Sutherland's Sketchpad, a two-dimensional graphic computer-based design facility (Sutherland, 1963). Also, McLaughlin (1967) cites an analytical study that indicated a total reduction in shiphull design time by a computer-aided-design system from 40 to 4 hours, as well as several unofficial estimates of actual experienced computer-aided reduction in integrated circuit design time from 124 to 24 man-hours and in telemetry data reduction from 29 to 1 man-day.

Beyond the systems for data retrieval and computer-aided design already mentioned, on-line systems have also been demonstrated for numerical and non-numerical analysis (Kaplow et al, 1966; Martin, 1967), interactive budgeting (Ness, 1968), statistical analysis of large files of data (Miller, 1967; McIntosh and Griffel, 1968), administration and analysis of psychological tests for intelligence (Elithorn and Telford, 1969) ad hoc costing of manufacturing products (Morton and McCosh, 1968), and monitoring and control of space vehicle pre-launch checkout (Chesler and Turn, 1967a and 1967b), as well as surveys of other such systems (e.g., Licklider, 1965b). Several early demonstrated systems deal with general on-line aids to the process of diagnosis, using Bayesian decision theory (Shuford, 1964; Edwards, 1964a and 1964b; Schum, 1966), and later systems have been applied to the areas of medical diagnosis (Gorry, 1967; Betaque and Gorry, 1969) and juvenile probation decision making (McEachern and Newman, 1969).

Despite the large and growing number of such demonstrated systems, there have been relatively few attempts to study their detailed impact on human problem-solving behavior, apart from a few external measures of their effectiveness. But interest in the detailed decision making behavior of MMDS has been growing (e.g., Charnes and Cooper, 1965; Oettinger, 1965). Experimental MMDS studies have been conducted by Ferguson and Jones (1969), Wilkins (1968), and important field studies have been carried out by McKenney (1967) and Scott Morton (1967). The specific results of these studies will be discussed later within the general decision system framework.

Although such MMDS research is growing, however, there still is no recognized body of theory as such. Sackman describes what he calls a "general theory of man-machine digital systems" (Sackman, 1968b, p. 527); but this theory is, in fact, composed of a set of very general normative ideas for the design of MMDS, rather than a set of operational hypotheses about their behavior. (Sackman's "theory" will be discussed in Chapter 4 on the design of MMDS.)

In order to build toward such a theory of MMDS, the next section reviews several models of decision making. It then proposes a hybrid model for use in the general decision system framework of this chapter.

3.3 Models for Decision and Decision Making

This section will introduce several models for decision and decision making, both normative and descriptive, as well as other behavioral theory that is relevant to the theory of MMDS. The terms, "decision task" and "problem" will be used synonomously, as will "decision making" and "problem solving." In other words, with Simon (1960), a very broad view of the meaning of "decision making" is taken here, carrying it far beyond its typical connotation of having to do with the processes of choice alone.

Very loosely, a problem will be defined as a gap between goal and status; whereas problem solving will represent action aimed toward reducing that gap in some working "problem space." This is a definition derived from the so-called information-processing school of cognitive psychology (e.g., see Newell, 1966).

Classes of Decision Task

A primary dimension that will be used in classifying decision tasks is represented by the spectrum from programmed to non-programmed decisions. This spectrum is introduced by Simon (1960), and is analogous to the structured versus unstructured spectrum used by others (e.g., Scott Morton, 1968, p. 4). A "programmed" decision connotes decision making that is algorithmic, definite, or capable of complete specification; "non-programmed" connotes decision making that is subjective, vague, judgmental, or intuitive. Note that a sufficient condition that a decision be viewed as non-programmed is that it have a non-programmed component to the decision process; on the other hand, the fact that a decision is non-programmed does not necessarily imply that it contains no programmed functions. This last point, as has been suggested before, has implications for the potential of MMDM to allow the machine to assume some programmed functions in a basically non-programmed task.

Although this spectrum is itself rather subjective, one can speak with some confidence of more and less programmed decisions. For example, search for alternative problem solutions within a given solution space is a more programmed decision function (in the sense of involving more explicit rules and criteria) than is the search for alternative solution spaces.

Ackoff adopts a measure very similar to programmed versus non-programmed when he classifies decisions into three main groups, in order of increasing lack of structure:

a. Decisions for which adequate models are available, or can be constructed, from which optimal (or near optimal) solutions can be derived. In such cases the decision process itself should be incorporated into the information system thereby converting it (at least partially) to a control system. A decision model identifies what information is required and hence what information is relevant.

b. Decisions for which adequate models can be constructed but from which optimal solutions cannot be extracted. Here some kind of heuristic or search procedure should be provided even if it consists of no more than computerized trial and error. A simulation of the model will, as a minimum, permit comparison of proposed

> alternative solutions. Here too the model specifies what information is required.
>
> c. Decisions for which adequate models cannot be constructed. Research is required here to determine what information is relevant. If decision making cannot be delayed for completion of such research or the decision's effect is not large enough to justify the cost of research, then judgment must be used to "guess" what information is relevant. It may be possible to make explicit the implicit model used by the decision maker and treat it as a model of type (b).
>
> (Ackoff, p. B-154)

He recommends a man-machine system as providing the most flexible and adaptive approach to the last two groups. The implication is that the MMDS for decision type (b) will involve stored and accessible models in the machine, whereas for decision type (c) it would primarily provide search tools for scanning and manipulating the data base.

Another useful dimension of decision tasks involves the spectrum from simple to complex, a measure of the number of variables and of their interrelationships in a given problem situation (Scott Morton, 1968). Note that complexity does not necessarily imply a non-programmed decision although complexity may force a non-programmed approach to an otherwise clearly defined problem. The classic example is chess, a very rigorously structured game, which could be played in a perfectly programmed, but practically infeasible manner. In other words, the sheer combinatorial complexity of this very structured problem requires judgmental approaches if reasonable progress is to be made. This is also an example of an optimal approach to a problem in a narrow sense (e.g., exhaustive search) being a quite non-optimal approach in a larger sense.

In summary, two general dimensions of a decision task have been identified. The first is the spectrum from simple to complex; the second is the range from programmed to non-programmed.

Models for Decision Making

Models for decision making range from the normative to the descriptive. Normative decision models and techniques are discussed throughout the literature of economics, management science, and operations research, and a broad survey of these will not be conducted here. It should be noted, however, that most of these normative approaches require relatively well structured decision tasks if they are to be applied effectively. "Rationality" in decision making implies conforming to such normative procedures in decisions with clearly defined goals, task structures, and constraints--hence, the meaning of "rationality" outside the realm of relatively well defined decision tasks is not very clear, nor is what constitutes "good" or "bad" decision making. Developing good or normative procedures for a non-programmed decision is a highly non-programmed process itself, involving judgment and sensitivity in designing and applying decision aids and heuristics. It is too simple to conclude that a given decision was irrational by a consideration only of the explicit, well defined aspects of the problem. Hence Simon (1955) argues for substitution of a notion of "approximate" or "limited" rationality in place of the classical "global" rationality of economic man when facing fundamentally unstructured tasks.

The classic normative decision theory model focuses primarily upon choice of the optimum alternative among an array of alternatives. At its simplest, the required elements and parameters of the decision theory model for any given decision are the following:

1. All relevant alternatives, A_n;
2. All relevant possible "states of nature," S_m;
3. The utilities to the decision maker of each alternative-state combination, U_{nm};
4. The conditional probabilities for all alternative states, $P(S_m/A_n)$.

In such a decision task, a rational decision maker is one who chooses to maximize his expected utility. Unfortunately, in most real and non-programmed decision situations, the four "givens" above are practically impossible to generate. Although formal approaches to eliciting logically consistent subjective probabilities and decision maker utilities are discussed in the literature, approaches to the other two steps of (1) generation of alternatives, and (2) generation of relevant states of nature are not so well developed except in highly bounded and structured situations. In addition, beyond issues of alternative solution generation, the processes of problem recognition are not treated in this model at all. In other words, the requirements of this decision theory model may be difficult, if not impossible, to meet for many real problems. Therefore, a somewhat qualified notion of what is normative decision making is required in light of limited technological, economic, and human cognitive resources and of the unstructured nature of many real decisions of significance.

A more descriptive model of human decision making is that of Simon (1960), which is representative of a class of similar models (see, e.g., Kotler, 1967, or Dill, 1964). Simon's characterization involves the following three phases:

Intelligence: the scanning of the environment and collecting of information on various trends; then the recognition of problems as gaps between actual status and goals or normative criteria.

Design: the search for alternative means of solving the problem and/or development of methods to exploit the opportunity.

Choice: the evaluation and selection of one of the alternatives.

Simon makes the point that these functions are both iterative and recursive in any real decision process.

Soelberg introduces a similar model for non-programmed decision making, but with more detailed phases:

1. Participation
2. Recognition and Definition
3. Understanding

4. Design and Evaluation
5. Choice Reduction
6. Implementation
7. Feedback and Control

Aside from the further explication of Simon's three phases, Soelberg explicitly introduces in his model the phases of (1) commitment and goal setting, (2) implementation, and (3) control.

Another view of problem solving is that embodied in the General Problem Solver (GPS) of Newell, Shaw and Simon (1958, 1959). The basic notion in their model is that of defining a problem as a _difference_ between current _status_ and some desired state, or _goal_, and applying _operators_ to the problem situation to reduce that difference. Imbedded in the theory are several useful heuristics; one of these is called _means-end analysis_. This heuristic refers to the factoring of the problem by treating means or operators at one level as subgoals, or ends, for a lower level; thus hierarchy is introduced into the problem solving process.

Note that means-end analysis implies another heuristic--_working backwards_. In other words, one need not solve a problem by working in one direction only, by applying operators to current status to bring it closer to the goal state. The aim is to discover the "map" (and the associated set of operators) in the problem space that leads clearly from status to goal. Once can discover this map as easily in principle by applying (reversible) operators to the goal state to bring it closer to status.

Another heuristic is that of abstraction, (or the "planning method") the screening of some complexity from the problem situation, solving the resulting simpler problem, and then introducing real complexity back into the situation, attempting to maintain a feasible solution.

Note that there is no guarantee in this process that the problem solving procedure will be optimal, or will even find a solution. The particular alternative solution paths that are explored depend upon the way differences are perceived and measured, and upon the way operators and differences are associated, as well as upon the organization of the heuristics mentioned above. By attending to differences in the right order and associating operators in the right order, GPS could be quite efficient; these ordering rules, however, are heuristics and in general do not yield optimal performance.

Elsewhere, Newell refers to GPS as a version of the "heuristic search" model of problem solving, where means-end analysis is the prime search technique (Newell, 1966). He emphasizes that the choice of search space is key in this process, and is little understood.

To summarize, GPS has three basic components:

1. a memory for holding and associating information;
2. a set of basic operators, which are used to process the information stored in the memory;

3. a set of rules for combining the basic operators into decision making models or procedures.

Thus the theory of decision making represented by GPS suggests that complex human decision mechanisms can be broken down into these primitive elements.

Another model of human problem solving behavior is that of Miller, Galanter, and Pribram (1960). Despite the differences in terminology, however, this model is quite similar to GPS (e.g., "Plans" for "programs of operators" and "Images" for "current status" or "desired state") with its emphasis upon heuristic search. Their work relates to Soelberg's also in their proposal of a continuous "background" activity aimed at improving Images or models, and at initial problem recognition and understanding. They further relate to Newell in their assertion of the importance of problem representation to problem solving--that once the Image is "correct," the Plan follows almost automatically.

In fact, Miller et al assert that humans are much less capable of thinking explicitly about their Plans than about Images:

> ...the imaginal part of the process is much more accessible to awareness than is the part that deals with the formation of a Plan. An ordinary person almost never approaches a problem systematically and exhaustively unless he has been specifically educated to do so. It is much more natural for him to visualize what is and what ought to be and to focus on the gap between them than to visualize some huge set of alternative possibilities through which he must search. In other words, the phen-

> omenological aspects of problem-solving are more frequently connected with alternative Images than with alternative Plans.
>
> (ibid., p. 174)

Thus their model places emphasis on information gathering and analysis in order to improve an Image. A problem is that human decision makers seem to have relatively poor Images of their Plans in unstructured decision situations.

Their concept of a Plan is quite general. Plans can be hierarchical, with metaplans having a capability to generate many Plans. The higher is the hierarchy, the more heuristic and less rigorous the Plan. Problem solving, in their view, is the transformation of Images by Plans such that they become closer to goal Images.

3.4 Synthesis of a General Decision System Framework

This section sketches the synthesis of a general decision system framework that is held to be useful for analysis of decision systems in general and for representation and design of MMDS in particular. This general framework has several major components which are developed in the sections below. These major components are the following:

1. decision process model;
2. decision system elements;
3. characteristics of decision system interaction;
4. processes of decision system adaption.

It will be argued that these major components constitute a reasonably complete and general representation of the behavior and content of a man-machine decision system.

The decision process model encompasses the aggregate phases or activities of a decision process. The decision system elements represent the main classes of primitive decision system components or mechanisms necessary to support the decision behavior summarized in the process model. These two major components of the framework--the process model and the system elements--apply generally to decision systems, both purely human and machine-aided. The third component, characteristics of decision system interaction, outlines general characteristics relevant to a special kind of decision system involving interaction between a man and a machine--a MMDS, that is. Finally, processes of decision system adaption identify the mechanisms for long run evolution of a decision system, its process and elements, both in general and for MMDS specifically.

The general components of this framework will be summarized in the following sections, and each component will be elaborated in detail in Sections 3.6 - 3.19.

Decision Process Model

Several of the models surveyed in the previous section outlined phases of a decision process, as opposed to mechanisms or components of the decision system exhibiting the decision behavior. The object here is to synthesize a model capable of representing phases of decision making as exhibited by a decision system that involves a human decision maker (i.e., a MMDS). At the same time, the model should be general and complete enough to be capable of representing a wide variety of decision behavior.

It is proposed that Simon's general decision process model of Intelligence, Design, and Choice satisfies the above criteria. In order to provide closure and completeness to the model, however, the additional phases of Implementation and Control will also be adopted, as suggested by Soelberg's model. Although Simon considers these last two phases implicit in his model (Simon, 1960), it is useful to make them explicit as a general representation.

Another character of a decision process beyond these five general phases is the structure of the phases. In other words, there is no necessary implication that the five phases are executed in an independent and rigid fashion. In fact, there may be a significant amount of iteration between the phases, as noted by Simon (1960). The additional characteristic of decision structure is meant to summarize the pattern of this interaction in any specific situation.

Thus the decision process phases that have been identified are the following:

1. Intelligence - the process of problem recognition.
2. Design - the process of alternative solution generation.

3. Choice - the process of alternative evaluation and selection.

4. Implementation - the process of putting the chosen alternative into effect.

5. Control - the process of monitoring, evaluating, and recognizing problems in the implemented solution and the decision process itself.

6. Decision Structure - the pattern of iteration in execution of the above five phases.

Decision System Components

The review of decision theory of the previous section revealed two basic kinds of models: the decision process type of models, such as that just discussed above, and models of decision system components, such as represented by the GPS of Newell et al (1958), which support the decision process or behavior. The components of GPS are adopted here as a complete representation of the general mechanisms of a decision system. These components are the following:

1. Memory - this represents the data structures and images relevant to the decision task.

2. Operators - these represent the primitive functions applied to the data structures in memory in the process of decision-making.

3. Plans - these represent the rules, heuristics, and models used in combining primitive operators into problem-solving procedures.

Note that the only difference between these components and those of GPS is that the term "plans" of Miller et al (1960) has been adopted as a name for the third category. In any case, the "plans" of Miller et al are quite analogous to the rules and heuristics of Newell et al.

These decision system components, then, support the decision behavior represented by the process model mentioned earlier. In fact, they apply through all of the phases of that process model. In other words, there are specific operators associated with each of the process phases of intelligence, design, choice, etc. In fact, one might view a decision system in the form of a matrix having decision phases associated with columns and decision system components associated with rows, thus displaying all intersections of process phases and supporting primitive mechanisms.

Characteristics of Decision System Interaction

One attribute in particular that distinguishes a MMDS from decision systems in general is that the MMDS involves the interaction of a man and a machine. This is not to say that other decision systems involve no interaction--in fact, it is difficult to conceive of a decision situation involving a human decision maker where he would not interact in some way with external information sources or processing tools. Nonetheless, the MMDS focuses on a special kind of interaction, that between a man and a computer.

Since this man-computer interaction is one major distinguishing factor about MMDS, the characteristics of that interaction should form a significant part of any description of a MMDS. Such a set of characteristics should describe the dynamics, structure, media and language of that interaction, as well as the degree of flexibility available in these characteristics.

The dynamics of the man-machine interaction shall be referred to here as the mode of interaction. This characteristic includes the frequency distribution of message flow in that interaction, as well as the degree of control over frequency held by the decision maker.

Similarly, there is a structure of interaction, which specifies the network of communication flows in the interaction of the decision maker with the machine. For example, does the decision maker interact directly with the computer, or does he use it via another human interpreter?

Another aspect of the MMDS is the language and form of interaction. In other words, is information being transferred in the form of natural language, graphics, or computer code? Is the decision maker communicating by means of voice, handwriting, or keyboard?

A further aspect of the interactive process is the flexibility of interaction. This applies to all of the above characteristics and refers to the degree of immediately available variety in form, language, structure, or mode of interaction.

Thus the general characteristics of interaction developed above are the following:

1. Mode of Interaction.

2. Structure of Interaction.
3. Language and Form of Interaction.
4. Flexibility of Interaction.

In a sense, these characteristics can be viewed as a set of design parameters by a MMDS designer. For, in general, the nature of these characteristics will determine in part the ultimate behavior and performance of the decision system. The precise linkages between these aspects of interaction and resulting decision system behavior should be a part of a general theory of MMDS that is directly relevant for design. The manner in which these MMDS parameters of interaction may be affected by a designer will be suggested in the detailed discussion of sections 3.14 - 3.17. In addition, based on the limited experience with MMDS to date, several hypotheses will be proposed linking these parameters to MMDS behavior.

Processes of Decision System Adaption

The components of the decision system framework above characterize a decision system at some relatively local period in time. Another general component of a decision system representation should be concerned with the long-run adaption and evolution of the decision system--its process, components, and characteristics of interaction. These adaptive processes are analogous to the "planning process control" discussed by Carroll and Zannetos (1966) as a component of "intelligent information systems," only here they refer to decision process control.

The long-run evolution and adaption of a general decision system involving a human decision maker has to do with the manner in which the system learns from experience and trains new decision makers based on that experience. Thus learning and training are important processes in the long-run adaption of a general decision system.

In the specific case of a MMDS, an important process for adaption and growth is that by which the system programs aspects of the decision process or system and transfers them to the machine. This long-run continual process of explicating and structuring the decision process will be called decision programming. This is the process by which the line of demarcation in the "division of labor" between man and machine is shifted over time. In particular, the process is concerned with the shift of data structures, operators, and plans to the machine.

Note that these adaptive processes apply to all elements of the decision system identified earlier. That is, these processes are concerned with evolution in decision structure, in plans and operators, in language of interaction, etc.

Thus the primary processes identified in the long-run adaption of the MMDS are the following:

1. Learning and Training.
2. Decision Programming.

Research and evidence relevant to describing these processes are reviewed in Sections 3.18 - 3.19, and some possible relationships between other characteristics of the MMDS identified earlier and these processes are proposed.

3.5 The General Decision System Framework

The general decision system framework synthesized in the previous section is outlined below. This framework will be elaborated, element by element, in the following sections. The framework will be used to classify and organize theory, empirical research, and experience that is relevant to a theory of MMDS. The aim is to produce a summary of MMDS theory and evidence, such as it is, in a form that will be useful to the researcher and designer in the analysis, representation, and design of MMDS.

The elements of the general decision system framework are the following:

A. Decision Process Phases

1. Intelligence
2. Design
3. Choice
4. Implementation
5. Control
6. Decision Process Structure

B. Decision System Components

1. Memory
2. Operators
3. Plans

C. Characteristics of Man-Machine Interaction

1. Mode of Interaction

2. Structure of Interaction

3. Language and Form of Interaction

4. Flexibility of Interaction

D. Decision System Adaptive Processes

1. Learning and Training

2. Decision Programming

It is not argued here that this is the only such decision system framework, nor that it is necessarily the best for any particular purpose. On the other hand, it is argued that it is reasonably complete in summarizing the salient characteristics of MMDS content and behavior. So far, it has survived every attempt by this investigator to come up with another significant MMDS element or characteristic that does not fall easily within one of the existing categories of the framework. In addition, it has proven adequate as a framework for classifying and organizing the MMDS literature surveyed in this chapter; in other words, the theory and evidence of MMDS behavior to date falls conveniently within this framework.

It should also be noted that no single aspect of this framework is particularly unique in its application to decision systems. (For example, Scott Morton [1967] applied Simon's three-phase decision process model in the design and analysis of his MMDS.) Nonetheless, the synthesis of the complete framework and its application in the analysis and design of MMDS here are unique. The value of this framework in the process of MMDS design, in particular, will be argued and evaluated in Chapters 4 - 6. It will be noted that problems found in early MMDS designs can be attributed in part to a focus of attention upon only parts of this framework to the exclusion of a comprehensive view.

In the sections that follow, each element of the framework will be treated individually. Each section will survey and organize theory and evidence relevant to the particular element of the framework from research on MMDS and human decision making in general. In addition, each section will suggest some of the key dimensions or attributes which might characterize each element of the framework. It is these attributes which begin to describe the state and behavior of a MMDS. Many of them represent control parameters to the designer, who must select their "values" for a given design based on his prediction (or intuition) as to their effect on ultimate MMDS behavior.

For the designer to have much confidence in his "prediction," of course, he requires a model or theory of MMDS. As stated before, a comprehensive theory does not exist. This decision framework, however, represents a step toward this theory by identifying some of the key characteristics of a MMDS. In addition, several of the following sections propose tentative hypotheses relating some of these characteristics to MMDS behavior and performance.

3.6 Intelligence

The Intelligence phase of decision making has been defined earlier as involving the scanning of task environment and the recognition of problems in that environment as deviations from goals or normative models.

Relevant Research

Problem recognition is defined as a gap between status and goal. One typically expects to recognize a problem by spotting a situation whose status has drifted away from some set of normative criteria. It should be noted that another way of recognizing a problem is by encountering an opportunity which stimulates a modification of one's normative model, thus causing a gap. The former has been referred to as a "need-means" process, and the latter a "means-need" process (Utterback, 1968). The "means-need" opportunities, may affect the evolution of one's goals. Note that for such opportunity-stimulated innovation to take place, however the opportunity or means must not only be perceived, but it must be associated with the right problem. This process of association of means (or operations) with problems is a key function in the GPS of Newell et al, yet it has received little attention in the context of opportunity-stimulated problem recognition. The work of Marquis and his colleagues in the study of sources of innovation is a notable exception (e.g., Myers and Marquis, 1969). Note that opportunity-stimulated problem recognition is not the same as recognizing a problem via use of an "extra-organizational model" or "other people's models" as described by Pounds (1969). Pounds is referring to the adoption of a normative model from an external source. The opportunity-stimulated process refers to identification of a new mechanism, association of that mechanism with a problem situation, construction of a new model of the situation using the new mechanism, which sets higher standards than previous normative models, and thus defines a problem.

Goals, of course, are pervasive in their importance in decision making. They are relevant to Intelligence, to aid in problem recognition, to Design, as a guide to alternative search, and to Choice as determinants of selection criteria. In real decision making, the application of goals may be less rigorous than in classical decision theory. Humans are often guilty of sequential attendance to goals, rather than attempting to find a solution that maximizes some total of performance along all goal dimensions simultaneously (March and Simon, 1958).

The determinants of any given decision maker's goals at any point in time are quite complex. They may be partly a function of his personal goals, as well as of the organizational constraints relevant for his particular role (Simon, 1964). Simon also introduces the notion that some goals may be used in alternative solution generation, while others are useful in evaluation. Soelberg (1967c) hypothesizes that a decision maker has only a few primary goal attributes which, if satisfied, would dominate all other goal considerations.

Functions of Intelligence

The decision phase of intelligence involves the recognition of problems by comparing status measures with goal-related criteria. Thus the following major functions should be served by a decision system in the intelligence phase:

1. Status monitoring.
2. Goal and standard monitoring.
3. Status - standard comparison.
4. Problem definition.

Each of these functions may be performed by a variety of specific mechanisms in a decision system. The resulting behavior for each function may also exhibit a variety of characteristics. Finally, the specific mechanisms and behavioral characteristics of each function will have an impact on both the intelligence phase as well as on overall decision system behavior and performance. Some possible mechanisms with associated characteristics relevant to these functions will be suggested here, and several hypotheses will be proposed as to their effects on decision system behavior.

For example, status monitoring may be performed by such mechanisms as direct observation, formal report, graphic display, and the like. The use of these mechanisms may be characterized by such dimensions as frequency, duration, range, level of detail, etc.

Goal and standard monitoring, on the other hand, involves the definition and maintenance of the decision system's structure of goals, criteria, and standards of performance. This structure may be characterized by its consistency, complexity, degree of rigor, etc. The mechanisms by which the system maintains this structure, however, are very poorly understood, and the descriptive literature on the subject is relatively small. Nonetheless, the attributes of the decision system's goal structure are clearly important to its performance.

Status-goal comparison may occur given new information on either goals or status, or some indication that either may have changed. The goal attendance (or criteria application) may be sequential or parallel. The comparison mechanisms may be subjective, rigorous, graphic, numeric, detailed, aggregated, etc.

Problem definition occurs as gaps are sensed between status and goals. This problem definition may be either "opportunity-stimulated" or "need-stimulated." The definition could be in terms of global goals or local criteria. The problem recognized could be vaguely or clearly defined.

In a specific MMDS design, it is conceivable that several of these Intelligence functions could be programmed and very efficiently performed by the machine. For example, given normal human undependability in uniform goal attention it may be that a programmed status monitoring and periodic status-goal comparison facility could allow the system to define certain problems more efficiently and currently. This could go so far as to define "problems" based on statistically significant status variations, along the lines of the suggestions of Zannetos (1966a).

These functions, mechanisms, and attributes of Intelligence are some of the characteristics of a MMDS that the researcher should study and the designer should be aware of. Just how these characteristics actually affect MMDS behavior is not clear in general and has been directly observed in only very few special cases. Therefore, with the current state of the theory, the MMDS designer has little to go on besides intuition in structuring these mechanisms of Intelligence for any given decision system. Still, this preliminary list of functions and mechanisms at least give him a comprehensive view of some of the relevant Intelligence parameters of his design, whether all of them are controllable or not. If he formalizes his design intuition in the form of explicit design assumptions and expectations, and he controls on them after MMDS implementation, however, he will contribute to the development of a theory of MMDS.

One very simple hypothesis, for example, could be the following: given a decision system which allows the decision maker easier access to current status information, the decision system will tend to exhibit more status monitoring activity. In other words, if a decision system appears defective in the sense that problems are not recognized immediately enough, the MMDS designer might aim to make status information access easier and more timely for the decision maker.

Another MMDS design expectation might be this: given status-goal comparison mechanisms that operate on relatively aggregate dimensions of status, then the decision system will tend to define problems more in aggregate terms than local, detailed discrepancies. If the decision system appears to be defective in an excessive focus on very local and narrowly defined problems, then an MMDS designer might attack the problem by providing status-goal comparison mechanisms and comparison standards that make overall or aggregate dimensions more salient to the decision maker.

Other such expectations or hypotheses as to effects of specific MMDS Intelligence mechanisms on decision system behavior can be generated readily for a specific decision task environment, as will be demonstrated in Chapter 5. Whether or not such expected behavior is judged as desirable or not, of course, depends upon the specific situation. For example, more problem search activity is not necessarily good per se unless the value of more problems discovered is judged to be worth the cost of the mechanisms to allow and encourage such activity. Since such hypotheses are expected to be task dependent, no attempt will be made here to generate a list of them. Rather, specific MMDS hypotheses are generated in light of the specific decision task of portfolio analysis and revision in Chapter 5.

3.7 Design

The Design phase of decision making was defined earlier as involving search for and construction of alternative solutions to the problem defined in the Intelligence phase.

Relevant Research

Search processes are fundamental to decision making and to the design phase in particular. A decision maker at different times may search for problems, models, alternative solutions, and even for alternative criteria. The search for and design of models relates to understanding, learning, and the "background" improvement of Images mentioned by Miller et al (1960).

Newell emphasizes the importance of search process in the "heuristic search" model of decision making embodied in the GPS:

> Without excessive oversimplification, it may be asserted that all the successes so far in problem solving programs have come from the investigator choosing a task, discovering a suitable problem space, and programming a computer to search for solutions in this problem space...More generally, it is a common notion that hard problems are solved by finding new "viewpoints"; i.e., new problem spaces. In human problem solving different people use different problem spaces, especially in regard to the operators that are available. Not surprisingly, those with objectively more powerful spaces do better.
>
> (Newell, 1966, p. 20)

He proposes that a more general model of problem solving must treat processes of search for and selection of problem spaces, and for translation to and from problem spaces. Note that, the <u>abstraction</u> heuristic mentioned earlier in fact involves translation from a complex problem space to a simpler one and back again. Newell's purpose is to raise some of the issues of representational forms for problem spaces. Specifically, he discusses the qualities of efficiency and generality of problem space. Although by no means rigorous, the quality of efficiency suggests problem spaces which allow for (1) simple discrimination between success and failure, (2) concise formulation of powerful operators, and (3) easy elimination of irrelevant detail. Newell points out that there are no general rules for design of such problem spaces.

The converse of representation of one task in several problem spaces is representation of several tasks in one space. This raises the issue of generality, or how to construct a problem solving system to solve many problems if it is incapable of modifying its problem space significantly (e.g., GPS).

Soelberg's research delves deeply into search processes (Soelberg, 1967a, 1967b, 1967c). He identifies three basic forms of search for new alternatives: (1) "hunt and find" (HF) type search; (2) "generate and screen" (GS) type search; (3) "design, test, and modify" (DTM) type search (Soelberg, 1967a, pg. VI-30). HF-type search is that described by aspiration level models (such as Simon's satisficing), where new alternatives are identified by search one at a time and evaluated sequentially.

DTM-type search occurs in problems characterized by lack of a well-structured definition of the problem space and a lack of effective alternative generation mechanisms. Examples are R&D problems, market strategy design, reorganization design problems, etc. The basic mode is that of designing one complex alternative, testing it by applying goal criteria, modifying the alternative to better fit the criteria (or modifying the criteria), and so on. In other words, the problem solving takes on the character of a large iterative design exercise with increasing detail introduced at each iteration, as well as some factoring of the problem.

GS-type search has received primary attention in Soelberg's research. This search behavior is characterized by a relatively passive role for the decision maker after he has set his active alternative generators in motion, parallel presentation and evaluation of multiple alternatives, and successive screening of alternatives by a sequence of goal-attribute criteria, resulting in an "active roster" of a small number of alternatives for the final choice phase. Examples of this type of decision are job choice upon graduation, plant location, major equipment purchasing, etc. There are a number of distinctions between this type of problem solving and that of the satisficing model. One is the parallelism of search and of evaluation; i.e., a cycle of search within each generated alternative and evaluation against non-compared goal dimensions can be going on simultaneously in a number of alternatives. Also, search does not necessarily terminate as soon as a satisfactory alternative is identified; several acceptable alternatives may be placed in the final "active roster."

Cyert and March (1963) introduce the notion that the investment of energy in search for or design of an alternative creates a commitment of the decision maker to that alternative which goes beyond its inherent value. Whitehead reiterates this point and relates it to a need to avoid uncertainty by committing to the already explored alternative (Whitehead, 1967, pp. 97-98). Allen (1966) has observed this commitment effect in research and development projects.

In the area of MMDS research, Carroll proposes that the processes of solution search in particular may be aided by a machine, allowing for the explicit consideration of many more alternatives than otherwise possible (Carroll, 1965, p. 3). In fact, machine-aided search may allow for less dependence on human heuristics for limiting search space and alternatives considered. Of course, the relative effectiveness of such MMDS search activity depends in part upon the costs of machine search and the quality of the human heuristics. As a rough hypothesis, however, one would expect generally that the human might be relatively better at defining search spaces and the machine relatively better at search within a defined space. The degree of man-machine interaction <u>during</u> search will depend in part upon the degree of definition of solution criteria; i.e., if criteria are well defined, search is more programmed and may proceed with a minimum of dialogue with the man.

Carroll hypothesizes that another advantage of a MMDS with powerful search facilities is that it allows solution search to be deferred until a decision actually must be made (Carroll, 1965, p. 20). In other words, it can avoid a necessity for excessive pre-planning and generation of alternatives before adequate information for the particular decision is available. It can make the decision system more flexible and capable of adaptation to the unique variations of each decision encountered from the general class of decisions under consideration. For example, without such an on-line search facility, the common approach is, in effect, to conduct a limited number of preplanned and regularly scheduled searches (e.g., periodic reports). The typical result is then that the human decision maker must adapt his decision process to the available search reports, to their content and format, not vice versa. For highly programmed decisions, this sort of pre-planning may in fact be most efficient; but the opportunity costs of extending such an approach to unstructured decisions may be very high indeed.

The results of a MMDS study by Scott Morton (1967) are particularly relevant for the design phase of decision making. His field study is the first known major study of MMDS used by responsible decision makers in a live, unstructured decision situation. Basically he provided the marketing manager, the market planning manager, and the production manager of the Laundry Division of Westinghouse with an interactive CRT system for aiding in monthly design of a 12-month rolling sales forecast and production plan. Scott Morton focused his analysis upon the changes in decision process, going beyond many previous studies which attended mainly to external performance measures. The major impacts of the addition of the terminal system were the following:

1. Time. The elapsed time for monthly solution was reduced from 25 days to 1/2 day; the man-days from about 6 to 1/2 each.
2. Alternatives. The decision makers previously operated as extreme "satisficers"--as soon as they found a satisfactory solution, they stopped. With the new system Scott Morton observed them trying several alternatives beyond the first feasible schedule, a tendency toward "optimizing" behavior.

It should be emphasized that Scott Morton was studying a group of decision makers using an interactive terminal system, and the resulting group processes make the phenomena he observed more complex and perhaps different in character than the individual problem solving situations emphasized in the rest of the literature.

Discussion

The Design phase should involve functions such as the following:

1. specification of search spaces.
2. search for design components.
3. design of alternative solutions.
4. pre-screening of possible alternatives.

The actual performance of these functions may follow different modes depending on the type of problem and solution. Soelberg (1967) suggested three: generate and screen (GS), hunt and find (HF), design-test-modify (DTM). Clearly hybrid forms are possible as well, such as HF search for solution components with DTM search for a total solution specification.

The search for alternative solutions may involve search for bounded search spaces or search within a given search space. As noted earlier, the form and structure of the search space employed has a significant effect upon the type of alternatives generated (Newell, 1966). Also as mentioned, the form and timing of the search mechanism also affects significantly the resulting alternatives (Simon, 1966). In particular, one's mechanisms and perception of search space may affect dramatically the breadth of search conducted; it is hypothesized that an ill-defined or poorly perceived search space will result in highly local search, probably taking the form of incremental modifications to existing status.*

In a MMDS, the function of search in a defined space may be served by a general mechanism for scanning and filtering the space or set, based on pre-defined criteria. Mechanisms to aid in the process of solution design, on the other hand, are likely to be very problem dependent in nature.

---

*It is, of course, conceivable that a better perception of the problem space will show explicitly some previously unanticipated uncertainties, thus tending to even more conservative and local search.

One would expect that as some of these functions of searching, screening, and designing are shifted to the machine, more alternatives would tend to be generated in the decision making process (McLaughlin, 1967; Carroll, 1965; Simon, 1966; Scott Morton, 1967). Whether or not having more alternatives generated is good or not, of course, depends on the particular problem and decision system. In fact, Allen's (1966) research shows higher performance of research project teams who consider fewer alternatives.

Given a shift to the machine of some functions of solution search and design, one might expect an increase in all of the following:

1. the area of solution search;
2. the frequency of solution search;
3. the degree of guidance and control of solution search provided by problem definition in global goal terms;
4. the number of alternative designs created for comparison;
5. the relative difference between the designs considered and the current status.

Here, as elsewhere, there is a danger that shifting functions to the machine will introduce disfunctional inflexibility into the MMDM behavior. In this case, a specific danger lies in defining the search space too narrowly, biasing MMDS attention away from broader search and possibly more innovative solutions (e.g., Boehm, 1967).

3.8 Choice

The choice phase of decision making involves the evaluation, comparison, and choice from among the alternatives generated in the design phase.

Relevant Research

There is a body of psychological literature which pictures human decision makers as attempting to satisfy a set of aspiration levels, rather than maximizing some objective function (McWhinney, 1967). Simon coined the term "satisficing" to describe and contrast this behavior to optimizing (Simon, 1957). The basic notion is that the decision maker searches for or generates alternatives, typically one at a time, until he finds one that is "satisfactory," that fulfills his aspiration levels. Aspiration levels tend to drift over time, moving lower with continued lack of success, shifting upward with easy success. Also there is a basic tendency for aspiration levels to rise over time, *ceteris paribus*, as well as upon encountering outside information which indicates that other organizations or persons are doing markedly better on a similar problem even if the exact method of solution is unknown (March and Simon, 1958, p. 183). One implication of satisficing behavior is, of course, that the particular solution achieved (out of several *possible* satisfactory alternatives) will depend greatly upon the processes of search and goal attention. In a problem environment characterized by shortage of time, limited cognitive resources, and a constant backlog of problems seeking attention, a satisficing procedure could be viewed as an attempt to optimize in some larger sense.

Research into technical problem solving has supported this notion of shifting aspiration levels by indicating that goals or criteria are modified by humans to fit available alternatives in some situations (Frischmuth and Allen, 1968). That is, when faced with a set of alternative solutions, none of which meet current solution criteria, a decision maker has three alternatives:

1. modify an existing alternative to better conform to the criteria,
2. generate a new alternative, or
3. modify the criteria.

In some cases he will choose the last of these.

Note that satisficing behavior can apply to selection of decision means (operations) as well as of ends (solutions). A decision maker may tend to screen out those decision making aids or techniques that are relatively difficult to employ or to access, even if they are "more normative" in the abstract, in favor of more primitive approaches which are more available. In other words, one hypothesis about the impact of a MMDS is that as more normative decision aids are made easily accessible and usable, both decision <u>mechanisms</u> and <u>criteria</u> employed will tend to shift over time.

Soelberg's study of non-programmed decision making (1967a) indicates that human decision makers may in fact commit to a particular alternative before they are willing to admit having made a choice. This commitment is followed by a "confirmation phase" characterized by much laboring and anxiety over the choice process on the part of the decision maker and considerable perceptual and interpretational distortion of information in favor of the implicitly chosen alternative. The implication for design of MMDS is that premature commitment to an alternative is a human characteristic which must be anticipated. One way to change this behavior may be to make processes of search and comparison sufficiently easy to use so that the decision maker develops less early commitment to any single alternative by virtue of the energy invested in its investigation. In other words, part of the explanation of the behavior observed by Soelberg may lie in a human need to avoid the sheer complexity of comparison of many complicated alternatives along a multitude of non-comparable dimensions. Provisions of tools and mechanisms to cope with the complexity may allow for changes in decision behavior toward less biased evaluation and choice.

Risk and uncertainty are important attributes considered in the evaluation of alternative solutions. The experiments of Edwards indicate that humans do not handle uncertainty according to the "rational" decision theory model:

> men are incapable of extracting all of the certainty from information that Bayes' theorem indicates is in that information. To put it another way, men are conservative information processors.
>
> (Edwards et al, 1964, p. 303)

That is, in complex situations, humans tend to express less conviction in predicting an uncertain state of nature (e.g., they choose probabilities closer to 0.5 with two alternatives) than would a strict application of Bayes' theorem. The results of Soelberg (1967b, p. 25) in fact suggest that decision makers tend to aggregate uncertainty <u>into</u> their rating of alternatives along the value attributes of concern. In other words, a human tends not to make decision using "pure" probability assessments. Soelberg (1967a, p. VI-29) also notes that uncertainty does relate to search, in the sense that too much perceived uncertainty on an important goal attribute may trigger further search within the alternative under consideration.

This basic human conservatism noted by Edwards may be one factor in explaining human avoidance of risk taking in uncertain situations noted by Marquis and Reitz (1968). These studies, however, also indicate that groups tend to exhibit riskier (more decisive) behavior than individuals in the same situation; the explanation appears to be largely due to the increased familiarity and analytic understanding of the problem in the group process (Marquis, 1968). These results suggest that a MMDS which allows for more effective analysis of the problem situation may exhibit more decisiveness and less apparent risk avoidance.

For MMDS evaluation of complex solution alternatives, Carroll proposes that a model of problem system behavior is invaluable; e.g., with a simulation model of the problem area, the problem solver may test and evaluate the hypothetical behavior of alternative solutions via the simulation (Carroll, 1967, p. 356). This sort of man-machine interaction, of course, has already been demonstrated amply via batch computer systems--there has been much less evaluation of such experience with on-line models. If the human is essential for alternative generation, the on-line approach does have the advantage of shortening the cycle of alternative generation, test, and evaluation. This advantage is particularly great if the alternative search process depends to a significant degree upon the evaluation of immediately prior alternatives.

Discussion

The choice phase may involve the following functions:

1. criteria selection
2. alternative evaluation
3. alternative comparison
4. alternative selection

It is proposed that a shift of some of these choice functions to the machine will make evaluation apparently easier for the human and perhaps lead to evaluating more alternatives and less premature choice and commitment (Soelberg, 1967). The programming of some evaluation functions may also make them sufficiently explicit that some of the distortions involved in building a "rationalizing" decision rule as observed by Soelberg may be precluded. Short of complete choice programming, there are a variety of screening, comparison, ranking (multidimensional) and manipulative capabilities that could be performed by the machine.

The programming of such mechanisms for comparison and choice in a MMDS could lead to increase in the following:

1. the number of distinct alternatives explicitly compared before final choice;
2. the number and globality of dimensions of comparison.

Once again, the specific effects of programmed choice mechanisms are likely to be task-dependent. Chapter 5 suggests some specific hypotheses relating to the impact of a MMDS on choice behavior in the context of portfolio management.

3.9 Implementation and Control

The implementation phase of decision making involves the communication and execution of the chosen solution. Although the process of implementation is a critical phase in the decision process, it is also highly task-dependent. Thus it will not be discussed further within this general framework, except to note that the MMDS designer should attend to the problems of supporting a smooth transition from choice to implementation, and he should be sure the decision system possesses adequate mechanisms for communication and execution of the decision.

The decision function of control involves the sensing of deviations of chosen solutions from expectations and introducing corrective actions. In the sense of recognizing problems as gaps between status and goal (or expectations), the control function bears a strong relation to the intelligence phase. In fact, it is the control function which "closes the loop" in the decision model by feeding back new problems resulting from previous choices which require reiteration through the decision process. Note that control can operate at least at two levels:

1. control on implementation;
2. control on the decision process.

The second kind of control implies adaption or modification of the decision process as a function of experience. This phenomenon of decision system evolution and learning is discussed in detail in Section 3.18 on learning and training. Mechanisms for the first sort of control have already been discussed in Section 3.6 on the intelligence phase.

3.10 Plans

The components of a decision system which relate to Plans are the rules, heuristics, criteria, and models by which the decision system reacts to its task environment and generates procedures to manipulate it.

Relevant Research

With regard to the use of formal models by a human decision maker, Little (1970) and Amstutz (1968) have both asserted the importance of keeping such models understandable to the decision maker who must rely on them. This assertion receives support from the clinical study of McKenney (1967). McKenney provided a computation center manager with a simulation model of the center's operations so that the manager could experiment with new strategies. The model was not available on-line, but was accessible via a tutor-programmer who modified the model and interpreted it for the manager. As McKenney notes, "the only consistent characteristic was change--in both the manager's activity and the resultant model" (McKenney, 1967, p. 33). Eventually, however, the model became too complex for the computer center manager to understand--at that point, he lost faith in it and ceased to make effective use of it. McKenney asserts that part of the problem was the fact that the model was written in FORTRAN, which the manager did not understand, and thus much of the interaction with the model had to go through the tutor-programmer. This form of interaction eventually proved too laborious to maintain user understanding of the complex model, and it broke down. McKenney proposed, in fact, that one solution would have been to make the model on-line, with a command language that allowed the elimination of the interpreter intermediary.

The effectiveness of on-line interaction of a decision maker with a model has been asserted by many (e.g., Carroll, 1967; Emery, 1965). In fact, a recent experiment indicates that, with an appropriate conversational system and language, managers can learn very readily to interact effectively with an on-line model in solving a relatively structured problem (Gerrity and Black, 1970). There has been very little investigation of man-model interaction for unstructured decision making, however, either inside the laboratory or in the field.

Beyond such programmed models, the programming of decision rules or procedures may also be of value in a decision system. The research of Bowman (1963) suggests that decision performance may be improved simply by making application of a decision rule consistent over time.

Discussion

Plans are defined here as consisting of the models, concepts, and associations by which a decision maker describes his decision environment and the rules and heuristics by which he generates and applies procedures for operating upon those models or other information relevant to the task.

It is clear that some of these models or rules may be programmed and transferred to a machine in a MMDS. The research surveyed suggests that this is neither an easy nor a well understood process. However, given some effective means of interaction, the evidence does suggest that such a transfer to the machine in a MMDS may allow (1) for use of more complex, explicit models, and (2) for more consistent application of particular rules or heuristics. Section 3.19 on decision programming raises some issues relevant to the mechanics of transferring models and procedures from man to machine over a period of time.

3.11 Memory

The function of the Memory in the decision system is to store information relevant to the decision task and environment in a usefully accessible form.

Relevant Research

The characteristics of human memory are relevant to the theory and design of MMDS in the sense that they may place significant constraints on human problem solving ability. The general theory held by many today is that there are really two human memories--one short-term and the other semi-permanent with a finite, limited transfer rate from the former to the latter (Simon, 1969). The short term memory capacity appears to be quite small, about seven simple items for undistracted memory (Miller, 1956). The transfer rate from short-term to long-term memory would appear to be around 5-15 seconds per unfamiliar syllable, although there has been relatively little study of this phenomenon (Simon, 1969, p. 37). This relatively long transfer rate forces heavy reliance on short-term memory for rapid problem solving; and the limited short-term memory capacity places extremely tight constraints on the complexity or globality of problem solving strategies the human can effectively employ. Simon summarizes the limitations of the human information processing system:

> The evidence is overwhelming that the system is basically serial in its operation: that it can process only a few symbols at a time and that the symbols being processed

> must be held in special, limited memory structures whose content can be changed rapidly. The most striking limits on subjects' capacities to employ efficient strategies arise from the very small capacity of the short-term memory structure and from the relatively long time required to transfer a chunk of information from short-term to long-term memory.
>
> (Simon, 1969, p. 53)

These limitations imply a need for machine aid in a MMDS to remember data, problem solving context, and the alternative problem solving functions available at every stage in the process. They also imply need for a machine-aided ability to process, juxtapose, and compare information that cannot be stored efficiently in the human mind for processing.

Discussion

The limitations of human memory, especially short-term memory have been noted. These suggest a significant shift of the task of remembering to the machine, at least for those data, models, or operators which are sufficiently formalizable and retrievable. This is particularly true for those data that are manipulated in an explicit, programmable fashion in the process of problem solving. Other storage media are available beyond the machine (e.g., hard copy files), but these can suffer from a lack of flexibility or responsiveness.

Beyond the simple function of storage is the function of retrieval. Human beings are often strikingly good (though erratic) at associative information retrieval--they are much less effective at explicitly defined searches of large data files. This suggests that the latter function is likely to be more appropriate to shift to the machine, despite some progress in computer-based associative retrieval systems.

3.12 Operators

Operators are the primitive functions for retrieving, manipulating, transforming, and displaying information structures relevant to the decision system.

Relevant Research

Some of the most significant MMDS research in the transfer of decision making operations to the machine was done by Newman and Rogers (1966). They designed a MMDS for a particular class of problem solving behavior, inductive reasoning or concept formulation. The experimental tasks set for the subjects were similar to those employed by others (e.g., Bruner et al, 1956) in the study of concept formation. There were two main classes of task. Both began with presentation of a complex pattern of objects (via a CRT terminal) to the subject. The objective in one class of task was to discover the rule for *classification* which distinguished the given set of objects from some larger set. The objective of the other group of tasks was to discover the *relationship* which determined the pattern of objects within the given set. The latter are much the more complex of the two groups of tasks.

The MMDS provided the subjects with capabilities to screen, manipulate, and transform the symbols displayed to them. These machine functions were defined by reflection upon the general problem-solving processes involved as identified by previous research with such tasks. There were 48 subjects in all, divided into several groups: machine-aided, non-aided, and control.

One significant result was that the aided group solved more problems, solved them faster, and made fewer errors (unsuccessful rule guesses) than the non-aided group. Not only that, but the disparity in performance between the aided and non-aided group was greater for the more difficult, relational tasks.

The system aids that supported symbol transformation and recoding were used more often and with more success than other aids. In particular, this transformation approach to reducing complexity was preferred to approaches involving filtering or elimination of portions of the pattern.

Newman notes in conclusion that "the distinction between objective and subjective decision making is disappearing" in systems like the MMDS demonstrated in his experiments; i.e., decisions need no longer be thought of as either programmed or non-programmed (Simon, 1960), but it can be recognized that any complex decision may have functions of both sorts and that those functions can be shared between man and machine as suggested by Licklider (1960). The implications of this functional "intertwining" are stated by Newman:

> We are going to have to revise our thinking about the limitations of human performance...If we concentrate on providing people with computer and display aids that overcome their limitations--for example, by providing memory aids, and extending the ability to manipulate, recode, and transform data--then problem solving and decision making abilities can be expanded tremendously...We presented our subjects with problems considerably more difficult than those normally used in psychological experiments. Even so, we found that these problems could be readily solved, using the computer and display aids, and we had to increase the level of problem difficulty far beyond what we had at first anticipated in order to keep the subjects challenged. There are undoubtedly upper limits to man's intellectual ability, but we are a long way from determining just where those limits are.
>
> (Newman, 1966, p. 10)

The particular operators employed by Newman and Rogers were functions for recoding symbols, partitioning symbols, filtering symbols, etc. Scott Morton provided a number of manipulative operators in his MMDS as well, although he noted that a function for direct comparison (e.g., by overlaying two graphs) was not included and sorely needed (Scott Morton, 1967, p. 262).

Discussion

From the theoretical and empirical literature on problem solving, plus some limited experience with MMDS, the following basic operator classes have been identified by this investigator as a starting point for consideration by the MMDS designer:

1. Display - display a representation of the objects, pattern, etc. called for.
2. Logical - produce intersections, unions, complements, etc., of sets.
3. Filter - eliminate complexity by screening out portions of a set, image, etc.
4. Compare - allow two or more objects, images, patterns to be compared directly according to some aspect.
5. Mark - delineate an attribute or aspect of interest to be highlighted in a complex pattern of objects.
6. Statistical - allow for statistical aggregation of an attribute across some set of objects.
7. Create - create a semi-permanent representation of a new set, object, image for later use.
8. Delete - allow for permanent deletion of a set, object, image.
9. Ordering - allow for sequential ordering of objects according to some rule.
10. Distribute - produce frequency distributions for an attribute across some set of objects (e.g., histogram, scatter plot, contingency table, etc.).
11. Algebraic - allow for creation or calculation of a new variable as a function of others.
12. Execute - execute a procedure consisting of some string of operators such as those above.

Some of these operators may be quite primitive, others very complex. To add to the complexity, the notion of meta-operators (or meta plans) has been introduced--these are strings of simpler operators executed as programs.

Clearly, one can conceive of many such operators which are quite programmable. The key in MMDS design is identifying those programmable operators (with explicit operands) relevant to the problem of interest. Man tends to be particularly slow at explicit mental application of some of the types of operators defined above. With the machine assuming some of these functions in coordination with a large memory, a very powerful and complex MMDS can result.

3.13 Decision Structure

Decision structure pertains to the time and interactive relationships between the decision phases outlined in previous sections. It is characterized by attributes such as degree of iteration, recursion, consistency, hierarchy in the decision process.

Carroll argues that the form of a decision maker's information system is a significant factor in determining the structure of his decision process. His logic is as follows:

> Decision making is one of the fundamental purposes of a management organization; the organization is therefore structurally dependent on the nature of the decisions and its decision-making entities; decision-making is manifestly an information dependent process and can therefore be profoundly affected by new information technology.
>
> (Carroll, 1967, p. 345)

This view is supported by psychological research (Bavelas, 1950; Guetzkow and Simon, 1955), as well as by the theoretical arguments of Simon (1955). Carroll feels the impact may be particularly great in the case of MMDS:

> Since time-sharing permits a flexible division of labor between man and machine, as well as a temporal comingling of the two, it has profound implications for the procedural aspects of decision making.
>
> (Carroll, 1967, p. 351)

Note that this view is in direct contrast to that of Dearden (1966), whose projections of a limited impact of MMDS appear to be based upon an assumption that a manager's decision process is static and will not change significantly with changes in his information processing tools.

Part of the structural impact that can be expected is that decision making will become more agile and flexible, with previously rigid phases tending to blur together (Newman and Rogers, 1966). This notion is supported by the research of Scott Morton, who found a significant increase in the degree of iteration through problem solving phases after his MMDS was installed. He observed that the prior process was characterized by a relatively rigid and sequential attention to the various decision phases; whereas, with the new system, the phases became more tightly iterative, tending to blur into one another (Scott Morton, 1967, p. 229).

3.14 Mode of Interaction

The mode of interaction in a MMDS refers to the frequency and dynamics of interaction between the decision maker and computer, as well as his degree of control over those dynamics.

One representation of this iterative cycle of man-machine interaction is the "step-display-look" cycle, first proposed by Yntema (1964):

1. Step. The man takes an action which changes the state of data or process relevant to the problem at hand.
2. Display. The machine calculates and displays the results of this step.
3. Look. The man perceives and evaluates the result of his step, and he begins to plan the next step.

One of the major benefits often argued for highly interactive computer systems is their ability to reduce the cycle time from "look" to "look." There are two arguments to support this assertion of the benefit of reducing this cycle.

First, a shorter cycle reduces the amount of preplanning or foresight a person must engage in. In other words, as the time from look to look grows relatively large and costly, the human user is pressured to compensate by making his "steps" longer and longer, to attempt to plan his analysis or solution further and further ahead. Newman (1966) asserts that human beings are often not very good at this sort of pre-planning and that they operate better when they proceed sequentially with small steps and tighter feedback. Note that this does not mean that one should take a short-sighted approach to the problem during the "look" phase; it means that one should not feel overly pressured into taking long "steps" by the apparent high costs and delays of the "step-display" phases (e.g., the data entry, calculation, and output phases). One clear example of this problem is the batch computer programmer who has such a long delay between run outputs that he feels pressured to make many debugging changes at once, even trying unlikely alternatives, at a potentially high additional cost in machine time over a more iterative approach. One danger in shortening the "step-display" cycle is, of course, that the heightened pace of interaction may lead to a dis-functional shortening of the creative "look" phase as well, that highly interactive problem-solving may have an inherent tendency to become short-sighted. This will be an important phenomenon to observe and control in any operational MMDS.

The second argument for a shorter look-to-look cycle involves the "frictional" loss in shifting attention from problem to problem. In other words, if this cycle time is sufficiently long, the human decision maker will shift his attention to another problem. There seems to be a certain "set-up cost" involved in getting oneself fully into a problem again after one has left it. The more a decision maker has to shift attention from problem to problem because of the unresponsiveness of his decision tools, the higher the cumulative costs of start-up he incurs.

Various writers have suggested the possibility of several modes of man-machine interaction; where different modes are defined by the amount of processing and time the man must take between interactions. For example, Schrage's (1969) survey indicates that the distribution of user "think times" typically appears to be hyperexponential, thus indicating more short responses and more long responses than would be suggested by a Poisson model. The implication is that there may be two broad classes of computer message: that which demands a minimal though almost automatic response, and that which requires significant thought. (Alternatively, the implication might be that the human has two characteristic response modes, one much faster and more automatic than the other.) Another observation of the survey is that mean user think time appears to vary greatly among different interactive programs.

On the other hand, evidence reported by Sackman and Gold (1968) seems to indicate that there is a third significant cycle, that of hours between console sessions. Their experimental task involved complex model-building with a rigorous payoff function. Their results show very high jumps in individual user performance between sessions, with only moderate, incremental improvement within sessions. The inference is that the interactive periods of machine use are largely involved with test and verification of preconceived hypotheses. Roughly, one might view the interactive periods as alternative search and evaluation within a defined and structured solution space, whereas the creative periods of seclusion could represent search for and structuring of alternative search spaces. (In other words, the between-session restructuring of decision rules and policies within the subject's model could be viewed as the highly unstructured search for search spaces, while the on-line "tuning" of parameters within a fixed model and policy structure could be viewed as much more structured search within a defined space.)

Support is given to this notion that the dynamics of interaction affects problem solving behavior by the work of Soelberg (described in Simon, 1966, p. 47):

> Two groups of subjects were given a task amounting essentially to discovering the maximum of a function of several variables in the presence of noise. The first group was permitted to evaluate the function repeatedly for different sets of values of the independent variables, selecting a new set of values after receiving the results of the previous choice. The second group was required to make a batch of, say, ten evaluations at a time, receive the results of that batch, and then permitted to specify a new batch. Subjects in the first group almost without exception conducted local hill-climbing explorations of the function, taking account on each trial of the results of the past couple of trials. Subjects in the second group selected sets of values of the independent variables in such a way as to carry out a systematic exploration of the shape of the function.
>
> Either of these two procedures might be the more efficient for particular classes of functions. The important point is that they are different, and radically different, and that the choice was made between them not on a deliberate, rational basis, but by the impact of the stream of feedback information.

Simon (1966, p. 48) asserts there should be three modes of man-machine interaction as a result of what he calls minimum human "swap times." That is, when focused on a topic, a human may be taking steps in a second or two; when making a change of context within a task (e.g., to a new solution alternative), he may require minutes; when making a major context shift (e.g., the design of a major new alternative or new problem search space) the time required may be hours. Thus Simon suggests giving the user the alternatives of:

1. operating in a conversational mode,
2. operating with a ten to twenty minute turnaround time, or
3. submitting tasks that will be processed in about a day.

Thus the decision maker should have full control over the dynamics of interaction, plus a reasonable estimate of when he might expect a response in any given mode. The value of such control over dynamics of interaction is amply established in the psychological literature (Johnson, 1960) and experience with MMDS (Neisser, 1964; Sackman and Gold, 1968).

The effects of long response time for certain functions in biasing user selection of system facilities toward fast-response functions have been established by Scherr (1965, p. 104). This suggests very strongly that the apparent "economics" of one's problem solving tools can affect the form of problem solving behavior significantly. Certainly the results of a large number of on-line programming studies have indicated that system users get increasingly uncomfortable and irritated when average system response increases to ten seconds and beyond. In addition to length of delay, the uncertainty of an unpredictable delay can have even further negative effects on problem solving--Sackman and Gold assert that this uncertainty can be at least as significant a cost as having overly long response times in the first place.

Certainly Neisser's (1964) studies of Project MAC computer users seems to indicate that a rapid, controlled pace of man-machine interaction can produce improved performance. Neisser suggests that this improvement seems to come from "psychological continuity," the fact that a complex programming task need not undergo significant interruptions while waiting for the machine to perform a highly programmed function, such as a compilation and test run (Neisser, 1964, p. 9). One would expect that this psychological continuity effect would be generalizable to a wide variety of MMDS areas. On the other hand, Neisser suggests some of the potential costs of such continuity: (1) easy compilation may encourage sloppiness; (2) easy program modification may encourage "tinkering" beyond the point of marginal returns.

Introspective analyses by users of other time-sharing systems tend to support Neisser's main findings. One additional observation, however, is suggested by the interview of a JOSS user who noted that, "on one problem, the easy access to computing power led him to postpone a more thoughtful approach that finally succeeded" (Shaw, 1965, p. 14). This investigator has had similar experiences and has heard others report them. There appears to be a danger that a time-shared computer can lead to a detrimental attention bias in at least two general ways:

1. within a man-machine problem solving situation, the simple pace and mechanics of the man-machine dialogue may inhibit rewarding conceptual thought;
2. before beginning problem-solving, the easy availability of the machine may push overly narrow machine-oriented solution alternatives to the fore in that uncertain moment when one is selecting the first step to take.

3.15 Structure of Interaction

The structure of interaction refers to the structure of the network via which the decision maker interacts with his tools and environment. Of particular concern here, of course, is how he interacts with the computer: directly, or via an intermediary.

Certainly the notion of a MMDS suggests a totally different man-machine structure in use of the computer in problem solving than has been common to date. Weil suggests one aspect of this difference:

> Too often in the history of computer usage, the mechanics of problem solution have been mistaken for the problem itself. Now almost certainly, time-sharing will divorce the mechanics of the computer system from

> the techniques of problem solving, and in so doing, it will remove the thick layer of technologists who have grown up around the computer, returning its control to the ultimate user. Then, there will be less need for "interpreters" of the computer, those keepers of the glass-walled sanctuary who have grown as part of the computer "mystique" but who are, in fact, manifestations of an inability to use the computer easily and efficiently.
>
> (Weil, 1965, p. 58)

If Weil's assertion is true, it indicates that one major cost saving of MMDS lies in the reduced need for "interpreters" (with, of course, an attendant increase in investment in the greater complexity of computer software and hardware typically necessary to support a MMDS). It is clear that the general trend of computer hardware costs has been downward, whereas the trend for computer people costs has been upward, and that surveys have shown that the largest (and increasingly so) fraction of the data processing dollar spent by industrial enterprises has been for people (Garrity, 1963; McKinsey, 1968). Such cost trends certainly suggest that there may be profitable returns from investing the marginal dollar in systems to make the hardware more responsive and directly accessible to the human user. Problem-oriented computer languages, time-shared systems, graphic terminals, and ultimately MMDS are all manifestations of this suggestion. Similarly, Licklider asserts that effective use of computers as problem solving aids requires that they be highly interactive and <u>directly</u> available to the human problem solver:

> To solve a problem effectively, one must be able to move quickly forward from a hunch or hypothesis through definition of procedure to test, and then back again, either to revise the basic notion or to incorporate it into a larger structure of thought. Problem solving is a succession of such forward and backward excursions. The net movement must be forward, but the retreats are no less important than the advances.
>
> (Licklider, 1965, p. 21)

His implication is that conventional batch processing systems do not allow the problem solver the necessary amount of flexibility in procedure.

The "batch" model of man-machine interaction for decision making involves the human "interpreters" of Weil (1965) as shown in Figure 3.1. Three other possible versions of the structure of MMDS are also shown in the figure. The first is the direct structure, with the interpreter serving only the function of occasional system designer or modifier.* This structure is the vision of Weil and Licklider (1965).

The intermediate structure involves the "interpreter" more directly in the decision process, but does not completely separate the decision maker and the interactive system. This is the structure experienced in both phases of the clinical study by McKenney (1967, 1968), and to some extent by Scott Morton (1967).

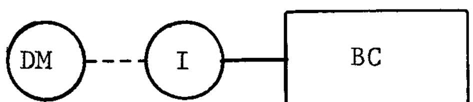
The final structure is one proposed by Mr. David Chapman of I.B.M.

---

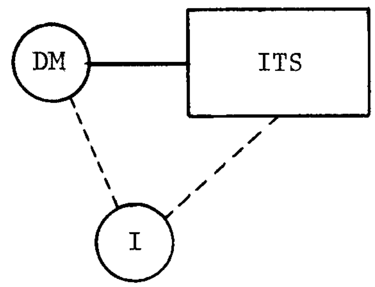
*Obviously there is a further extension of this structure where both the decision maker and even the system itself may modify or expand the system.

Abbreviations

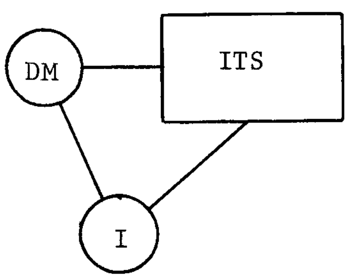
DM = Decision Maker
I = "Interpreter"
ITS = Interactive Terminal System
BC = Batch Computer



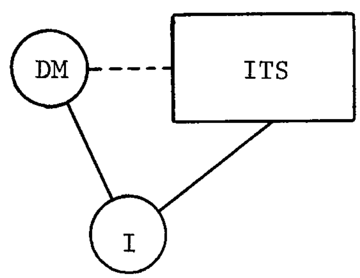
The "Batch" Structure



The Direct Structure



The Intermediate Structure



The "I.B.M." Structure

FIGURE 3.1

Various MMDS Structures of Interaction

in a Summer, 1969, seminar on MMDS given by the author. Chapman indicates that this is the structure that evolved for high level executive use of interactive terminals within I.B.M. Apparently the executives initially had desk top terminals in their offices but found quite early that they could not achieve sufficient facility with the devices to use their full power. The final configuration involves a "slaved" terminal on the executive's desk with a telephone connection to a staff analyst who is on call during the day. The analyst interprets the executive's request, issues the appropriate instructions on the "command" terminal by his side, and the executive then sees the results on the screen in his office. This scheme clearly offers tremendous flexibility, if the analyst is talented, but it has the obvious drawbacks of (1) the extra translation of the request through one more human, and (2) the cost of keeping the analyst available, even if he is "timeshared" with other tasks. Also, the seeming inability of these executives to use the terminal appears inconsistent with other cited evidence on the ease of subject learning of the mechanics of the interactive system--this certainly warrants further investigation of the problems treated and the form of the system at I.B.M.

3.16 Language and Form of Interaction

The language and form of interaction have to do with display forms, command languages, context-dependence, complexity, media, and other characteristics of the two-way dialogue of man and machine in a MMDS.

McKenney's (1967) research into a manager's interaction with a simulation model indicated that there was a tendency for the model not to stabilize but to continue to evolve in response to the manager's changing understanding. McKenney's resulting recommendation is that the language of communication between man and machine should be designed to facilitate this change process. In particular, he proposes that the language (1) should relate to the manager's own problem concept and language, (2) should allow for easy model modification and manipulation, and (3) should allow on-line access to the model, eliminating the over-dependence on the tutor-programmer as go-between.

Another exploratory MMDS study which further underlines the importance of man-machine language is that conducted by Ferguson and Jones (1969). These investigators constructed an on-line system which allows for MMDM in the scheduling of a small simulated job shop. Using a typewriter-type remote terminal as the medium of man-machine communication, about 300 subjects (students, businessmen, professors, and research assistants) exercised the MMDS under observation of the researchers. One of the main observations on this experience was the following:

> Format and vocabulary were crucial to the success of the system. As we developed formats and vocabulary which were more universally understandable and learned to present the terminology and lay-out of information more clearly, participants made increasingly better use of the system.
>
> (Ibid, p. B-559)

The general conclusion of this study is that a variety of people, many of whom have no computer background, can make meaningful use of an interactive computer-based decision aid.

Oettinger (1965) makes the suggestion that effective MMDM will make use of hierarchical forms of representation. He observes that effective systems will provide "a facility that obviously has played a major role in stabilizing both natural and artificial languages, but whose nature is adequately understood by neither philosophers, linguists, psychologists nor computer scientists. This facility is the ability to define new entities in terms of old, and the closely related and equally ill-defined and understood process labeled in mathematics as 'simplification'." (Ibid) He also observes that the form of the language of this man-machine communication will have a significant effect on performance, but he proposes that English is not necessarily the answer:

> We rebel against the constraints that badly designed computer languages impose on users but the inference that the solution is English rather than a well designed, but nevertheless specialized language, is unwarranted even where generals or chairmen of boards are concerned.
>
> (Ibid, p. 13)

He, in fact, notes that two-dimensional, graphic forms may well prove an important means of communication in MMDS's.

Baker (1964) also emphasizes the importance of language form in man-machine interaction. He is especially concerned with the prevalence of languages that are overly rigid, that are not sufficiently "forgiving" of human error or inexperience, and he quotes Watson's resulting anthropomorphic characterization of the computer as a "lightning fast, nit-picking, myopic idiot" (Watson, 1964). As a possible solution, Baker proposes use of graphic input devices, thus limiting available human responses only to those alternative regions available on the CRT screen; he argues that this would reduce the available possibilities for mechanical error (if not decision process error). Implicit in this proposal, of course, is the notion of a "menu selection" mode of communication between man and machine, with the machine presenting "menus" of alternatives and the man selecting from among the given set. This mode has the advantage of allowing very sparse, yet highly coded input from the man, as well as permitting easy change of the resulting software code structure. Baker concludes by decrying the lack of adequate research into interactive man-machine languages:

> So, even though the slowest and noisiest loop in the system is the man-to-computer communications link, little research is being done to alleviate the problem. The prevailing philosophy seems to be: wait for the development of exotic devices (e.g., automatic speech recognition and interpretation devices; electrophysiological inputs, etc.), and the problem will clear up.
>
> (Baker, 1964, p. 430)

A recent study by Wilkins (1968) contains an excellent survey of human factors literature as it relates to the design of both the man-to-machine media (input) and the machine-to-man media (output). He also relates this literature to the design of a MMDS for a specific problem-solving task: resource scheduling for BBC program development and execution. He then describes the results of experimental use of a prototype version of the operational system by five graduate student subjects. Unfortunately, however, the description of the results of this experiment is very limited in detail, and it seems to focus on the problem of input errors caused by typing rather long command words into the system. Although the operational system is projected to reduce BBC program development scheduling effort to 1% of its current level, measures of efficiency or effectiveness were not treated in the experiment. As did Baker, Wilkins argues for graphical "light button" input rather than typing as a way to code input information more efficiently and to reduce errors by controlling the number of possible responses that can be made. Also, as did Ferguson and Jones, Wilkins indicates that his experimental subjects had no significant difficulty in adapting readily to the system and making meaningful use of it. He concludes that user training requirements for the system will be minimal.

One concern in the design of a MMDS is that the limitations of many current computer terminal devices allow for display of only a limited amount of information at any one time. The experiments of Johnson in serial display of information, however, indicate the "efficiency of problem solving differs little, if at all, from efficiency under complete exposure" (Johnson, 1960, p. 75). These results should be qualified by the observations that (1) the tasks studied were small and short, and (2) the subjects had complete and speedy control over the sequence of information display.

There seem to be two general results in the studies of human tolerance for complexity in information displays. One is that, as information complexity is increased, the information processing being performed by the human on this material increases also for a time, passes through a maximum, and then falls off with increased complexity. In other words, displaying too little information does not use the full human capacity; too much information confuses it. The other general result is that the maximum point in this curve shifts toward higher and higher complexity tolerance as the human gains more experience with the problem area (i.e., as he learns better how to organize and store the material).* (See, e.g., Vitz, 1965; Schroder et al, 1967).

In the MMDS experiment of Newman and Rogers (1966) described in Section 3.12, it was noted that a recoding or transformation approach to reducing complexity was preferred by users to approaches involving filtering or elimination of portions of the pattern. This relates to the work of Sweetland (1964) in developing a display system for improved aircraft maintenance engineering which produced the following

---

*Note that the human's innate information processing "throughput" may not be changing nearly as much during this shift toward more information complexity. The higher apparent throughput may be a result of the fact that the human has absorbed much of the information already, such that a significant proportion of the display is now redundant.

observation:

> He and his colleagues reasoned, on the basis of psychological theory, that the display should emphasize spatial and verbal rather than numerical elements. This approach was highly successful and also produced an unanticipated effect: The pictorial-verbal display apparently made numerical data more tolerable. Given the general picture, the maintenance personnel persistently requested, and were able to use, more detail.
>
> (Newman, 1966, p. 6)

In other words, the information processing capability of the MMDS may indeed be greater than the man alone. In general, human tolerance of complexity in problem-solving seems to be a function of the form and structure of the information displays. In the Newman and Rogers case the humans were given direct control over that form and structure, with considerable success. As Newman notes:

> "people don't mind dealing with complexity if they have some way of controlling or handling it...if a person is allowed to structure a complex situation according to his perceptual and conceptual needs, sheer complexity is no bar to effective performance."
>
> (Newman, 1966, p. 9)

The use of graphic displays is advocated strongly by Scott Morton (1968), and supporting evidence of the efficacy of graphic displays is provided by Sweetland (1964) and Newman (1966). Qualifying evidence that graphics offers no automatic improvement unless well designed to be appropriate to the task is given by Gerrity and Black (1970) and Bell (1968).

3.17 Flexibility of Interaction

Flexibility in a decision system refers to the variety and immediate adaptability of the mode, structure, language and form of interaction. In other words, it refers to the ability of the decision system to react quickly and adapt to a short-run change in task requirements, or even to a new human decision maker with a different problem solving style and capability.

Research in decision making has established the great variety in human problem solving strategy and styles, especially for nonprogrammed decisions (e.g., Hunt et al, 1966). Sackman's (1967) survey of on-line programming experiments indicates wide variations in overall task performance as well, often more than an order of magnitude between best and worst performers, even with relatively stratified subject samples. Patrick (1967, p. 44) noted one on-line programming study which showed that the range of hours required for debugging varied by a factor of 25:1, that the program sizes varied by 5:1, and that the running times varied by 13:1.

On the other hand, psychological literature has indicated some of the rigidities that purely human problem solving can fall prey to, such as premature and excessive commitment to an alternative investigated (Cyert and March, 1963), and the consequent distortions of pre-decision (Soelberg, 1967) and post-decision (Festinger, 1957 and 1964) dissonance reduction.

An MMDS, however, may offer sufficiently increased flexibility in the decision process that these observed rigidities and distortions may be reduced by lowering the apparent cost to the decision maker of investigating a new alternative solution. This potential flexibility of MMDS has been noted both by Newman and Rogers (1966) and Scott Morton (1967) in the highly iterative "blurring together" of what were previously rather rigidly sequential phases of decision making. Others assert that the greater flexibility in procedure possessed by a MMDS, the better is the resulting performance in nonprogrammed decision making (e.g., Licklider, 1964). Carroll (1965) asserts the resulting reduced need for rigid preplanning of the decision making process. Newman (1966) notes the potential to avoid getting locked into the wrong "set," a great delayer in creative or unstructured problems. His experiments indicate that the tendency to become locked into the wrong set is reduced by the flexibility of an efficient step-look-display cycle. Needless to say, such flexibility is not an inherent property of a MMDS, but achieving it may be an important design goal.

3.18 Learning and Training

The Learning and Training element of the general decision system framework has to do with the long-run cognitive growth and behavioral changes of the decision system. This may imply long-run adaption in understanding, operators, and decision structure, as well as mode, structure, language, and flexibility of interaction.

In describing the results of their MMDS analysis, Sackman and

Gold (1968) also review some of the psychological literature, focussing especially upon massed versus spaced learning and whole versus part learning. Massed learning is seen as analogous to on-line learning since it represents more intensive and continuous activity over longer periods of time than spaced learning, which, like batch, is made up of shorter sessions distributed between long intervals. The analogy breaks down to some extent in that batch interaction withholds feedback until the next session, whereas experiments with spaced learning give immediate feedback within each session. From the literature, the general characteristics of massed learning that emerge are the following:

1. Less time invested in "warm up";
2. Better for short, simple tasks learned directly to completion;
3. Better for tasks requiring much exploration;
4. Better for very complex tasks;
5. Less forgetting between sessions.

The contrasted characteristics of spaced learning are the following:

1. Less fatigue and boredom;
2. Better for longer, but routine tasks;
3. Lessened tendency to become locked into wrong "set";

The literature of part versus whole learning shows that less experienced subjects do better when long tasks are factored into manageable subproblems. This provides one rationale for Sackman and Gold's experimental observation that the less competent problem solvers did relatively better on time sharing, which allows for shorter exploratory steps than does batch.

Bruner (1964), in discussing cognitive growth, identifies three basic human representational forms--enactive, iconic, and symbolic--motor, perceptual, and ratiocinative.

> By enactive representation I mean a mode of representing past events through appropriate motor response...Iconic representation summarizes events by the selective organization of percepts and of images, by the spatial, temporal, and qualitative structures of the perceptual field and their transformed images. Images "stand for" perceptual events in the conventionally selective way that a picture stands for the object pictured. Finally, a symbol system represents things by design features that include remoteness and arbitrariness. A word neither points directly to its referent here and now, nor does it resemble it as a picture.
>
> (Ibid, p. 2)

Bruner suggests that human capacity to <u>use</u> these representational forms develops slowly through childhood, with passage beyond enactive to include the iconic stage around age one year, and on to symbolic around age two to three. Although Bruner does not emphasize it, there appears to be an interaction between problem type and the cognitive mode applied to it; i.e., for some problem tasks, experiments indicate the transition from iconic to symbolic taking place around ages six to seven, not two to three.

There are also strong experimental indications that the form of display of task related information, apart from its contents can have a significant effect upon the cognitive mode used and hence upon problem solving effectiveness (Ibid, pp. 6-7). One experiment was used by Piaget (1969) in which a five or six year old child is shown to be unable to understand the conservation of water when it is poured from one beaker into another of a different shape; the child, comparing the different iconic representations, believes the amount of water is different. Using the same experiment, Bruner showed that by shielding a view of the beakers by a screen, hence focusing attention on the flow of water as poured, children aged four to seven are able to recognize the principle of conservation in the transformation and realize that the amount of water in each beaker is the same. This recognition showed a high degree of transference to similar later problems without a screen for all but the youngest children.

Incidentally, Bruner is careful not to imply that the symbolic mode is the best form for solving all problems. In fact, he notes that maps and flow charts can be highly useful translations of symbolic representations back to iconic.

In the same vein, McKenney (1968) indicates that even mature adults may maintain the iconic mode as part of their "cognitive style." He points out evidence that suggests, in particular, that managers may tend to think in an iconic mode, whereas operations researchers think in a more symbolic mode; i.e., one's cognitive style may be in part related to role and to the type of problems typically faced in that role.

This theoretical and experimental work with variations in cognitive style and growth suggest the two dimensional space in Figure 3.2, which relates cognitive mode, problem type, and maturity. Within this chart, it should be noted that most experimental evidence deals only with simple, programmed tasks (i.e., the bottom of the chart) and with subjects who are young children. The hypothesis implicit in the chart is that even adult humans will tend to use an iconic or even enactive cognitive mode in approaching highly nonprogrammed problems. Also, it is not clear whether individuals have inherently different cognitive styles and pick their roles to fit the style or whether their primary cognitive mode is developed in response to the problem environment in which they operate (e.g., managers with less programmed problems than operations researchers). Nonetheless, with the assumption that variations in cognitive style do exist, it becomes important for a MMDS designer to anticipate such variations in the flexibility of his design and in its capability for evolution in cognitive style. The existence of such wide variations in individual problem solving style and strategy is underlined by the research of Hunt et al (1966).

Some of the on-line programming studies indicate not only a variety in performance, but also a substantial class of error-prone individuals who have an extremely difficult time mastering on-line tasks (Sackman, 1967, p. 63). Sackman suggests that systems for

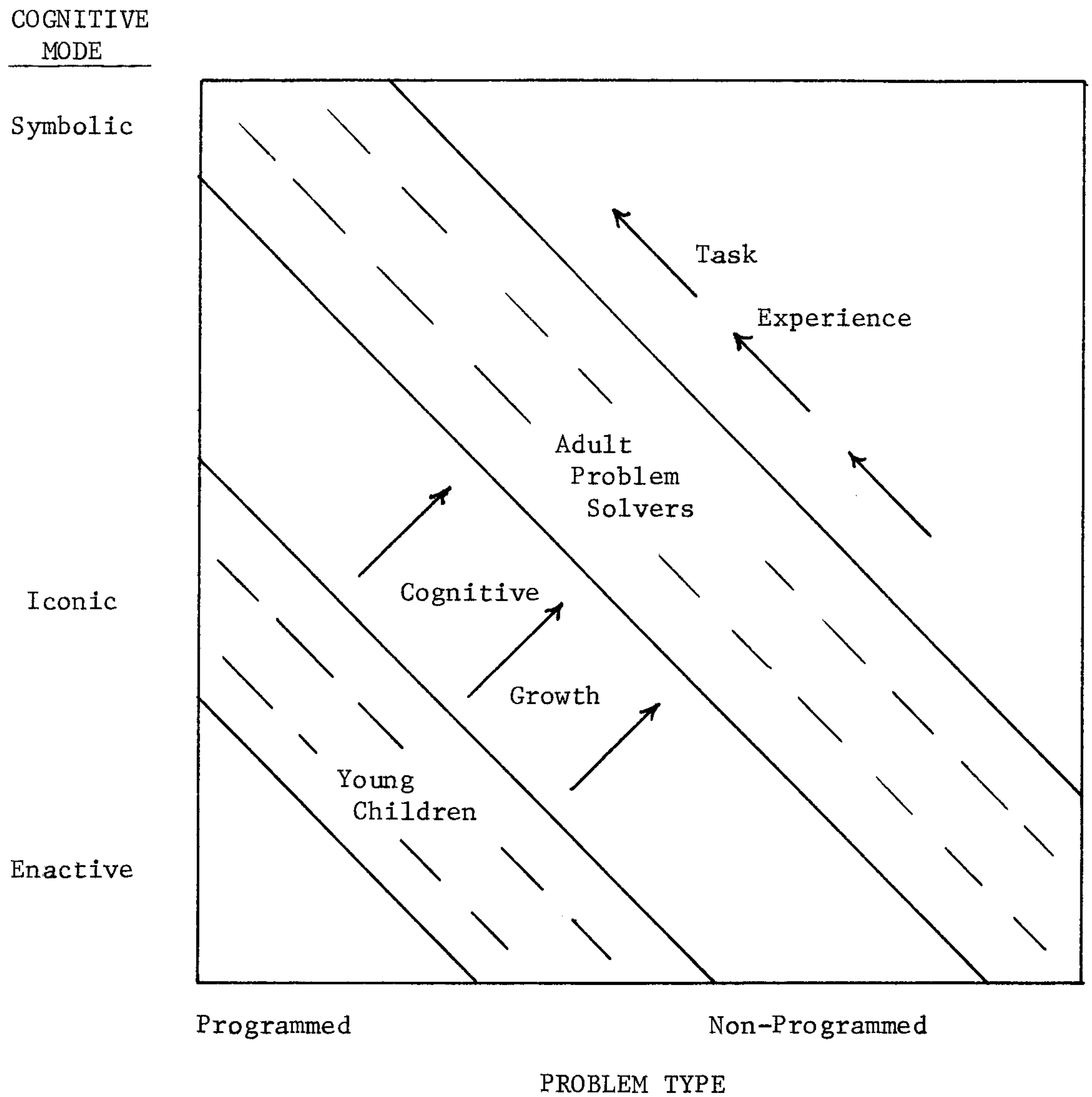


FIGURE 3.2

Cognitive Mode Vs. Problem Type for Problem Solvers of Varying Ages and Degree of Task Experience training MMDS users will have to be very versatile and highly individualized to accommodate both the slow learners and the high performers.

Note that there are two main trends over time suggested in Figure 3.2. One, the trend of cognitive growth is to shift problem-solving locus upward and to the right with increased maturity. Also, a more dynamic trend, upward and to the left, suggests the growth in individual task experience noted by McKenney as leading to both a more symbolic cognitive mode and also a more programmed view of the problem.

The work of Bruner et al (1956) in concept formation takes categorization as a fundamental activity in forming concepts, in building cognitive structure, allowing for complexity reduction, association, and prediction (Ibid, pp. 12-13). They hypothesize a general "cognitive need" which motivates humans to concept formation. A major contribution of their experiments is to indicate the nature of the strategies humans adopt in order to form concepts in the face of high informational complexity. The results are that highly local, focused search for the concepts prove more effective than more global, "rational" strategies, in light of human limitations in memory and information processing. As indicated earlier, Newman and Rogers (1966) have shown that programming operators to support such strategies in concept formation on an interactive computer system can increase effectiveness dramatically.

A field study which revealed the strongly evolutionary character of man-machine systems for unstructured decision making was that of

McKenney (1967) cited earlier. Also, Little (1970) has asserted the value of conversational systems and on-line models for user learning, arguing that such systems "are user-instructing and introduce a person to the issues of the problem and the model much faster than would otherwise be possible."

There may be an implication that on-line systems may be aids to cognitive growth in that they allow a decision maker to gain a "feel" for complex models without a necessity for understanding the detailed mathematics of the models. Particularly for a manager, or for any decision maker who is not usually inclined to think in abstract or symbolic fashion, this may be a great aid to increased understanding.

The results of Newman and Rogers (1966) were striking in that they showed substantial evidence that transfer of increased problem-solving ability was much stronger for a computer-aided group, both within and across levels of problem difficulty, than for the non-aided group. There were even indications that intellectual skills developed during interaction with the computer will carry over to non-aided tasks. In particular, the aided group showed significantly greater gains in performance on two standard intelligence tests administered before and after the experiment than did the non-aided groups. This would seem to support strongly Carroll's hypotheses about MMDM improving the problem-solving <u>process</u> as well as the solutions found, by making the man more explicitly aware of the nature of the process.

There is a question as to how quickly different decision makers can be expected to progress through the stages of cognitive growth with a MMDS. For example, Scott Morton observed that he seriously misjudged the time required to educate the decision makers in the full use of the device. Learning the mechanics of the device and how to control each of the available manipulative capabilities progressed very quickly--faster than anticipated, in fact. However, the natural blending of each individual capability into a smooth, coordinated decision process did not occur for much longer, requiring about two months (about 14 man-hours) of training sessions (Scott Morton, 1967, p. 212). Scott Morton used only test data on the system during these training sessions. He found later that when live data was used the managers needed little help in making the desired manipulations via the system.

Carroll asserts the value of a MMDS in a training capacity alone, apart from its operational use in decision making:

> There is a related effect in the training and education of new managers. In the world I have portrayed thus far, there are valid models ranging from the most detailed to the highly aggregated version of the environment. And given the interactive mode, the neophyte can "play" with the models and gain an unparalleled understanding for both the environment and operations of the firm. The facility for gaming is totally a by-product of the operating system.
>
> (Carroll, 1967, p. 362)

Licklider, in fact, in discussing the properties of a normative "composite time-sharing system," asserts that the computer system should assume much of the responsibility for this training function:

> One feature of this system is that it teaches the newcomer at his own pace, gives him all the coaching he asks for but also all the initiative, and makes it easy for him to recover from mistakes.
>
> (Licklider, 1965, p. 23)

Note, however, that the details of how one would design such an MMDS training mode are not clearly established and may represent a non-trivial extension of the basic decision making facilities.

One way to increase the ability of the MMDS to learn and improve may be to provide a decision control system; i.e., a formal system for monitoring decision system behavior, comparing it with a model of expected behavior (which may be nothing more than a collection of the MMDS designer's assumptions), and reporting gaps to decision makers in the system (for short run adaption) and to the MMDS designer (for long run modification). Given the highly evolutionary character expected of a MMDS, such a system should be especially valuable. Initial monitor capabilities representing steps toward such decision control systems have been demonstrated by Scherr (1965) and Elithorn and Telford (1969), and the value of such systems for psychological research has been asserted by Oettinger (1965).

Another design tactic to improve user learning and adaption to the MMDS is to make some of the functions and procedures of the system compatible with existing decision maker habits, operations, and forms. Licklider (1965) argues that such an approach will provide positive transfer from old ways to new. Then, from this secure base of familiar decision tools, the decision maker may gradually explore and adopt new and more complex procedures. This design tactic requires evaluation in practice.

Many writers mentioned above have cited the impact of the computer on human problem solving ability. Some have even suggested that man's concept of self and society may shift with the use of the computer, similar to the way that other human tools have assumed a larger meaning through their continued use. In this case, however, since the tool, the computer, can exhibit near-intelligent capabilities, the impact is expected to be even larger. As Ulric Neisser notes:

> (The computer) serves us not only as an instrument, but also as a metaphor, as a way of conceptualizing man and society. The notions that the brain is like a computer, that man is like a machine, that society is like a feedback system, all reflect the impact of cybernetics on our idea of human nature. This metaphoric status of the computer is closely bound up with its use as a tool. The goals we set out to achieve and the society we want to make depend on our idea of what men are really like.
>
> (Neisser, 1968, p. 206)

Lee reiterates this point:

> I don't think that we will simply learn to "accept" the computer--rather, I think that we will eventually develop a new and broadened conception of man himself, now that machines exist which he can use as an extension of his intellect.
>
> (Lee, 1968, p. 222)

Lee also suggests that today's somewhat apparent anxiety about computers on the part of some people will disappear as general familiarity with machines increases. He sees this borne out in his own research, and in the work of Neisser in 1964 at M.I.T.'s Project MAC:

> In the course of last summer, I interviewed over 60 of MAC's users. All had what might be called a "healthy" attitude toward the computer. None of them personified it; none of them feared it; for none did it play the architypical role of the dangerous machine. To be sure, many were angry at one or another feature of the system, and many--if not most--were occasionally frustrated by it. But their negative reactions seemed natural and appropriate.
>
> (Neisser, 1968, p. 216)

This result is counter to an idea prevalent in much of the writing about automation in general that the computer will encounter considerable resistance as it is applied to more and more functions now performed by humans. Whisler acknowledges this latter problem:

> The organizational stress produced by technological change, well known for a long time to students of industrial management, takes a new twist in the case of information technology. This twist results from the fact that for the first time, historically, those most affected by the change are also those responsible for initiating and planning it. Managers, seeking to make use of this new and powerful technology, must be able in an objective and deliberate fashion to consider the impact upon themselves, and to reorganize themselves as necessary. Understandably, those who see their own positions being threatened by a change will be reluctant to adopt the change.
>
> (Whisler, 1964, pp. 11-12)

One possibility is that there exists a certain exposure "threshold" to new technology and to interactive computer systems in particular--once over that threshold, anxiety is replaced by familiarity. The problem, then, is to manage educational resources so as to get potential users over this threshold in an effective manner. Such educational techniques should be a component of a normative methodology for the design of MMDS.

The general question of the impact of technology (and of computers in particular) upon society has received a great deal of study and speculation through history. The literature of this field has been surveyed by Heilbroner, who concludes that substantive progress toward answers has been minimal:

> Even the "simplest" of questions--the over-all impact of technology on employment and output--is still only uncertainly understood. Far less do we comprehend the effect of technology on "man" and still less again its enormous pressure on the moulding of society...our ignorance is not merely the result of the obduracy of the issues. It is symptomatic as well of a failure to mount a bold intellectual assault upon the problem itself.
>
> (Heilbroner, 1962, p. 25)

The work of Bruner in concept formation (Bruner et al, 1956) and in cognitive growth (Bruner, 1964) has suggested strongly that man tends to alter his perceptual and mental categories to fit and to discriminate within his immediate informational environment. This suggests two observations about MMDS:

1. the provision of richer set of procedures and data relevant to a problem via the machine should expand the complexity and discrimination of the man's problem concepts;

2. the _form_ of information presentation should be designed with care so as to avoid bias or rigidity as much as possible.

The first of these observations may be a basis for a testable hypothesis, given the measures of cognitive space dimensionality and complexity developed by Kelly (1955) and applied by Wilcox (1970) to a particular decision task. This observation is supported by the experiments of Newman and Rogers (1966), which show high transfer of learning from MMDM to decision making without machine aids, as well as by Gold's (1969) measures of greater "perception and understanding of the problem." It is also supported by the assertions of Neisser (1968), Lee (1968), and McLuhan (1965) as to the impact of tools or media upon information assimilation and concept formation. Finally, the observation is supported by research in the tolerance of complexity, which shows a trend towards higher and higher complexity tolerance in information displays with experience, a phenomenon also observed by Amstutz, (1967) in stockbroker use of displays.

Because of the wide variations in human cognitive style, problem solving processes, error-proneness, and programming abilities noted earlier (Patrick, 1967; Sackman, 1967; Hunt et al, 1966), the impact of a MMDS may be expected to be quite varied also. This may imply a requirement for considerable tuning of the system to fit each individual user, plus considerable flexibility in the machine system to accommodate different users at different times. Note, however, that one effect of a MMDS may be to reduce this decision performance variance in some situations, as observed in the analysis of Sackman and Gold (1968).

Experience in user learning of the mechanics of MMDS are also beginning to accumulate. There are indications that there may be several levels of understanding achieved, with the later stages considerably more difficult to attain than the earlier. There may be early stages of either (1) resistance to the change or to the new rigor of MMDM, or (2) over-fascination with the appearance of the system and its facilities. This may be followed by a stage where the user achieves some mechanical proficiency at using the machine, somewhat suggestive of the enactive stage of cognitive growth. Still later may come an appreciation of the forms or contents of individual operations within the system and some understanding of the full array of such facilities available--this may be analogous to the iconic stage. The iconic stage appears to have been relatively easy to reach with subjects in the few experiments or case studies conducted (Ferguson and Jones, 1969; Wilkins, 1968; McKenney, 1967; Scott Morton, 1967; Gerrity and Black, 1970). The next step in user adaption, however, may be much larger and longer. This is the step to the symbolic stage where the user maintains a working model of the MMDS that allows him to plan how its facilities may be blended into an on-going decision process--the potential combinatorial complexity of MMDS behavior makes this stage of the learning process difficult to achieve. It is important to recognize this fact and not to mistake the apparently facile use of the system of the "iconic stage" with this final step--on the other hand improved decision making may result from the iconic stage alone if the individual facilities and operators are carefully designed. The realization of the full power and potential of a MMDS, however, is liable to come very slowly through a gradual evolution into this symbolic stage of system understanding.

3.19 Decision Programming

Decision programming pertains to the evolutionary programming of operations or procedures in a decision process as they become better understood or more structured. It involves use of hierarchy in decision system structure in order to reduce complexity, as well as the building of decision procedure "macros" composed of primitive operations.

One assertion often made in the literature is that the use of an interactive computer system by a man in a MMDS helps to make the decision process itself more apparent to the man, thus improving his understanding. This is somewhat analogous to the oft-cited experience of computer systems analysts to the effect that the knowledge of current operations gained in the pre-computer analysis is often more valuable than any computer system that is eventually installed.

Whisler appears to be one of the first to suggest this notion:

> Thus an important effect of the new technology on decision-making is the simultaneous quantification and explication of decision-making processes.
>
> (Whisler, 1964, p. 6)

The point has been reiterated by Boehm (1967, pp. 15-16) and Carroll (1967). Whisler (1964, p. 9) has also warned, however, that such apparent increased visibility of the decision process may introduce risks of (1) overemphasis of the quantitative aspects of decision making, and (2) making the decision process overly rigid. Amstutz also asserts this tendency of the decision process toward more rigor as a result of man-machine interaction:

> As management gains experience in working with well organized and accessible data, they become increasingly interested in and prepared to use more advanced analytical procedures.
>
> (Amstutz, 1968, p. 7)

Certainly Amstutz's (1967) experience with stockbroker use of graphic displays indicates a strong tendency for the user to demand data representations that are more and more tightly coded. The initial displays in Amstutz's system were relatively sparse in content with explanatory table headings and the like. Over time, the user demanded tighter and tighter coding, with the end result being densely packed, almost totally numeric displays that are virtually unintelligible to the novice.

In the same vein, Carroll views a MMDS as a self-adaptive system which becomes more explicitly structured over time:

> As a corollary to the flexible division of decision labor and the feedback of results (real or simulated), the division can change over time. What I have in mind is a role of self-improvement, or more to the point, self-automation of the man. The man can be allotted his place in the system flow chart and recourse taken to his heuristic skills when occasion arises. As he accumulates experience, his treatment of certain problems may begin to follow a pattern, a clear-cut decision "protocol" may emerge. When it does (assuming that the reward system provides him with reasonable self-interest) the decision to program the protocol for the computer can be made. Given the feasibility, the automation becomes an economic question.
>
> (Carroll, 1967, p. 360)

In other words, what Whisler saw as a problem, Amstutz and Carroll view as an opportunity for MMDS--i.e., the evolutionary improvement and programming of the decision process. The hypothesis imbedded in these views has two components:

1. that continuing decision making experience with a MMDS will lead to greater understanding of the decision process than will similar experience without a computer; and
2. that greater understanding of the decision process on the part of the decision system will lead to improved decision behavior.

Over time, this greater understanding and "self-automation" is expected to lead to a gradual shift in decision resource utilization from man to machine.

The value of hierarchy in language and for problem-solving has been asserted by many (e.g., Simon, 1969). It is called variously, "simplication" (Oettinger, 1965, p. 7) or "abstraction" (Newell, et al, 1959), and it is implicit in the effective transition from "part" to "whole" learning with increased experience and understanding (Sackman and Gold, 1968). Hierarchy also appears in the very powerful notion of a "meta plan" in human problem solving (Miller et al, 1960), or the "macromove" of Newell (1966, p. 28). Such discussion suggests the value of a capability for building "macros" of existing MMDS functions directly at the console. In other words, the decision maker might benefit from having direct control over the process of decision programming, and over the building of a hierarchy of programmed procedures.

The advantages of incorporating explicit programmed hierarchical forms should be emphasized. Given man's limited span of attention and tendency to become locked into localized "sets" in problem solving, the costs of long clerical digressions within problem solving can be high in terms of the decision maker's loss of problem overview and perspective. The ability to program such functions, transfer them to the machine, and then view them as a "substability" (Zannetos, 1966) from a higher level in the decision making hierarchy can greatly improve the decision maker's ability to structure and understand a complex process in toto. It can, as a consequence, improve the decision system's ability to change and to improve:

> ...complex systems will evolve from simple systems much more rapidly if there are stable intermediate forms than if there are not. The resulting complex forms in the former case will be hierarchic. We have only to turn the argument around to explain the observed predominance of hierarchies among the complex systems nature presents to us.
>
> (Simon, 1969, p. 98)

Hierarchy is evident in computer programs, in organizations, and especially in human problem solving behavior, where it aids not only the organization of processes but also the comprehension of complex systems or problems (ibid, p. 106). These observations suggest the value of using the notion of hierarchy explicitly in the following areas in the design of a MMDS: (1) in the organization of the conversational language; (2) in the models available in the system (e.g., in the varying degrees of "resolution" of the models) (3) in the evolution of system tasks and descriptions over time (Emery, 1965, pp. 246-48); (4) in the resolution levels of data or graphic displays available. For example, as usage of a particular sequence of operators increases over time, the MMDS should evolve a more concise code or command for that sequence. This is analogous to the way natural language evolves to follow Zipf's Law, with word length being inversely proportional to usage frequency, but at a much faster rate. This evolution can also allow the individual user to tailor the MMDS to his own particular procedural idiosyncracies, although the functional base of operators would be the same from user to user. In addition, more responsive and powerful control over the decision process via this decision programming capability helps in "learning" the process by helping to develop the decision maker's own decision process model.

3.20 Summary

This chapter began with a survey of literature related to the field of MMDS. The early prevalence of a machine-centered or a data-centered view of man-machine systems was noted, as was the beginning development of a decision-centered view. The need for a theory of MMDS structured about models of decision making was asserted.

Next, several dimensions of a decision task were identified, and several models for decision system behavior and content were surveyed. From this base of theory a general decision system framework was synthesized which included phases of the decision process, fundamental decision system components, characteristics of decision system interaction (particularly relevant to MMDS), and decision system adaptive processes. It was argued that this framework was complete in the sense of covering all reasonably significant elements of decision system content and behavior, and that it would be useful to the MMDS researcher and designer.

The elements of the general framework were then taken individually and elaborated. Theory and evidence relevant to each element was reviewed and some detailed implications for expected MMDS behavior and design were drawn. In the case of the decision process phases of Intelligence, Design, and Choice, some specific functions that should be a part of each phase were identified. The bases of existing research relevant to the MMDS framework were discussed in some detail here since, to this writer's knowledge, this integration has never been carried out before. In addition, this decision system framework is viewed as an essential component of the MMDS design methodology proposed in the next chapter.

CHAPTER 4

THE DESIGN OF MAN-MACHINE DECISION SYSTEMS

4.1 Introduction

This chapter will introduce a methodology for the design of MMDS. The methodology will rely heavily on the general decision system framework outlined in the previous chapter.

The chapter begins with a brief survey of the literature of information systems and MMDS design. Methodology for the design of interactive man-machine systems for unstructured decision-making is quite undeveloped, despite the published multitude of assertions of their great potential for improved decision performance. The pressing need for such a methodology is clear if one notes the rapid growth of time-sharing and interactive computer systems over the past several years. Such systems are obviously being used. It is not at all obvious that their use is planned or controlled intelligently, especially in their application to unstructured decision problems.

Based on the special characteristics of MMDS identified in the previous chapter, this chapter proposes several principles of MMDS design. It then uses these principles to guide the elaboration of a MMDS design methodology. This methodology is distinct from other information systems design approaches discussed in the literature in its emphasis upon (1) the early phases of analysis and general design,

(2) a decision-centered view of the system, and (3) explicit control on the design performance based on original designer's assumptions and expectations. On the other hand, there is very little discussion here of the specific techniques and technology of detailed design, programming, and implementation, since these are amply treated in the literature.

4.2 Issues in the Design of Information Systems

Information system design techniques were reviewed by Davis in 1964. She observed that information systems design was not yet a profession, where:

> ...the crucial criterion of a profession is the existence of a systematic body of knowledge of substantial intellectual content and the development of personal skill in the conscious application of this knowledge to specific cases.
>
> (Davis, 1964, p. 78)

She concluded that information system design has not yet achieved much coherence but seems instead to be a loose collection of techniques whose derivation cuts across many professions and practically all of science.

Kreibel (1967) surveys the use of operations research techniques in the design of management information system (MIS). The net impression is that, even in this relatively restricted area, a coherent body of design knowledge has yet to emerge.

In the plethora of more recent books on information systems design, however, some coherence in approach is evidenced, although the emphasis is on the more programmed phases of the design process. This recent work tends to focus on the technical "building blocks" of the system -- the alternative hardware and software system components -- almost to the complete exclusion of (1) the human components of the system and (2) the specific problem characteristics. They also tend to focus on the later stages of design implementation and "fine tuning," largely neglecting the early phases in the process of problem definition and general design of solution alternatives, but rather just assuming their completion (see, e.g., Martin, 1967; Stimler, 1969).

Those texts that do discuss early phases of design tend to dwell on the traditional techniques of descriptive modeling of information systems (Boutell, 1968; Glans et al., 1968; Gregory and Van Horne, 1963). These involve interview, forms collection, data file analysis, processing volume analysis, and documentation, using such representational devices as flow charts and decision tables.

Many of these texts also identify the traditional design phases: (1) describe the current system, (2) develop system requirements, (3) design a new system, (4) implement (e.g., Glans et al., 1968; Optner, 1968). Gregory and Van Horne (1963, pp. 200-205) go a step further and outline several major design approaches available to the systems designer:

1) Simplification - just simplify the existing system, eliminating redundancy in files and operations (the "economy approach").

2) Mechanization - simply mechanize some of the currently manual operations within the same structure (the "efficiency approach").

3) Data System Redesign - this involves a restructuring of the data inputs and processing in a significant way, without introducing major changes in the output to the user. This requires more freedom of action be given the designer.

4) Management System Design - this is the normative method (the "optimal approach"), allowing for essentially complete freedom to overhaul or replace the current management information and decision system.

Although the fourth alternative of management system design is noted, neither this text nor others face the detailed issues of how it is to be accomplished. Most attention (and indeed, most design activity) is focused upon the first three alternative approaches.

Blumenthal's work (1969) represents one of the most ambitious approaches to the theory of information systems design:

> This entire book is an affirmation of the belief that defining and solving business systems problems can itself be a systematic process. And not only a systematic process, but also a scientific one, in the sense that a hypothesis-experiment-confirmation approach to systems definition and development is possible to a large degree.
>
> (Ibid., p. 14)

Blumenthal, in contrast to other writers, does recognize the importance of the early design phase of analysis and general design and their interaction with later phases. He recognizes clearly the need for mechanisms to allow for transformation back and forth from a functional, goal-oriented view of an information system to a hardware-software technology-oriented view. He asserts that the functional requirements should be user-oriented in form and should define the job to be done without particular concern for the detailed methods of implementation. He sees them as representing the interface between the users and the systems group. The iterative translation back and forth is necessary because the requirements placed on a system by the goals and characteristics of its users are most easily expressed in "functional" (or problem-oriented) form, whereas the issues and constraints of technological cost and feasibility are most easily related to a hardware-oriented system model. These goals and constraints must interact to some extent -- hence iteration occurs.

The need for such mechanisms to enhance communication between ultimate systems users and systems designers have been cited by many. Zannetos states this need, and a requirement for flexibility in MIS as well:

> Once the system is designed and installed, the damage is done. It is not easy to dismiss it as you can dismiss a payroll clerk. The inflexibility and specialization of data structures and software packages is enormous. The only way to guarantee that the system will effectively serve you as managers in your planning and control activities and serve as an extension of your intelligence is by influencing its design.
>
> (Zannetos, 1967, p. 18)

The issue remains as to what mechanisms will help to realize this user participation without necessitating that he become an information systems expert, hence doing all the transformation between functional and hardware-oriented models implicitly in his head (although many would argue that a few steps in this direction would be a good thing).

Based on his experience and observations of MIS, Amstutz (1968, p. 5) notes that the following are the four consistent characteristics of "successful systems":

1. The system is founded on management's conception of the decision environment.
2. The user-manager understands the system structure.
3. The system is based on disaggregated data files.
4. System development has proceeded to increasing levels of sophistication through a process of gradual evolution.

(Amstutz, 1968, p. 5)

In particular, although Amstutz recognizes the problems of user involvement -- the high cost of time and the "threat of having to make one's decision assumptions explicit -- he accepts those as necessary costs, asserting that it must be done. He also recognizes the evolutionary character of an MIS and its need to reflect the user's view of the problem if it is to be incorporated into his decision process.

Characteristic Approaches to Information Systems Design

In surveying this literature of information systems design, as well as by observing many actual computer systems applications, this investigator began to discern the following general viewpoints toward design:

(1) Machine-centered design. This is characterized by utmost concern on the part of the designer with hardware system efficiency, throughput, and capacity utilization, as well as by a certain air of isolation (if not insulation) from user problems.

(2) Process-centered design. This approach is close to the "mechanization" alternative of Gregory and Van Horne. It is characterized by a factoring of an existing process into small modules, then programming these pieces one by one (where one "piece" might be a

roomful of clerks performing a defined processing function). The oft-observed result of this approach is the transfer of a fundamentally manual system first to a card system, then to a tape system, and eventually to a disk or on-line system, with no real thought given to overall system redesign in any of these steps. It is granted that each step may involve increases in efficiency, but the approach ignores the full potential offered by the changing information technology.

(3) Data-centered design. The philosophy of this approach is often called "total systems." It manifests itself, in practice, in the design of systems from the "bottom up," starting with the data base. The notion seems to be that if one starts with all of the operational control data in an organization and builds applications up from there, then one will eventually arrive at the integrated system which will serve the decision needs of all, including the top managers. One special manifestation is the design that is driven by the availability of a large machine-readable data base, rather than by carefully defined problem-oriented requirements.

(4) Information system design. This is basically the "data system design" of Gregory and Van Horne, which involves some system restructuring but producing essentially the same outputs as before. In other words, its primary concern is with efficiency rather than effectiveness of the processing structure.

(5) Decision-centered design. The philosophy of this approach is that the value of information derives from the decision functions which it supports, so the requirements are developed from an analysis of the decision system.

It should be emphasized that the above represent characteristics or philosophies of design, not clearly distinct methodologies. Nonetheless, the author's experience with information systems applications is that most fall into the first three categories, very few in the fourth, and practically none in the last.

Ackoff, in his article on "management misinformation systems" (1967), argues strongly for replacing traditional data-centered or machine-centered views of MIS design with a decision-centered approach. He begins by criticizing some typical assumptions in MIS design and then goes on to propose his own framework for design:

1) Analysis of the Decision System.

2) Analysis of Information Requirements.

3) Aggregation of Decisions.

4) Design of Information Processing.

5) Design of Control of the Management Control System.

This framework is unique in several aspects. One, it places decision system analysis foremost in the steps, with information requirements following as a consequence. It also interjects a third step of decision system redesign and simplification which restructures decision responsibility according to data and decision process commonalities, eliminating much redundancy in the process. It leaves actual technology-oriented system design, the point where most of the "cookbook" texts begin, until quite late in the process. Finally, it places emphasis on an explicit control function as absolutely essential in such a constantly evolving system. Ackoff also asserts, with regard to this last point, that a man-machine interactive system is most attractive because of the additional flexibility it implies: "No completely computerized system can be as flexible and adaptive as can a man-machine system...whose precision and generality is continuously increasing with use" (ibid., p. 155).

Ackoff seems to recommend taking a "total systems" approach to design of the decision system. The experience of Zannetos (1967), however, in the evolutionary implementation of "intelligent" information systems, is that this total approach is doomed to failure given current resources. He recommends instead taking a more bounded portion of the current system, improving it, taking care that such marginal changes do not preclude integration with a total intelligent system in the future (ibid., p. 26).

4.3 Principles of MMDS Design

This section will begin by noting some of the characteristics of MMDS which distinguish them from information systems in general, and which therefore imply different design tactics. Then a set of general principles for MMDS design will be proposed and elaborated point by point.

The design and evaluation of highly interactive man-machine decision systems is inherently different from that of non-interactive computer systems. The latter tend to be aimed at the performance of highly structured, operational processing tasks, and their impact and effectiveness are relatively clearly defined in terms of such tasks. For example, the computer processing of payroll every week has primary payoff in the increased efficiency of task performance, where the inputs and outputs are quite well defined and possess a predictable and easily modeled relationship. For a MMDS, on the other hand, system effectiveness derives from the performance of the decision maker-user, from changes in the structure and content of his decision process, and these are relatively less predictable and measureable effects. Since the behavior of such man-machine decision processes is little understood, as Chapter 3 has suggested, traditional computer system design has focused on operational task objectives and has either ignored such decision-making effects or paid lip service to them as 'intangible benefits'. In fact, this traditional approach may be quite reasonable, given that the most significant expected benefits of the proposed system lie in its more efficient performance of well-defined operational tasks, and the apparent interaction with decision makers is quite indirect and slow. The primary benefits of MMDS in a non-programmed decision-making situation, on the other hand, must lie in this presently "intangible" realm, where traditional design methodologies are of limited usefulness without an explicit model of man-machine decision behavior. These "intangibles" and unpredictable characteristics of MMDS behavior and evolution imply several principles for design, which are outlined next.

Outline of Principles for MMDS Design

Chapter 2 has developed the structure of a general methodology for design. What is required now is a set of principles for design of MMDS in particular, which will serve as guides in modifying the general design methodology to fit this specific design task.

One characteristic of MMDS that should have become clear in Chapter 3 is that we have only a very limited understanding of their behavior. Another characteristic is that the MMDS is designed to serve a decision maker, whose acceptance of the system is a key to success. The combination of these two characteristics suggests an evolutionary (as opposed to revolutionary) approach to design that maintains high user consideration and involvement throughout. Therefore one principle of MMDS design is that the design should include a plan for careful evolution of MMDS capabilities from the existing base of decision functions. Although the system may provide the opportunity for eventual radical change in the decision process, it should also allow for the user to <u>start</u> with an array of familiar functions and data files and learn to use additional or new functions at his own reasonable pace.

A second principle of design implied above is that the methodology should place heavy emphasis on mechanisms for keeping ultimate users involved, aware, and contributing to the design process. This has been shown to be important with any information systems effort (e.g., Garrity, 1963), but it is asserted here to be even more critical in the design of highly interactive MMDS. The reason is that the MMDS is a much more intimate and personal tool of the decision maker than less interactive systems -- therefore, he will be much more sensitive to its detailed characteristics and hence must participate in design.

A third and related principle of MMDS design is that explicit decision models are essential to the design process to maintain our objective of a decision-centered approach. There is a requirement for frameworks or general models to aid in development of descriptive and normative models of decision making in general and of MMDM in particular. Further, if the design tactic is going to be evolutionary and adaptable to the human user, it is then asserted that the appropriate decision framework is one patterned after human decision behavior.

A fourth principle is that MMDS design should not aim just at providing desirable data for the decision maker but rather should provide manipulation capabilities and operators as well. These programmed operators should relate to the requirements discovered in the decision modeling phase -- that is, some will be recognized in the descriptive decision model as operations that the human was already performing before, others as new operations suggested by the normative modeling effort.

Given our acknowledged lack of complete understanding of MMDS, a key to successful design is having adequate representations of MMDS "design space" and mechanisms for search within that space. In other words, although we espouse a careful evolutionary approach, we consider wide search for possible designs to be critical to long-run progress. Thus a fifth principle is that a MMDS designer should employ design search and generation mechanisms that encourage broad considerations of alternative possible designs.

A seventh principle, also related to the expected pressure for growth inherent in an MMDS, is that the design should incorporate all reasonable facilities to increase system flexibility in structure, mode, language, and form of interaction. This clearly has implications for the systems software architecture, as well as the man-machine interface.

An eighth principle is that a formal decision control system should be designed as one component in the MMDS. Such a control mechanism is attractive in most designs, of course, but it is even more essential in the design of MMDS due to the high unpredictability and rate of change of such systems.

Finally, given a lack of understanding of MMDS design that is at least as great as our lack of understanding of MMDS behavior, explicit control on the design process itself would be valuable. This need for decision process control will not be stated as a design principle, but it remains implicit in the exercise here of developing an explicit description of a MMDS design methodology with associated heuristic tools.

In summary, the following principles of MMDS design have been enunciated:

1. MMDS design should provide for evolutionary change of the decision system.

2. Ultimate users should be kept involved, aware, and contributing to the MMDS design process.
3. Explicit decision models, based on human decision behavior, should be utilized in the design process.
4. The MMDS design should aim at providing accessible programmed operators for manipulation of data and models in the system.
5. The design methodology should employ explicit design search and generation mechanisms that encourage broad consideration of alternative designs.
6. The design methodology should employ explicit mechanisms and models for predicting ultimate MMDS behavior.
7. The MMDS design should aim for all reasonable flexibility in structure, mode, language, and form of interaction.
8. An explicit decision control system should be designed as one component of the MMDS.

This section has merely introduced these principles of MMDS design with little logical support. The following sections will elaborate on the reasons behind the principles and indicate some of the ways they may be realized in the methodology.

Evolutionary Design Tactic

The proposed approach here is to design a MMDS which represents a compromise between the current decision system and a normative decision process. This approach recognizes a necessity for the system to accommodate itself to the decision maker to an extent such that he will use it. As Little (1970) has noted, this may result in a system that is different from what a management scientist might otherwise build, but it should be a system which is workable and can be blended into the user's decision process. Soelberg's research into non-programmed decision-making supports this approach, as he indicates in the following statement:

> A manager should work on integrating formal models of rational decision making with his intuitive, judgmental, common sense manner of solving choice problems and seek to adapt the former to fit the latter, rather than submit to a bastardization of his intuition in the name of some modern mathematical technique. Mechanical aids to management decision, like computer-based management information systems, will (and should) be resisted to the extent that their structure is incompatible either with the manner in which a manager codes relevant information for his own use, or the manner in which the manager intuitively feels that information should be reduced for arriving at a decision.
>
> (Soelberg, 1967-B, p. 28)

This is not to say that the MMDS designer should forego a goal of improvement of the decision process, but rather that he should provide a familiar base from which to provide improvement.

One approach to building a familiar base would be to identify programmable operations or data manipulations in what the decision maker now does and implement those in an accessible fashion with the appropriate data base. These familiar operations provide for positive transfer from the current decision process for the user. Once he is over this threshold of using the system, then he may begin to explore and incorporate new procedures implemented from a normative rather than descriptive model base.

In other words, the approach is specifically not aimed at optimization of a narrowly defined decision system, but rather at the support and programming of heuristics and operators that appear useful to the human decision maker. It is recognized that this is basically a conservative design approach and that there are more aggressive alternatives, such as totally programming the function and eliminating the man, imposing a totally new decision system on the man, or replacing the man with another more amenable to substantial decision process change. In fact, the evolutionary approach proposed here may allow for radical change of the decision process in the long run, but at a pace of evolution amenable to the decision maker and his capacity for adaptation. At this stage of our understanding and appreciation of MMDS, it is felt that there is substantially more to lose than to gain in the operational application of very aggressive and risky approaches to basically unstructured decisions.

User Awareness and Involvement

One of the assumptions that Ackoff (1967) attacked in his now classic paper on "misinformation systems" was the assumption that "a manager does not have to understand how an information system works, only how to use it." Ackoff's criticism is that the manager must understand his information system in some depth if he is to fulfill his responsibility to evaluate it. However, one can go even further. This assumption is often justified with singularly bad analogies of the form: "a manager does not have to understand the telephone (e.g., electromagnetic wave theory) in order to use it." The reply is that the telephone is neither so complex, expensive, nor subject to such constant pressure for change as is an MIS. Certainly the decision maker need not understand computers at the electromagnetic level either, but he should understand information and decision technology to an extent such that he can evaluate, control, and contribute to the design of his own system. Without such understanding, costly as it may be to facilitate, the decision maker is by default allowing important constraints on his decision behavior to be set by information technologists, enlightened as they may be. The need is for mechanisms -- people, languages, models -- to aid in this communication.

Several such mechanisms are suggested later in the discussion of the design methodology. These mechanisms basically are forms of representation which allow for the user to visualize concisely the developing MMDS design. In particular, a matrix representation is developed which allows a simple overview of the potential interaction between data and operators in the system, and a conversational graph is proposed which allows for very detailed mental simulation of the mechanics and behavior of the system. Beyond this, early prototype demonstration is found to be a great stimulus to user involvement and contribution.

Decision Modeling

Earlier sections have identified what has been called "machine-centered," "data-centered," and "process-centered" views of information systems design. It is asserted here that the design of MMDS, at least, should be decision-oriented instead. What this means is that the primary focus of analysis and design should be upon the particular set of decision processes of concern and that explicit normative and descriptive models of decision making play a major role in the process. Since the ultimate decision maker in a MMDS as defined here is a human, the form of these decision models should be representative of human decision making behavior and capabilities.

This explicit modeling process is felt to be essential to recognition of new decision or data needs. This notion is suggested by Ackoff's criticism of another erroneous design assumption, that "the manager needs the information that he wants." Ackoff's response is that for the manager to <u>know</u> what information he needs, he must understand the array of decisions he must make and have a reasonable model of each, a condition which Ackoff asserts is seldom satisfied.

The typical decision maker will be uncertain as to his needs and hence will ask for too much, compounding the problem by causing an overabundance of irrelevant information. It is possible to go even further to assert that the manager cannot say what information he needs unless he is also in a position to say what new decision processing capabilities he wants. In other words, he needs an ability to visualize alternative decision procedures, as well as the one he currently uses (which is likely to be largely a function of his own information processing limitations and those of his environment). It is asserted that a decision maker actively involved in the process is unlikely to visualize such alternative procedures without an independent normative modeling effort.

Nonetheless, in keeping with the evolutionary approach, it is proposed that the form of the normative as well as the descriptive decision model follow the form of a model of human decision behavior. Hence the decision system framework of Chapter 3, based on the models of Newell, Shaw, and Simon, is adopted in the design methodology as the decision modeling format. By keeping such human decision system models dominant throughout the design process, the designer helps to insure that the ultimate user will be able to integrate the resulting system smoothly into his decision processes.

Programmed Operators

Another assumption attacked by Ackoff is that "the critical deficiency under which most managers operate is the lack of relevant information." Ackoff counters that they may suffer more from an over abundance of irrelevant information; i.e., the so-called MIS's are deluging the decision maker with too much data. This is a characteristic result of the "data-centered" approach to design. Too often have MMDS designs appeared to be "driven by a data base" (e.g., see an on-line security analysis system design presented by Gal, 1966), or by hypnotic attraction of a particular hardware innovation, and have failed to recognize their end aim of supporting decision making.

One implication for systems design is that the decision maker should be given access to procedures and operations as well as data, and that techniques for simplification like (1) exception reporting, (2) graphical display, (3) filtration, (4) aggregation, (5) flexible report formatting, and the like will be valuable to the decision maker. There is a danger, of course, of overcondensation of available data, but this may be avoided as long as the decision maker has easy and active control over the depth of his search.

Chapter 3 has identified general classes of fundamental operators that might be useful in MMDS. One of the most powerful of these is the comparison operation, which indeed seems ubiquitous in human decision behavior. Its importance is asserted by Little

(1970) and observed by Scott Morton (1967). The MMDS designer should consider whether he has provided desirable mechanisms for comparison of (1) status to goal or standard, (2) alternative solution to status, and (3) alternative solution to goal.

Design Search and Generation

One general implication of the unpredictability and high rate of change of a MMDS is that the designer should beware of assuming a static situation and thereby just programming parts of the previous decision process. Thus, although the approach recommended here is conservative and evolutionary, it is not static by any means. The MMDS designer should make every effort to realize his full potential as a change agent, broaden the bounds of his system search to include normative decision considerations and to anticipate the change as best he can. Baker, in fact, suggests that one of the dangers in design of a MMDS is that the designer will assume that too much is "given," that the decision system will change little from the previous batch or manual system:

> Thus, a solution in one system had become a requirement in another, even though no such requirement had been established in terms of that system's specific needs.
>
> (Baker, 1964, p. 430)

Similarly, the designer must anticipate that one result of MMDS evolution will be a pressure to change the human organization, which, in fact, developed in part in accommodation to existing information and decision system characteristics. As Scott Morton notes,

> If we are going to make really effective use of interactive terminals we are going to have to restructure our thinking about the way decision making gets done in an organization and the kind of organizational structure that will be most useful for this.
>
> (Scott Morton, 1968, p. 29)

Galbraith (1969) offers a conceptual point of view for thinking about this interaction of task characteristics, information system, and organization structure.

A methodology for MMDS design must be very concerned with search processes. The possible solution design alternatives are distributed in a huge and largely uncharted alternative space. Hence, unlike information system design for more programmed problems, design of MMDS requires relatively larger allocation of design resources to global search as opposed to very local design "tuning," which can be a disastrous waste of resources if indulged in too early in the process. In traditional terms, the design of a MMDS demands more effort in the early phases of systems analysis and general design. One aid to such design search would be useful representations of the design problem or search space. As Newell has noted and Simon reiterates:

> ...solving a problem simply means representing it so as to make the solution transparent. If the problem solving could actually be organized in these terms, the issue of representation would indeed become central. But even if it cannot -- if this is too exaggerated a view -- a deeper understanding how representations are created and how they contribute to the solution of problems will become an essential component in the future theory of design.
>
> (Simon, 1969, p. 77)

One such representation which will be introduced in the methodology is a matrix display of potential operator and data type intersections. Earlier we have seen that another useful design generation mechanism is the normative-descriptive model comparison. Also, the list of MMDS operator classes in Chapter 3 is a rudimentary aid to design search. The general decision system framework itself is meant to be a stimulus to creative design thought.

An aid to the very early stages of decision system analysis and problem recognition would be formal aids for mapping from decision task characteristics to appropriate problem solving structures or systems. Sackman and Gold, (1968), asserted the need for a taxonomy of problems so that the nature of this mapping could begin to be explored empirically. Scott Morton's (1967) approach is to outline the attributes of a problem environment that could effectively use an MMDS. These attributes are: (1) large data base, (2) high volume of data manipulation required, (3) subjective analysis required at stages between manipualtion, (4) subjective evaluation required to select alternative solutions, (5) existence of very complex relationships between the variables relevant to the problem solution, (6) high multidimensionality of the problem and its solutions. Of further use would be an explicit set of archetypes of man-machine systems, the other side of such a mapping. Several such MMDS archetypes have been suggested by Miller (1965 and 1969).

Predictive MMDS Models

Another design assumption attacked by Ackoff is that "if one gives a manager the information he needs, his decision making will improve." The implication in Ackoff's criticism of this assumption is that data access without procedural power is often not enough. Without getting entrapped in the variety of possible interpretations for "the information he needs," however, the general problem remains that our understanding of the interaction of decision-maker, data, and operators is highly limited. Yet the MMDS designer must have some basis for developing assumptions or expectations of future system behavior in order to make detailed design decisions and tradeoffs. The survey of MMDS and psychological research and theory of Chapter 3 was meant to serve as one such basis for developing expectations or predictive models of MMDS.

This unpredictability of MMDS behavior also suggests the usefulness of investment in early prototype systems or "simulators" to get direct indicators of expected operational system behavior. This approach is strongly urged by Bell (1968, p. 111), based on his experiences with the unpredictable impact of graphic displays (e.g., only half of his original major design assumptions proved valid). The prototype could take the form of a very limited data base with a skeletal conversational language, tested on an experimental basis by a handful of subjects. Though the cost of such a prototype will seldom be trivial, it is a reasonable hypothesis that the value of such an exercise early in the design process increases significantly with (1) the unpredictability of operational system behavior, (2) expected rate of evolution of the operational system, and (3) cost of the operational system. All three independent variables in this hypothesis may be expected to be large for a MMDS -- hence the value of a prototype.

A man-machine simulator is a variant of the prototype approach with the objective of making the man-machine dialogue "feel" be as realistic as possible for subjects, but with no necessary development of directly relevant prototypic software or hardware behind it. At the simplest extreme, this means hand preparation and review of a reasonable scenario of pseudo-machine displays through which the user can sequence. At another extreme, it may involve a computer-based simulator closer to a prototype. The conversational graph representation of the design introduced in the methodology can aid in simple scenario generation.

Flexibility of the MMDS Design

The highly evolutionary and dynamic character of MMDS has been noted. Thus the design must anticipate that the MMDS development process will be open-ended -- i.e., the system will generate its own internal pressures to grow, to learn, to evolve. For the user, the very existence of new and explicit interactive computer capabilities will serve as a stimulus to search both for new capabilities not yet provided and also for unexpected ways to utilize the existing capabilities. Both the design itself as well as the design methodology must be prepared to adapt flexibly to these evolutionary pressures if the MMDS is to remain viable.

Also the methodology often must anticipate a diversity of talent and performance of individual users at any given period, as well as over time. This would imply a strategy of design whereby a "portfolio" of capabilities is provided, which ranges from the very sophisticated to the very primitive, as well as covering various problem-solving styles and vocabularies.

Despite such a need for flexibility, however, it should be emphasized that the flexibility must be bounded and logically conceived. That is, an asserted need for flexibility can provide no excuse for lack of early and careful explication of MMDS goals, behavioral assumptions, and their consequent design decisions. Beyond the sheer cost of providing more flexibility, there is also a trade off between flexibility and over-complexity in system impact on the user. Some evidence of the inhibiting effects of too much complexity in a MMDS is noted in the experiments of Bell (1968) and Gerrity and Black (1970).

The need for general purpose, flexible and modular MMDS software architecture has been asserted with regularity in the literature (e.g., Scott Morton, 1970). These goals may be rephrased as the need for a system structure that will accommodate the majority of individual functions that are likely to be required in the decision process and that will allow for easy reconfiguration as the MMDS attempts to evolve. Herein lies a conflict with the earlier expressed need to <u>focus</u> on a particular decision area to design a MMDS: in order to achieve generality, one would like to identify a set of general functions and elements in the decision making process and to structure the software system so as to support such general functions. With scarce resources, the best course is likely a compromise. In other words, the focus should remain localized in design, but an explicit attempt should be to recognize commonalities in function and form across local MMDS's, so that the approach to design may achieve more generality with experience. This generalizing process can be aided in part simply by being attuned to recognize <u>general</u> problem solving processes (through a familiarity with models of decision making) in specific situations and then by implementing those that are to be programmed in as general a form as is reasonable.

Beyond such desired flexibility in architecture and function, there is also a need for flexibility in language and form of man-machine interaction. The prototype MMDS described in Chapter 5 incorporates several approaches to such language flexibility which have the added advantage of keeping more complex (and more powerful) capabilities relatively invisible to the beginning user.

Decision Control System

The need for explicit control mechanisms for MMDS has been expressed by many who view an MMDS as a self-adaptive system. For Sackman (1968, p. 53) this control system is, in fact, the heart of his proposed "general theory of man-machine digital systems", which emphasizes the experimental self-analysis, hypothesis testing and adaption of the MMDS at an evolutionary tempo that must inevitably be matched to the pace of man (in some ways, one of the slower changing mechanisms in the system). Also, the fallout benefits to psychological research of such control mechanisms for intimate monitoring of the steps of the decision process have been noted by many, not to mention their value in the process of shifting more and more decision function responsibility to the machine as the process becomes continually clearer and better structured.

Some of the issues in design and use of such a formal decision control system are raised in the following discussion of the design methodology. One realization of such a system for a prototype MMDS is described in Chapters 5 and 6.

4.4 An MMDS Design Methodology

Based on the MMDS design principles described in the previous section and the general design methodology of Chapter 2, this section develops a proposed methodology for the design of MMDS. The general structure of the methodology will be described first, and following sections will elaborate each phase, as well as some specific heuristic tools for the process. The discussion of the methodology will be rather abstract in this chapter; thus the reader may wish to refer occasionally to Chapter 5 for a detailed application of the methodology.

Structure of the MMDS Design Methodology

The general structure of the proposed MMDS design methodology is shown in Figure 4.1. It is an elaboration of the general design methodology of Chapter 2 (see Figure 2.1), with the addition of Functional Model and Decision Control System Design stages. As before, the lines connecting one stage with another are not meant to connote a rigid, unidirectional progression through a set of clearly distinct phases. On the contrary, the actual design process is expected to be highly iterative, with the lines merely indicating particular interactions between phases that are expected to be especially strong. For example, there might ordinarily be a number of iterations through normative modeling, system bounding, descriptive modeling, etc., before the focus of design attention shifts toward functional modeling and

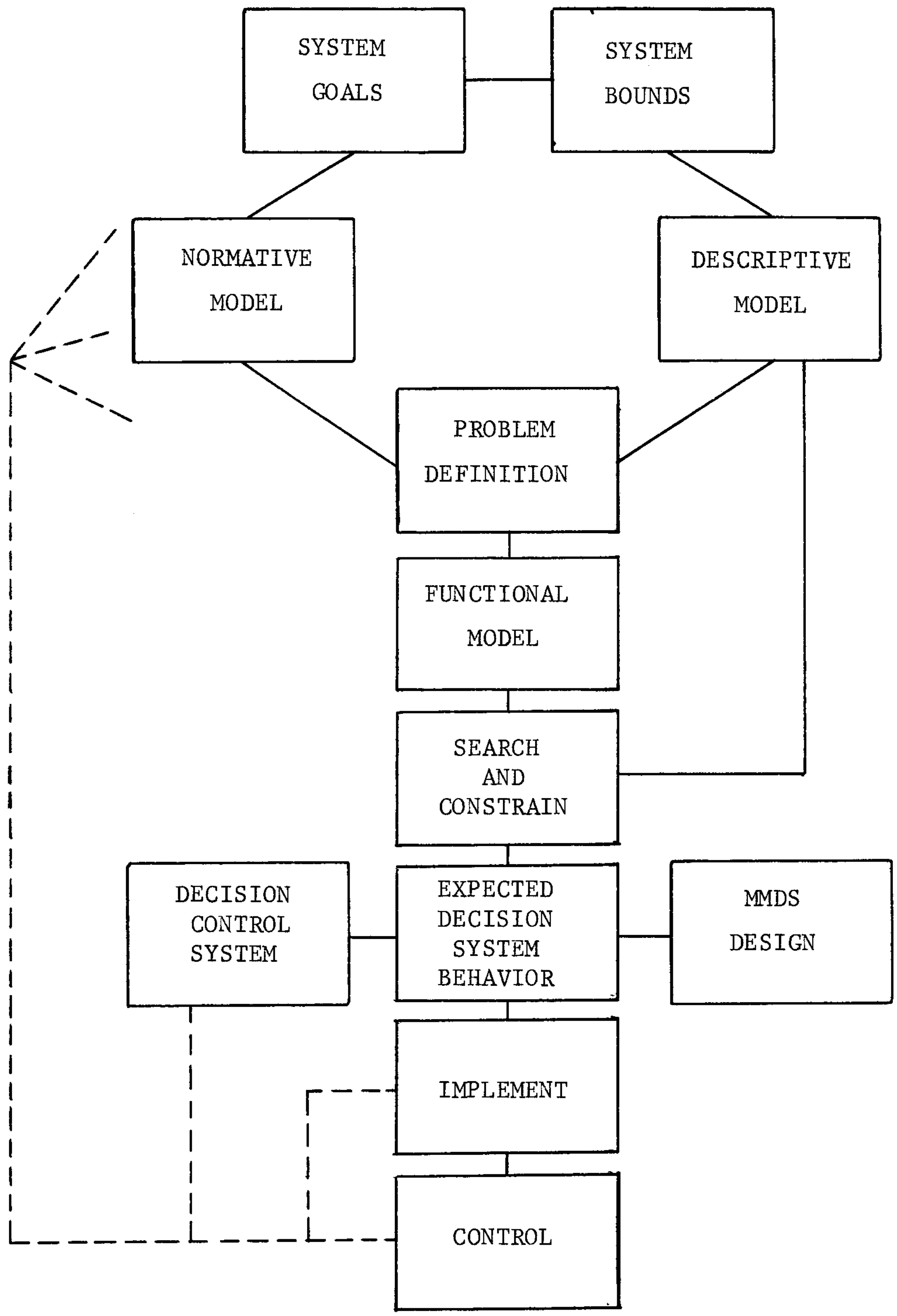


FIGURE 4.1

A MMDS Design Methodology

detailed design. In addition, there may be parallelism as well, with several people pursuing different tasks at the same time.

System Goals and Bounding

In the stages of goal definition and system bounding, the objective will generally be to focus on a decision system with reasonably tight bounds. At this stage in our understanding of MMDS, it seems prudent to select problems of manageable scope, perhaps focusing on just one type of decision. Nonetheless, as suggested by Ackoff (1967), some investment of effort in analysis of the general environment context of the selected decision system will be valuable in identifying outside constraints, as well as potential opportunities for later generalization of the decision system.

Initial decision system identification and bounding could be aided by possession of a set of archetypes of typical MMDS's (Miller, 1969) and of characteristics of decision tasks most readily supported by an MMDS (Scott Morton, 1967). It is in these early stages that a preliminary judgment must be made as to the character of the decision task and into which of the following categories it falls:

(1) totally non-programmable;

(2) totally programmable (either optimally or heuristically);

(3) a mix of programmable and non-programmable elements.

If it falls in the first or second category, then it may be allocated exclusively to man or machine respectively. In general, it is only in the third case that a MMDS approach may be called for, and the methodology described here applies to that case.

Decision System Modeling and Problem Definition

A decision system "problem" is defined here as a gap between the actual decision process and some standards or normative representation of the desired decision process. In this methodology, it is proposed that this problem definition process be made formal through (1) the development of a descriptive model of the current decision system, (2) the development of a normative model for the decision system which reflects standards for salient system characteristics, and (3) the comparison of the two to recognize gaps, or problems to be solved.

As proposed in the third principle of MMDS design of Section 4.2, the decision model frameworks used in descriptive and normative modeling should be based on models for human decision making. In particular, it is proposed that the decision system components and process phases of the general frameowrk of Chapter 3 be used:

A. Decision Process Phases

1. Intelligence
2. Design
3. Choice
4. Implementation
5. Control
6. Decision Process Structure

B. <u>Decision System Components</u>

1. Memory
2. Operators
3. Plans

There are, of course, a variety of models and representations of decision processes in the theory of human decision making, and it would serve the MMDS designer well to be familiar with more than just this framework (e.g., Chapter 3 relates some of this theory to this framework). This explicit use of decision making models in the design of a MMDS is important to help to avoid a tendency of computer systems analysts and designers to view the problem primarily in its machine-oriented aspects. This "attention bias" is to be expected given the technical emphasis of most analysts' backgrounds, combined with the pressure of "Gresham's Law" for more programmed tasks (e.g., machine systems design) to drive out the less programmed tasks (e.g., decision system design). The disciplined use of decision models in design can counteract this tendency by focusing designer attention on decision processes and mechanics.

Of course, at a meta-level, the design process itself may be viewed as a problem solving process. Here models for problem solving also may be of use in increasing understanding of methodologies for design. The parallels between the structure of the design methodology proposed here and the Decision Process Phases are readily apparent.

The objective in the descriptive modeling effort is not to produce a completely verifiable model of the human decision maker's current decision behavior. That is the approach of some researchers who are working in the simulation of human decision makers in order to develop and check theories of human decision behavior (e.g., Clarkson, 1962; Newell, et al., 1960); the MMDS designer will not ordinarily be able to afford the effort required to achieve such fidelity, although he may adopt some of the researcher's tools and results. Instead, his objective is to develop a reasonable working model of the current process to use as a basis for recognizing problems. The tests of this working model likely need be no stronger than consistency between the model and actual decision behavior as perceived by the designer and by the decision maker himself (whose involvement in the descriptive modeling process is one step in maintaining his attention throughout the design effort). Some of the tools of descriptive decision modeling, beyond those of traditional systems analysis (e.g., interview) could be decision protocol collection and psychological testing.

Decision protocol collection is a technique developed by researchers in human decision behavior aimed at direct, as opposed to retrospective, observation of decision behavior (e.g., see Clarkson and Pounds, 1963). Although based on the unproven assumption that the verbal trace of a decision maker "talking through" a decision is a reasonable mapping of the real underlying cognitive processes at work, protocol collection remains as one of the most powerful techniques in decision system modeling, although this approach has not been used before in MMDS design, with the exception of Scott Morton (1967).

Psychological tests have apparently not been used previously in MMDS design either. One such test, the Role Construct Repertory test of Kelly (1955), is applied in an exploratory way in the prototype MMDS design effort described in Chapter 5.

The normative modeling effort, on the other hand, is aimed at specifying detailed standards or criteria for a desirable decision system. It must be recognized, however, that standards for what is "good" decision making in a complex and unstructured decision situation are not at all obvious, and "optimal" decision behavior is, by definition, impossible to specify rigorously in such a situation. Carroll and Zannetos cite this problem in relation to planning, which is generally viewed as an unstructured decision task:

> ...it is not always abundantly clear just what one is attempting to achieve by planning, with the possibly innocuous observation that he is attempting to bring about some order to an otherwise chaotic situation and so structure his problem.
>
> (Carroll and Zannetos, 1966, p. 157)

And in a similar vein, Churchman observes:

> Probably the most startling feature of twentieth-century culture is the fact that we have developed such elaborate ways of doing things and at the same time have developed no way of justifying any of the things we do.
>
> (Churchman, 1961, p. 11)

In other words, the specification of the normative decision model, as defined here, is itself a judgmental and somewhat unstructured process. In this judgmental process, however, the normative models and mechanisms of management science and operations research are likely to be useful as guides for what is "rational" or "optimal" in the abstract, even if these normative guides must eventually be compromised with real resource limitations on available dollars, technology, time, and human capacity. Some tentative normative guides have already been suggested in the elaboration of the general decision framework of Chapter 3.

In accord with the fifth principle of MMDS design proposed earlier, the normative modeling effort is viewed as a mechanism to encourage wide search for alternative MMDS designs. Thus it is held as important to specify a normative model as a distinct activity, separate from (though not independent of) the activities of descriptive modeling and detailed design -- otherwise the designer is biased more toward incremental and short run goals by excessive attention to the details of the current situations. In other words, the normative modeling exercise is aimed at pushing the designer to look far enough ahead and widely enough to have some sense for the possible bounds of desirable MMDS evolution over the foreseeable future. It represents an attempt to force a breakout from the traditional systems analyst's descriptive "set." An overemphasis upon analysis in the process of design has manifested itself in the way information systems are designed traditionally with extreme emphasis on understanding and descriptive modeling the information system as it exists, before the new computer system is designed and implemented. This leads to the phenomena of incrementalism and suboptimization, and to a limited process-oriented view of information system design.

The comparison of the descriptive decision model with the standards of the normative model within the general decision framework will lead to identification of gaps in decision processes and mechanisms. Approaches to solution of those problems, or reduction in those gaps, will be developed in the functional model representation of the MMDS design described next. It should be emphasized once again, however, that this comparison step is really an iterative process. In any real design process, it will not happen just once, but many times as both descriptive and normative models develop over time.

It should also be noted that the MMDS design that results is expected to be a compromise between the normative and descriptive models. On the one hand, the result should not be based totally on an optimizing model which abstracts away so much of the real decision system that it proves unworkable in practice -- this is the extreme approach decried by Little (1970) and Soelberg (1967b). On the other hand, the resulting design should not be based completely on current expressed informational needs with the naive response of making all such "desirable" data available on-line to the decision maker -- this is the data-centered approach decried by Ackoff (1967). In other words, the MMDS design should represent a reasonable balance between long run decision process goals and short run limitations in resources and capacity for change.

The Functional Model

Given problems defined by gaps between the normative and descriptive decision models, the MMDS designer should develop approaches to alleviate these problems. These approaches -- programmable functions, operators, plans and their related data structures -- constitute the functional model. In other words, the functional model is a concise representation of those decision system components of operators, memory, and plans which should be transfered to the computer in the MMDS in order to improve the decision process (or to reduce gaps between the normative and descriptive models).

This phase of associating operators with problems, or gaps, is one of the most creative in the design process. It is here where the base of design assumptions and expectations -- the foundation for the model of expected MMDS behavior -- is beginning to form. For each specific association of operator with problem implies an assumption, a prediction that the chosen operator will affect MMDS behavior so as to alleviate the problem. These assumptions should be made explicit so they may be used for control on actual MMDS performance.

Thus the purpose of the functional model is to identify those programmable operators, operands (i.e., data structures and files), and plans which are expected to reduce the problems defined. Note that these functions may be derived both from existing decision system components which are programmable and also from new components suggested by the normative model. The fact that these operators are derived from human decision making frameworks -- both normative and descriptive -- helps to insure that they will fit into such a process in the eventual MMDS. In other words, it is suggested here that a test of the potential workability or attractiveness of a programmed operator to a human is whether or not it logically supports one or more of the phases of the human decision process model.

Note that human decision behavior has been simulated before by the process of identifying and programming those operators plus appropriate rules and plans (e.g., Newell et al., 1960; Clarkson, 1962; Swanson, 1964). That is not the intention here. Rather than attempt to program the whole structure of memory, operators, and plans as a means of describing a theory of human decision behavior, the object here is to program portions of the process in a way that allows improvement over existing decision behavior. Since plans are less well understood in human decision making than are operators and memory, it is expected that the latter components will be programmed first in the evolution of a MMDS.

The design of a man-computer system for monitoring and controlling automated space probe checkout, described by Chesler and Turn (1967), shows some of the aspects of functional modeling for MMDS design proposed in this section. In particular, they did develop a descriptive model of the basic functions involved in the process, and related those functions to both information and display format requirements. Their analysis, however, is not related explicitly to any general models of problem solving, either descriptive or normative. They do, however, raise the notion of complexity reduction through recoding of data in displays, and several behavioral hypotheses are suggested. One specific function of interest which they propose produces a "moving network," an activity network which moves across a line on a CRT screen which indicates the "present." This display aids in maintaining a sense for general activity context and for events immediately expected.

In the design of the interactive terminal system for use in their MMDS experiment in concept formation, Newman and Rogers (1966) develop an explicit functional model as a design guide. They base their functional model on operators derived from the following two sources:

1) An informal survey of highly trained scientists and engineers was made. Each of these was presented with a manual version of the problem and asked to try to solve it. Afterwards, they were asked what computer aids they would have liked to have in the process.

2) The theoretical and empirical literature on human problem-solving was surveyed, and some basic functions were abstracted (in particular, from the work of Bruner et al., 1956).

This approach by Newman and Rogers apparently represents the first rigorous attempt to identify detailed human decision functions and to program selected ones as a part of the MMDS design process. The fact that the resulting system performed so well, increasing both user performance and learning, is a significant comment on this design approach.

As an aid to generation of possible programmable operator or function types, the designer might review a list of general decision system operator classes, such as those suggested in Section 3.12. It would seem practically inevitable, however, for a complex decision task that the designer both will include operators which eventually prove useless in practice and also omit operators which become desirable later. This unpredictability, as noted in the discussion of MMDS design principles, suggests tight control on actual MMDS behavior, and upon individual programmed function usage patterns in particular. It also suggests investment in design flexibility such that later modification and addition of operators remains feasible.

Design Search and the Matrix Heuristic

Problems of representation and modeling are ever-present in design in general and MMDS design in particular. In proposing several principles for MMDS design earlier, a need for representations was noted both to aid in communicating the design to the ultimate user, and also to help structure a design search space for exploration by the designer. It is proposed that the matrix heuristic introduced below is valuable on both points in the functional model and search phases of the design methodology.

Information processing models of decision making involve the application of a variety of operators to a variety of operands (data structures in memory) in order to reduce the difference between goal and status -- to solve the problem. It was proposed above that this view be applied to the functional modeling of MMDS -- that of operators acting upon operands. Given that this is the basic form of the MMDS functional model, a useful representational heuristic for design is a matrix representing the intersection of operators along one axis and operands along another.

Note that the operators and operands conceived in the functional modeling phase are likely to have come to mind as specific operator-operand pairs in response to particular decision system problems. For example, if a problem in portfolio management decision making is that the decision maker needs a general way of viewing the aggregate structure of a portfolio, one operator-operand response to the problem might be a function called HISTO which could be applied to an operand PORTFOLIO. The result of this application would be a graphic histogram of the aggregate distribution of holdings in the portfolio along a selected dimension (e.g., earnings growth rate). Thus, in this case, the operator-operand pair is HISTO-PORTFOLIO. Note that if one constructs a matrix of all operators and operands defined in this pairwise fashion (e.g., Figure 4.2), the full matrix represents a "design space" of all possible pairwise operator-operand combinations. The HISTO-PORTFOLIO intersection identified earlier is only one point in this space. This matrix, however, focuses explicit attention on many operator-operand combinations that were not conceived of in the initial functional modeling steps. Of course, some of these pairs may be relatively useless, others even meaningless. Nonetheless, some may suggest intriguing ways to generalize a particular function to make the MMDS more powerful. In the example above, for instance, the matrix might suggest application of the HISTO operator to other operands beyond PORTFOLIO. For example,

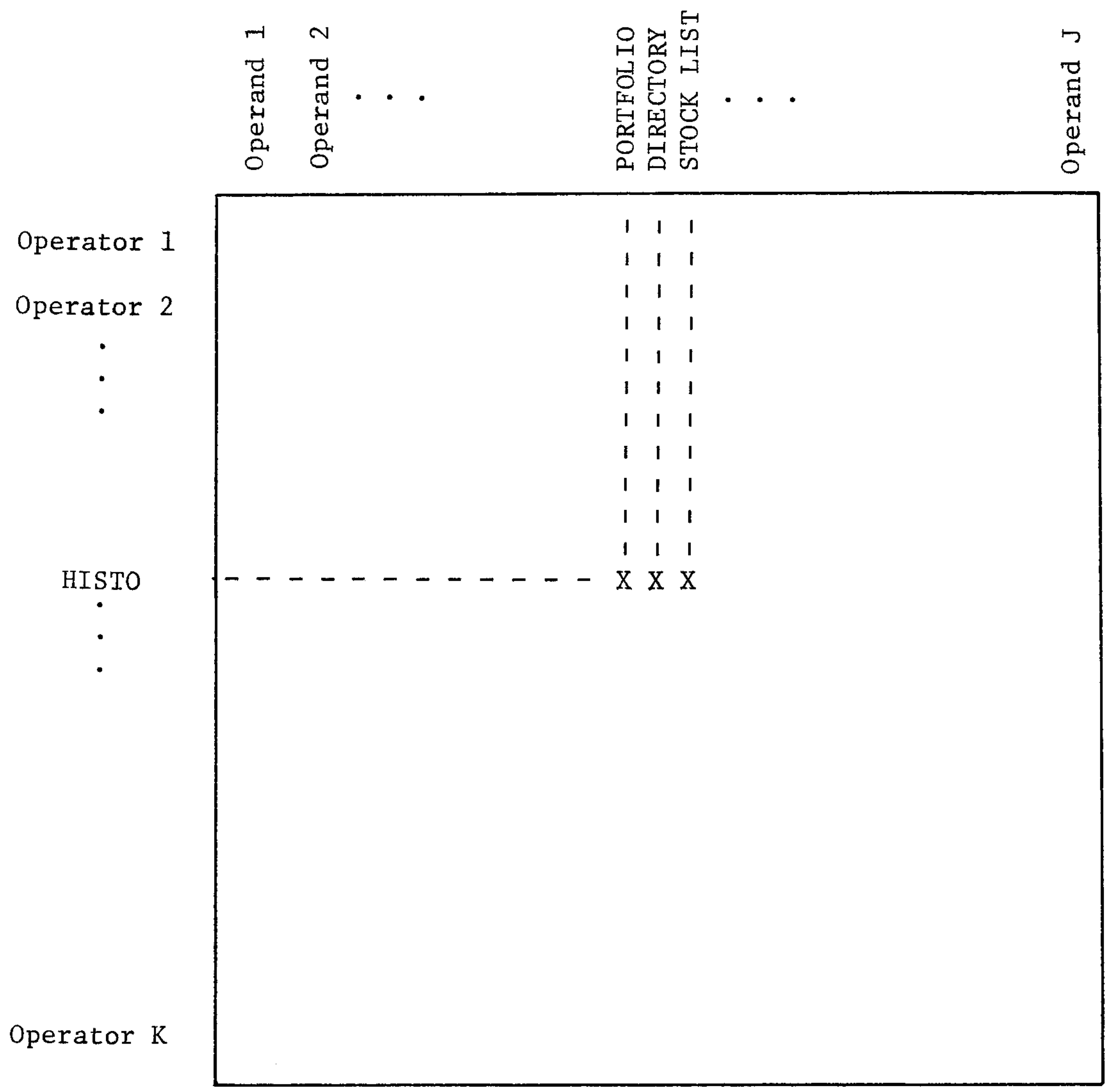


FIGURE 4.2

Sample Operator-Operand Matrix Representation

HISTO applied to a DIRECTORY of portfolios implies a distribution of all portfolios along a selected dimension; this could be useful in spotting portfolios in extreme or exceptional positions, or as a standard against which to compare an individual portfolio. Also, HISTO applied to a STOCK LIST (e.g., the Dow Jones Industrials) rather than a portfolio could also produce an interesting standard distribution against which to compare the structure of a particular portfolio (or the distribution of all portfolios of the decision maker). In other words, the matrix is viewed as a stimulus and aid to further creative design search and operator generalization.

Note that there is no reason that operands may not be complex, in the sense of representing lists of simple operators (or lists of lists, etc.). The capability to have an operand represent a list of operands and operators is useful in order to build "macro" command lists or procedures, an attractive self-adaptive facility for a MMDS.

Such a macro could be run by applying a primitive activating operator (e.g., EXECUTE) to an operand representing a operator-operand list. Flexibility for real-time control of the macro could be realized by allowing for use of "dummy" operands or operators in the list as arguments, where real-time substitutions could be made upon execution. For example, a macro called X might be built to apply HISTO to a dummy portfolio name to produce a specified distribution by percent of market value and price-earnings ratio. Then whenever "X name" was invoked, the pre-specified display would be produced for the given portfolio "name." This allows the decision maker to recode frequently used operator-operand procedures more concisely.

In summary, the matrix heuristic proposed here has the following attractive characteristics:

(1) it is a concise, aggregate description of the system;

(2) it structures, in a rigorous way, an MMDS design space, by displaying all possible operator-operand pairs (for the given arrays of operators and operands);

(3) it may direct attention both to attractive operator-operand combinations that hadn't been considered before, and also to others that represent easy and useful generalizing extensions to the design as originally conceived;

(4) it demands more explicit definitions of both operators and operands;

(5) it is capable of representing hierarchy in both operators and entities with the notions of macro-operators and meta-lists (lists of lists);

(6) it focuses early design attention upon the search for conciseness, uniformity, and generality in operator and operand definition.

Using the matrix representation as a reference, the process of specifying reasonable general design requirements for the MMDS is a highly iterative process involving search for new functions and capabilities as well as a screening of those capabilities based upon real constraints of technology, available economic resources, time, and organizational and individual limitations. It involves an examination of the descriptive model of current information and decision processes vis-a-vis the normative model and making judgments as to how much change is tolerable or desirable in the MMDS implementation. It also involves a consideration of the character of human cognitive capabilities, and thus the designer should be familiar with relevant behavioral theory such as that reviewed within the general framework in Chapter 3. One result of this application of constraints should be the selection of allowable operator-operand pairs from the full array displayed in the matrix representation.

Conversational Graphs and MMDS Design

In the detailed design phases of MMDS development, the designer requires some means of representing the terminal interface characteristics of the system. He needs this representation both to communicate design characteristics to the ultimate user, as well as to guide the detailed design of supporting software. One useful form of this representation is the conversational graph introduced here.

This graphic representation was developed in the process of design of the prototype MMDS of Chapter 5 and is being refined and rigorously evaluated in a follow-on research project (Carpenter, 1970). Also, it was noted recently that another researcher independently has pursued a similar course (Parnas, 1969).

The conversational graph is another means of representation of the MMDS design, but a more discursive and detailed representation than the matrix. The basic notion of the graph is that the machine portion of the man-machine dialogue can be represented by a finite number of "states," or points in the "conversation." The conversational graph is then just a form of state transition diagram which describes the allowable state transitions in the system. Within the graph a transition is typically initiated by the human user, along a branch selected by him. However, this is not always the case, as a transition may be automatically initiated by the machine as a result of a particular computation (and perhaps as a result of the particular state of the real-time data base). The actual dialogue can be represented within the graph at nodes and along arcs.

In other words, at any point in a man-machine dialogue, the MMDS may be thought of as being at a particular node in the conversational graph. The graph represents the computer's view of the context-dependence of the conversation; i.e., a given user response will be interpreted differently at different nodes in the graph. Examples of conversational graph specification are given in the description of the prototype MMDS design of Chapter 5, but a sample segment of a hypothetical graph is shown in Figure 4.3. Note that the graph form is completely consistent with the "menu-selection" form of man-machine language in the sense that at each point in the dialogue (at each node) the user has a defined set of alternative courses of action (departing arcs).

The form of the graph raises several explicit issues in MMDS design:

1. At any given point in the man-machine dialogue, how much explicit flexibility can be provided without confusing the human user? (E.g., how many possible arcs should come out of any given node?)
2. How much flexibility can be provided to change the conversational context radically? (E.g., how can the user "jump" from his current position in the graph to some remote node?)
3. How much flexibility to accelerate the pace of conversation at will can be provided? (E.g., how can the user compress the dialogue necessary to move deep into the graph hierarchy?)

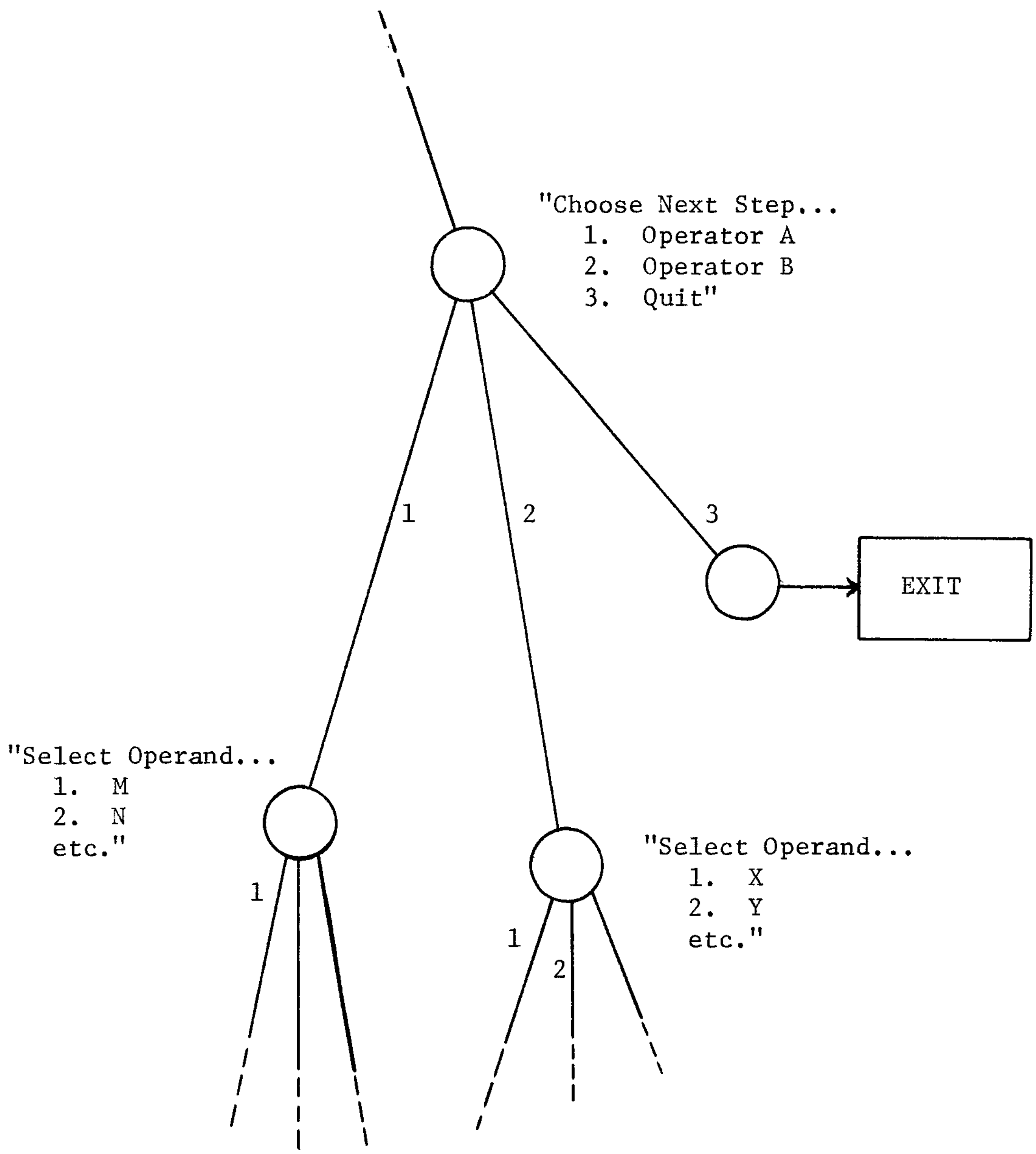


FIGURE 4.3

Sample Conversational Graph Segment

4. How much of a capability to recode the dialogue can be provided directly to the user? (E.g., how would a "macro" be constructed and used in the context of this conversational graph?)
5. How can flexibility to alter the structure of the graph easily be provided? (E.g., what software architecture allows for easy system modification following a change in desired conversational graph structure?)

Although all these questions are not answered here, one resolution to some of them is embodied in the prototype MMDS design described in Chapter 5.

In summary, the conversational graph can be useful to MMDS design in a variety of specific ways:

(1) It represents a concise, user-oriented detailed description of the MMDS;

(2) It is in a form such that conversational "scenarios" can be generated readily, mentally or by hand, in order to evaluate a sampling of the myriad possible decision process traces that the system may support. In other words, a significant amount of valuable human interface "debugging" can be done with a paper design;

(3) It is simple enough to serve as a direct aid in new user training, once the MMDS is implemented, since it combines sufficient detail with the advantages of a graphic overview of system conversational structure, yet without introducing unnecessary complications of a machine-oriented description of how the system is actually realized;

(4) For design of the MMDS monitor and control system, the graph represents a complete and extremely useful model of all aspects of the decision process conversation that are directly observable by the machine.

Given this conversational graph specification, the MMDS detailed design can then proceed in a "top-down" fashion, providing the routines necessary to support the functions specified in the graph and reiterating through earlier design stages where necessary when additional technical constraints are encountered.

Design and Application of the MMDS Decision Control System

Control, formal and informal, in the form of feedback and adaption is ubiquitous in human decision systems, such as a MMDS. Here, however, the primary focus is on a formal decision control system to allow the MMDS designer to check his assumptions and continually improve his design.

Essential to decision control system design is the collection and recording of all significant design assumptions and expectations developed at various points throughout the process of design. Given this codification, the control system design can then proceed by specifying explicit checks and control mechanisms where possible for these assumptions.

A primary mechanism for control would be a machine monitor capability for remembering "traces" of the man-machine dialogue for all user sessions. In other words, this decision monitor could maintain a complete sequential listing of any given conversation, as well as collect operator and operand usage statistics and any other special measures of interest. The conversational graph, in the sense that it represents the available conversational structure of the system, provides a useful framework for identifying specific monitor points within the system (e.g., the monitor could be set to count the number of times a particular node is occupied or a particular arc is traversed in the process of using the system). The conversational graph is particularly useful to decision control in the sense that each of its nodes and functions is designed to have meaning in the user's decision process -- thus transition paths through the graph are meaningful to the decision control system. Once a decision maker begins to use a MMDS, in fact, his decision process may be subject to more rigorous and consistent analysis, given the decision monitor capability, than was ever possible before. This enhanced analysis may lead to better decision modeling and eventually to more decision programming in a given decision system.

Other, more traditional mechanisms may be employed as well in the decision control system. In other words, interviews, questionnaires, protocol collection and psychological tests may be used either to confirm results found in the monitor trace or to check aspects of MMDS behavior relatively inaccessible to the decision monitor system. Examples of the application of some of these decision control techniques are given in Chapters 5 and 6.

Implementation

A detailed discussion of the process of MMDS programming and implementation is beyond the scope of this dissertation. However, several general points should be made.

One, it is expected that the programming and testing of a conversational computer system is greatly aided by the use of a general purpose time sharing system environment.

Two, the combination of a highly complex conversational system (with a huge number of possible system and data base states) and an unpredictable human providing the driving input can result in an incredibly complex man-machine system. The situation is even worse if the system relies on a data base that follows some real process, further increasing the number and unpredictability of system states.

Debugging and testing are correspondingly much more complex than in a more rigid and programmed situation. Hence, investment in automated check-out and programmed system exercise routines may well be worthwhile. Also, the need for extremely tight controls on input data or commands at the point of entry should be obvious.

Three, given the unpredictability of much MMDS behavior at our present state of knowledge, planning implementation so as to provide a "live" prototype or limited capability system may pay high returns in feedback from early user experience.

4.5 Summary

This chapter has begun by introducing some of the general issues in design of information systems. It has been noted that most of the literature of design focuses on the later, more structured phases of programming, system optimization, and implementation, at the expense of the early phases of problem definition, analysis and preliminary specification. It was further noted that many applications of the computer, even those aimed directly at supporting a decision maker, seem to take a machine-centered, data-centered, or other such approaches rather than a decision-centered approach.

Considering the special characteristics of MMDS identified in Chapter 3, several principles of MMDS design are then proposed and elaborated. Finally, a design methodology is proposed which is an adaption of the general methodology of Chapter 2 for the special case of MMDS design. This methodology is distinct from traditional information systems design literature in its focus upon (1) the early phases of analysis and general design, (2) explicit control on the design, and (3) a decision-centered approach.

This MMDS design methodology introduces several new heuristic tools (e.g., the operator-operand matrix, the conversational graph) over and above the more traditional techniques of information systems analysis and design. More important than the characteristics of these specific tools, however, is the general notion of exploring, elaborating, and evaluating such techniques in the development of a formal body of MMDS design methodology.

The methodology of this chapter is applied and described in more detail in the design and use of a prototype MMDS as outlined in the next two chapters.

CHAPTER 5

AN APPLICATION OF THE DESIGN METHODOLOGY IN PORTFOLIO MANAGEMENT

5.1 Introduction

The design methodology developed in earlier chapters was applied by the author in the design of a prototype man-machine decision system for portfolio management. The resulting MMDS was exercised by responsible portfolio managers in a realistic field situation. The design of this prototype MMDS is described in this chapter. The results of the design and of exercise of the prototype system are described in Chapter 6.

The tactic of an exploratory field study was selected as a first step in the evaluation of MMDM theory and design because of the very primitive state of the field at present. It was felt that such a comprehensive approach would yield a higher information return at this stage than a controlled laboratory experiment. The field of MMDS is currently characterized by little more than a few fragmented experimental results. The strongest current need of the field is for integration -- the later factoring into specific hypotheses for more rigorous testing will benefit from the guidance of a general framework for the field. Another payoff of an exploratory study should be the discovery of new insights and ideas, as well as refinement and clarification of the few identified existing hypotheses.

Such a study can be aimed also at exercising possible measurement instruments and aiding in the establishment of specific priorities for follow on research. Finally, such a study is conceived as having useful impact on the practice of MMDS design, to stimulate more design of such systems, and a greater diffusion of MMDS experience than exists at present.

The trade-off in adopting such an exploratory tactic is, of course, that one may sacrifice the notion of statistical significance of results for increased comprehensiveness and realism. The emphasis here is upon the design experience and upon in-depth observation of a relatively small number of subjects in a realistic field situation. The lack of experimental control that is characteristic of a field study poses dangers, some of which were encountered in the course of this research. (The pitfalls met and the resulting modifications in the original design of this study are discussed as a part of Section 5.9 on the Control System.)

Other limitations of field research with responsible decision makers as subjects in their on-going work environment are that (1) subjects cannot be inconvenienced seriously in carrying on with their job; (2) subjects may have inhibitions in communicating freely to the researcher about their decision behavior, despite assurances of "privileged communication", (3) "live" data collection is difficult so that some reliance on retrospective description may be inevitable.

Despite these limitations, however, there is a very important place for field studies in developing and supporting hypotheses about non-programmed MMDS behavior that are applicable to real decision situations. As Soelberg (1967, pp. 1-8) notes, "There is some doubt whether it is even possible to simulate critical decision problems in the laboratory, hence whether most of our knowledge about critical human choice behavior will not have to come from observations in less well controllable field settings."

The design and exercise of the prototype MMDS for portfolio management took place over the period from December, 1967, through February, 1970. The major milestones and phases of the project are sketched in Figure 5.1 on the following page.

This chapter traces the history of the project, using it as an illustration to shed light upon the MMDS theory and design methodology described earlier. The chapter is organized by phase of the design methodology, from normative model to implementation and control, and it concludes with a description of this designer's expectations as to resulting MMDS behavior.

5.2 The Decision Context

The project was initiated in December, 1967, with a feasibility study and proposal for the experimental development of a prototype conversational Portfolio Management System (PMS). The decision

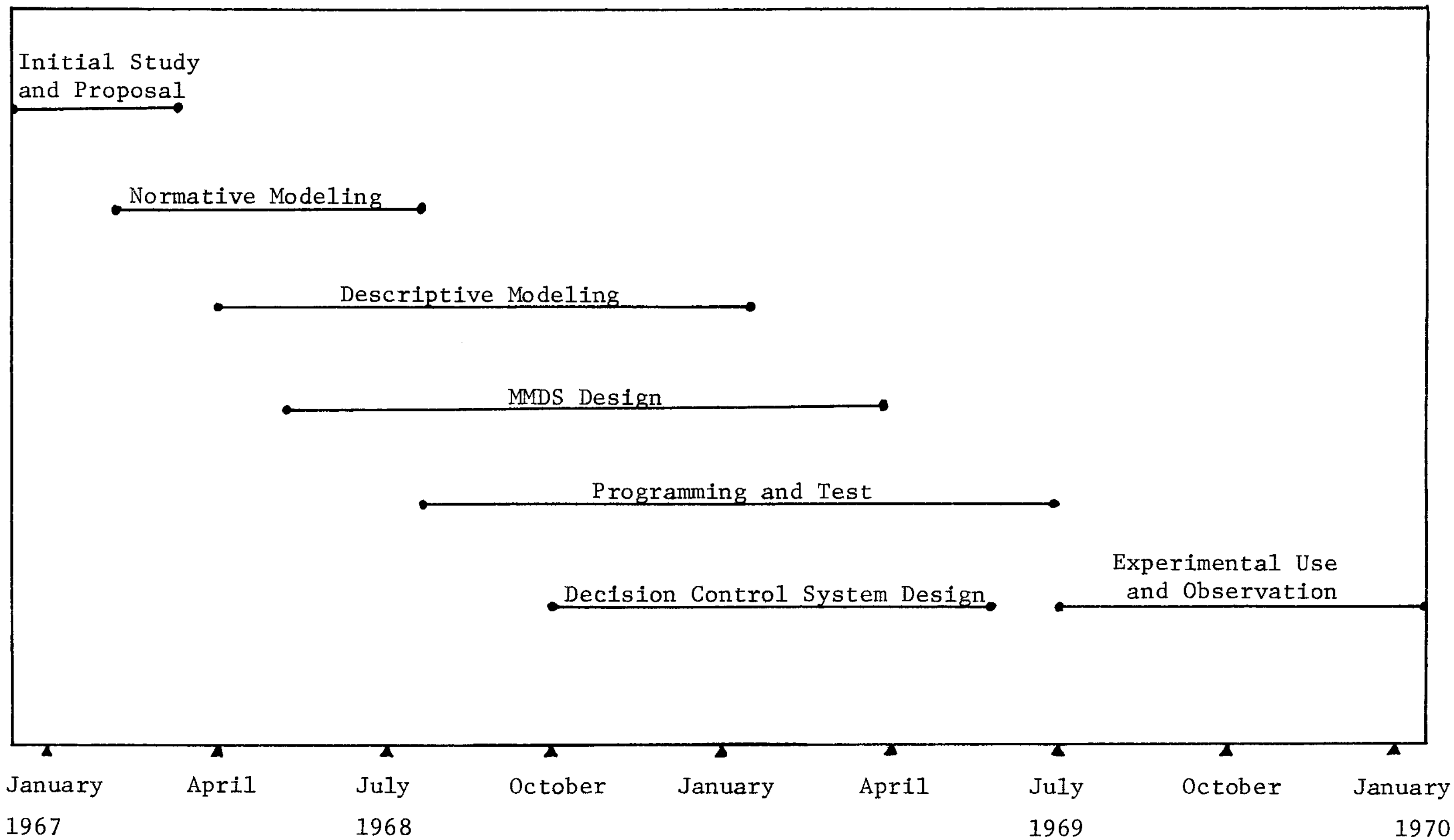


FIGURE 5.1: Major Phases in Prototype MMDS Development and Use

context was the pension fund section of a major U.S. bank. The aim was to develop and exercise a PMS which supported a handful of pension fund portfolios on an experimental basis.

The portfolio managers who would exercise the PMS operated within the organizational and informational environment outlined in the figure on the next page. This environment is similar to that encountered in many large institutional investment situations. The main components of the figure are described below:

1) Investment Research is the "intelligence" wing of the organization. It consists primarily of security analysts who are responsible for monitoring, adding, or deleting securities from the approximately 350 common stocks making up the Approved List. Approved List securities are authorized for investment by the portfolio manager -- this is his "universe" of investment opportunities.

2) The Committee Structure is responsible for reviewing all changes in status of the Approved List, setting portfolio management policy guidelines, and reviewing portfolio management performance.

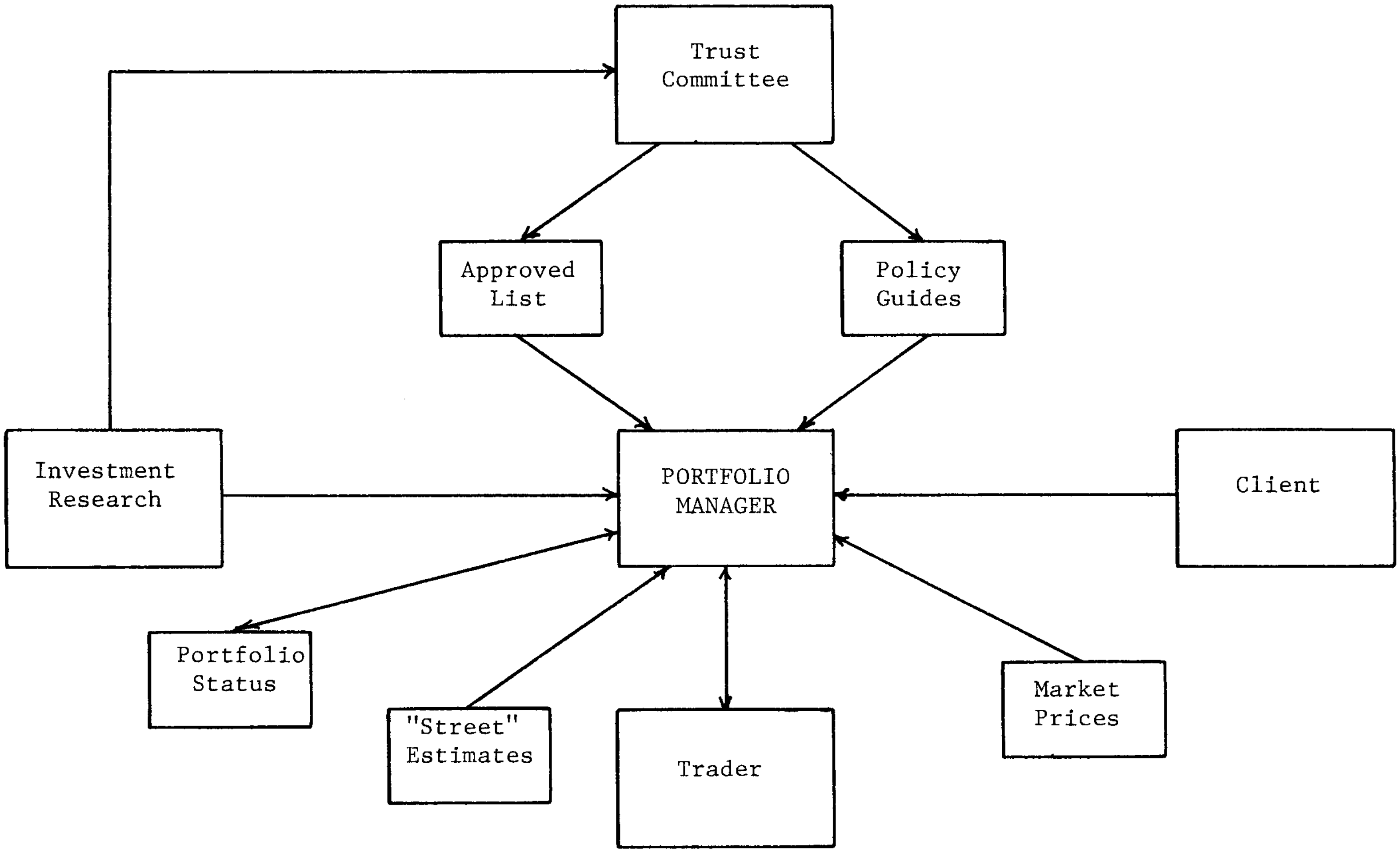


FIGURE 5.2

The Portfolio Management Information Environment

3) The Trader is part of a specialized trading group responsible for execution of all portfolio management trading decisions.

4) The Client, in the case of the pension fund section, is a company, typically represented by its treasurer or a committee.

5) Information used by the portfolio manager consists of periodic reports from Investment Research on Security Status and expectations, informal or formal reports from the "street", instructions or revisions of objectives from clients, periodic reports on portfolio holdings, and current security prices from the newspaper or a Telequote terminal.

Over the course of the project, there were on the average seven active portfolio managers in the pension fund area. Each man typically handled from 25 to 100 portfolios, ranging in asset value from \$2 to \$350 million each, normally with 30-90 common stock holdings per portfolio. The portfolio managers ranged in age from about 32-55, and all had at least a bachelor's degree. None had any prior direct computer experience of note.

The focus of the decision system was taken to be the portfolio analysis and revision process. This involves the scanning and analysis of information from all sources and the resulting choice of portfolio revisions.

The focus is also primarily upon the common stock portion of portfolios, not the bonds, mortgages, and other asset types. This focus was elected (1) because the common stock portion is the most active, and receives the majority of the manager's attention, and (2) because the common stock data base is more readily constructed and managed for a prototype system. (In most cases, common stocks constituted 60 - 70 percent of the portfolio.)

5.3 The Normative Decision Model

The normative and descriptive models of the pension fund manager decision process were developed partly in parallel through 1968. The general decision system framework developed in Chapter 3 was used as a guide for constructing both of the models. This framework consists of the decision phases of intelligence, design, choice, implementation, and control, plus the further primitive decision elements of memory, understanding, and operators.

Since much of the early effort was directed at the normative model, it is described first. It was developed as a direct elaboration upon the general decision system framework, after some reflection upon the task description, normative decision theory, and the literature of finance. The model is described below with goals or normative characteristics listed within each decision phase or element. The feasibility of satisfying each of these subgoals will not be faced until later project phases. Hence the model here is phrased deliberately in rather general terms, avoiding reference to any specific optimizing model or algorithm, or to any specific hardware. Note that this normative model refers to the total decision system, including both human and machine aids.

Intelligence

The decision system should have immediate access to current status information both for portfolios and securities. The system should possess current goal statements for each portfolio regarding desired appreciation, volatility, liquidity, income, etc., as well as operational standards or criteria related to those goals. The system should possess mechanisms for continual screening of current status against current criteria or standards so that differences, or "problems", can be recognized.

Design

The decision system should have mechanisms for search for and design of alternative portfolios. Such search activity should be problem-oriented in the sense of having a clear association between problem definition and the direction and degree of search for alternative solutions. The system should have a clear definition of the dimensions and feasible bounds of the portfolio search space.

Choice

The decision system should possess mechanisms for the analysis and comparison of alternative portfolios against the accepted criteria and standards. The system should have the capability of balancing the increased performance of any given alternative against the cost of trading involved in getting to that alternative from the current portfolio.

Implementation

The decision system should provide for immediate transmission of chosen trade orders to the appropriate trader for action.

Control

The system should provide explicit control mechanisms to operate upon:

(1) the trade execution;

(2) the decision process (i.e., its structure, content, consistency, etc.);

(3) the predictive quality of security analyst projections (from both inside and outside the organization);

(4) individual portfolio performance;

(5) portfolio manager performance across all portfolios;

(6) portfolio profitability for the organization, individually and by class of client.

The control system should provide for automated exception reporting whenever the behavior of any of these activities deviates significantly from accepted standards or expectations.

Plans

The decision system should have mechanisms for the development and improvement of models and procedures for aiding security evaluations, portfolio selection, and other elements of portfolio management process.

Memory

The decision system should have memory and effective data structures for handling information, models, and procedures relevant to the above decision process.

Operators

The decision system should have mechanisms for the development and application of effective operators for carrying out the decision processes described above. An initial list of such operators will be derived, after comparison of normative and descriptive models and during the functionalization phase described later. However, some general classes of such operators have already been identified in Section 3.4.4.

Thus the initial normative modeling effort has done little more than identify a set of desirable characteristics of a portfolio management decision system. Nonetheless, these characteristics and the general decision framework which lies behind them prove very useful as guides for problem recognition in the analysis of the actual existing decision system, described in the following section.

5.4 The Descriptive Decision Model

The descriptive model was developed within the general decision system framework used for the normative model. A variety of information sources was used: interviews, reports, decision protocols, questionnaires and psychological tests. These sources were not employed in any strict sequence, but the descriptive model developed iteratively with frequent checks back to the real decision process. The portfolio managers' reactions to the model were also solicited periodically.

Repeated interviews were conducted with seven of the pension portfolio managers through 1968. The form of the interview was structured about the general decision framework and aimed at eliciting the portfolio manager's view of his own decision process; how he used the information available to him, how he recognized problems, etc. As the skeleton of the model began to form, the model itself became one focus of the interviews.

At the same time sample copies of all periodic and ad hoc reports used by its portfolio managers were collected. The perceived relevance of these reports to the portfolio manager was examined during interviews and protocol collection. They were also analyzed in light of the goals of the normative model, and deficiencies in terms of content, structure, timeliness and flexibility were noted.

Decision protocols, the manager's verbal trace of his thoughts during the process of portfolio analysis and revision, were elicited and tape-recorded for six portfolio managers. The six sessions were an average of about 50 minutes in length, with each portfolio manager reviewing one or two portfolios and making buy-sell decisions. The developing descriptive model was checked against these tape-recordings.

Questionnaires were collected in 1968 and 1969 from seven of the portfolio managers with regard to their actions taken on 23 portfolios. These covered about three months of buy-sell transactions on these accounts, explaining the time and reasoning involved in each transaction. The volume of transactions on six of these portfolios for the first six months of 1968 were extracted from the department's accounting records. These six portfolios showed an average of 30 stock trades executed per month for each portfolio. In fact, these 30 trades represented only an average of 9 stock transaction orders per portfolio per month. (A single transaction order placed by a portfolio manager may be broken into several smaller trades by the trader for easier execution.)

Finally, Kelly's (1955) Role Construct Repertory Test was administered in early 1969 to six of the portfolio managers to elicit the major factors that they used in discriminating among stocks on the one hand and among portfolios on the other. The aim here was to use one more instrument to examine the way in which the managers perceived stocks and portfolios. The results of the two hour Role Construct tests were to be used as a base line against which to measure changes in perception as a function of use of the prototype system in the original decision control system design.

The data sources described above were used in the iterative development of the descriptive decision model outlined in the following sections. The general characteristics of the portfolio manager's job are sketched first, and then his decision process is described within the general decision framework.

General Description

The typical portfolio manager (PM) spends significant and roughly comparable portions of his time in (1) customer contact, (2) review and revision of portfolios, and (3) scanning of the security market and related information.

The time spent on a portfolio review may vary widely from a few minutes to several hours. However, the majority of these reviews involves a significant clerical effort on the part of the portfolio manager (PM) and his assistant to produce an adequate picture of current account status, updated from the latest computer-produced review. The pension fund manager typically will try to review his major accounts on a weekly basis.

Pension fund managers place a high premium on easy access to current portfolio status and to issue holdings across portfolios. The pension fund situation is inherently more dynamic than other trust management situations with its relatively high cash flows, both in and out, with high and increasing client pressure for performance, with increased potential for large block trading, and with more frequent reviews of account status.

Intelligence

Relative to other phases of the portfolio management process, the largest proportion by far of the PM's time appears to be spent in the intelligence activities of status monitoring, goal refinement, status-goal comparison, and problem definition.

The status monitoring function involves (1) reviewing portfolios and (2) scanning general market and security-related information. The first is directed at discovering problems in portfolios, the latter in discovering opportunities in the form of attractive securities. This dichotomy is analogous to "need-stimulated" versus "opportunity-stimulated" problem recognition as discussed in Chapter 3. Both types of problem recognition are present to a significant degree, although security sale transactions are relatively more need-stimulated than are security purchase actions. In general, the opportunity to buy a specific attractive stock appears to be considerably more important as a decision stimulus than the discovery of a need in a particular portfolio. This observation was strongly supported by the PM interviews and by the questionnaires, where the PM classified each transaction decision as either need- or opportunity-stimulated.

The primary source of portfolio status information is a fixed format report produced at most monthly. It is organized by class of security (e.g., stock versus bond) and then by industry group (e.g., oils). It is typically from 5-30 pages long and lists all holdings at cost and market value.

The primary source of security analysis information is another fixed format monthly report produced by Investment Research and covering the stocks of the approved list. This report is also organized by industry group and lists approximately 25 variables per stock (e.g., price/earnings ratio, latest earnings per share, projected earnings growth rate, etc.).

The primary source of security price information is the Wall Street Journal or a Telequote terminal located near the managers' desks. It is clear from the outline above that the PM's status information base is fragmented, rigidly formatted, and not always current.

The portfolio status reports give little information on aggregate portfolio structure beyond a breakdown of total cost versus market value for (1) the whole portfolio, (2) the major classes of assets in the portfolio, and (3) the industry group totals for the common stock portion of the portfolio. Interviews and decision protocols revealed that PM attends primarily to the following aggregate or structural portfolio characteristics, if any at all:

(1) industry diversification in the stocks;

(2) the proportion of stocks in the total portfolio;

(3) the total number of holdings;

(4) the number of holdings per industry;

(5) the amount of cash available.

The PM's use of the report is accompanied by considerable clerical effort involved in updating both holding status and asset prices to get a current picture of the portfolio.

On the other hand, when asked to discriminate among various subsets of 20 portfolios (which he managed) during the Role Construct Repertory Test, the average PM used mainly dimensions related to client goals. In descending order of frequency of use, the PM's discriminated among portfolios on the following dimensions:

1. portfolio has a relatively high (relatively low) total return goal;
2. client will accept little (high) risk;
3. portfolio is very inflexible (flexible);
4. portfolio is large (small);
5. high income (high return) is the primary goal;
6. client has very vaguely (clearly) defined goals;
7. client is very satisfied (dissatisfied) with performance;
8. client wants to become more (less) aggressive;
9. client has low (high) liquidity need.

With the exception of size and relative inflexibility (due either to asset structure or to client constraints), none of these dimensions perceived by the PM's relate to portfolio status or expected performance. This is not surprising given the dearth of portfolio-related measures available to him and the consequent focus of his information system upon individual assets.

On the other hand, the Role Construct Repertory Test, when applied to 20 stocks known to the PM, showed that the significant dimensions were very much performance or status related. Again in descending order of frequency of use by the PM's, the following dimensions were used to discriminate among common stocks:

1. high (low) expected earnings growth;
2. earnings growth is volatile (stable);
3. high (low) total return potential;
4. quality of stock is high (low);
5. risk is high (low);
6. company is very aggressive (conservative);
7. company has good (poor) management;
8. stock is undervalued (overvalued);
9. P/E is low (high);
10. income is high (low);
11. in the consumer (non-consumer) sector;
12. in a non-growth (growth) industry;
13. leader (laggard) in their industry;
14. industry holds no (high) market interest at present.

Besides being more performance-related, note that there are more dimensions here than in the portfolio test. In fact, some simple tests for cognitive complexity (Bieri, 1955) based on the test results indicate that five of the six PM's discriminate among stocks in a more complex fashion than they discriminate among portfolios. However, although this difference was substantial, it was not statistically significant at the 10% level.

Portfolio goal definition and refinement usually takes place in consultation with the client. The interviews established that the client typically only expresses very vague notions (if any at all) of his desires for return, liquidity, risk, etc.

In summary, the PM defines his problems more in terms of individual stock characteristics than in terms of portfolio goals and status and his focus of attention is more on single assets than total portfolios. His key reports on portfolio and security status are produced only periodically, are no longer current when received, and are rigidly formatted. His base of relevant data is fragmented into several sources and files, and he spends a significant amount of clerical effort in coping with the fragmentation.

Design

Like intelligence, the design phase of PM decision making also focuses upon individual assets. In interviews and decision protocols, the PM sees himself as choosing among single asset trades, not as selecting from alternative possible portfolios.

The PM exhibits little formal search activity in the process of design (or intelligence). His formal search is primarily restricted to periodic review (once a week for major accounts) of portfolio status reports and daily scanning of price and volume movements in the *Wall Street Journal*. The monthly Investment Research report on stock history and expected performance typically is scanned upon receipt and then seldom explicitly referenced except to answer specific client questions.

Scanning the periodic account status report generates both buy and sell candidates in the PM's mind. In other words, in receiving an account, the PM thinks first about either selling current holdings or buying more of or "dollar averaging" current holdings. If adding to current holdings does not reduce cash to a satisfactory level or if an asset not held looks particularly attractive, then the PM will look to other candidates from the approved list; this wider search is almost exclusively mental with no explicit reference to formal reports.

The other way buy and sell candidates are found is by the PM's receipt of new information which changes the attractiveness of a particular asset. This is opportunity-stimulated decision making. If an asset has become desirable to sell, the PM will sell out of all portfolios where such action is feasible, relying largely on memory to indicate the appropriate portfolios. On the other hand, if an asset becomes an attractive buy, the PM will try to have all appropriate accounts buy into the asset.

Choice

The PM choice phase appears to be well represented by the "satisficing", aspiration-level model of decision making. Alternative buy and sell candidates appear to be considered one at a time with "satisfactory" alternatives being chosen. Thus decisions made are very much a function of the search processes employed, since an exhaustive list of alternatives is not considered.

There are no formal mechanisms that allow the PM to view an alternative buy or sell in terms of the alternative portfolio the transactions would create. Hence the PM tends to consider stock alternatives primarily in terms of their inherent characteristics versus some mental standard. Occasionally, the decision protocols revealed pairwise comparison. (E.g., "This stock has a higher expected earnings growth with about the same quality as this other stock so I will buy the first.") The few heuristics which related individual asset decisions to overall portfolio status or structure were such rules as the following:

1. try to reduce the total number of stock holdings to less than 40;
2. avoid too many holdings in a single industry;
3. avoid having more than X% of the portfolio in security Y (X is a policy limit set by committee);
4. avoid having the portion of the portfolio in common stocks significantly different than Z% (Z is a policy target set by committee).

In general, there is no formal application of criteria to alternative transactions in the choice process. The screening is highly judgmental, involving a subjective balancing of heuristics such as those suggested above.

Implementation

The process of execution of a single security trade or a "program" (a set of several trade orders) is initiated by the PM's sending a written or verbal order to the trading desk, where specialists execute the trade, perhaps in several portions over several days.

If the PM wants to sell a particular security out of all of his accounts, the formal mechanism available to aid him is a computer program which lists the holdings of any given security across all accounts. This is a special system, however, only run on request, and it typically takes 2-3 days to get a report. Hence the PM tends to rely on memory or a manual search of portfolio status reports to derive such a list.

Control

There are few formal control mechanisms that are used in the PM decision system. For example, there is no explicit feedback to the PM concerning execution (or non-execution) of a trade. He may inquire directly to learn trade status, or he may wait to examine the next portfolio status report, but he gets no automatic confirmation.

Controls on the decision process itself, its structure and heuristics, are largely informal. The PM clearly learns from experience, yet he has few explicit measures of how well he has done based on past actions, nor has he any easily applied standards for comparison.

Plans

Given the limitations in available information and processing power evident in the decision phases outlined above, it is hardly surprising that the PM's decision process appears highly subjective, as indicated by the criteria, models, and heuristics employed.

As observed in the decision protocols and supported by the Role Construct Test the PM's criteria for discrimination do appear more complex and better defined for stocks than for portfolios. A reasonable hypothesis is that this primary focus on individual assets is due in large part to the PM's current information system, which gives him very few aggregate measures of portfolio status. On the other hand, he has many available indicators of stock status, and there is much conversation among PM's about questions of security evaluation.

The models for security or portfolio evaluation used in the decision system are almost exclusively mental. The Investment Research group has developed one model for estimating future annual total return for stocks on the approved list, but it is experimental and not fully adopted by the PM's. As for portfolio evaluation, there is no formal predictive model used, although rigorous historical performance evaluations are conducted on an ad hoc basis. A system to produce period-by-period reports of time-weighted and internal rates of return for all portfolios is currently being installed.

Memory

The formal memory of the PM decision system involves periodic internal reports, newspapers, advisory reports, magnetic tape files from the computer-based accounting system, etc. For all practical purposes, however, the working formal memory involves just the latest portfolio status report, in which changes are pencilled, and the Wall Street Journal. Beyond these, the PM relies largely on his own memory for other relevant information.

Operators

The PM has few formal or mechanized operations available to him. As noted earlier, he can call for a search for the holdings of a given asset across all portfolios, but it is typically two or three days before he gets the resulting report. He can also request the current computer system for an up-to-date report on a portfolio's holdings without prices, values, or costs, but this also takes a day or more.

Thus the fundamental operations employed in the decision system are mental or manual. For example, the PM or his assistant may manually pencil in holding changes as they are executed on last month's portfolio status report. One mental algebraic operation which is employed frequently in security evaluation is the calculation of a price-earnings ratio base upon current market prices. The primary operations involved in portfolio analysis are (1) scanning sequentially through individual assets, (2) screening of each asset against buy or sell criteria, (3) search of a mental "active list" for attractive buy candidates not yet in a portfolio, and (4) pairwise comparison of single stocks, usually stocks within the same industry. Again, most of the operations used are highly local in focus, with PM attention limited to a very small portion of the portfolio at any one time. The decision protocols revealed very few observations on the aggregate status or structure of an account. The over-all observations that are made usually involve the PM's noting the total common stock percentage of the portfolio, or the proportion of the account in a certain industry group.

Summary of the Current Decision System

In summary, the current PM decision process involves a great deal of intelligence or problem-finding activity, followed by a very local search for alternative solutions, considering potential asset transactions one at a time. Throughout the process, one is struck by the focus upon individual securities with little apparent perception of the status of a total portfolio as an entity.

It should be emphasized that the characteristics of the decision process described in this section are not at all unique to the particular group under study. They are similar to decision behavior in many other trust institutions, investment counselors, mutual funds, insurance companies, and individual investors. The decision process as described is not "good" or "bad" per se but represents a reasonable adaption by intelligent men to limitations and constraints in the information and processing systems available to them. In fact, the particular group under study here was achieving a very satisfactory rate of return on investment for the portfolios under their management. The question is whether or not that process can be aided so that performance can be improved further.

Throughout the descriptive model discussion above, problems were noted which represented apparent differences between observed behavior and the normative model developed earlier. The decision functions derived in the functional model phase described next represent an explicit attempt to cope with some of these problems in the current decision system.

5.5 Functional Model

The normative model of Section 5.3 identified desirable characteristics for each of the decision phases in the model. The subsequent description of the current decision system in Section 5.4 revealed some sharp departures from these desired characteristics. These gaps between the normative and descriptive models were defined above as problems. The functional model phase of the design methodology aims to carry the process one step closer to a designed system by identifying specific functions aimed at reducing these gaps between observed and desired decision behavior. This process of definition of formal decision system operators and operands is described in this section. The reader will recall from Chapter 3 that operators and memory are two fundamental components of a decision system. These correspond to the operators and operands (or data structures) defined below in response to previously identified problems in current decision system behavior.

Definition of Operands

The fundamental operand of interest is the portfolio, the entity being managed. The portfolio is a list of assets, primarily described by (1) size of holding, (2) cost, and (3) market value. There are other relevant attributes of the holding and of the asset held, but these are the most significant. Also involved in the system are lists of assets (e.g., the "approved list" of stocks, list of stocks in specific industry groups, etc.). Finally, any given PM manages a list of portfolios. Thus it appears that many of the relevant operands (or data files) in the problem can be thought of as lists, or lists of lists.

Therefore, the following four basic types of operand-lists are defined:

(1) stock list;*

(2) portfolio, a stock list with associated holding size, cost and value;

---

*It should be noted once again that this prototype design focused exclusively on the common stock portion of a portfolio.

(3) directory, a list of portfolios and/or stock lists;

(4) trade program, a list of stocks with associated numbers of shares, positive or negative indicating either a buy or a sell order.

Note that a single stock is also an operand by this definition, since it can be viewed as a stock list of length one. Similarly, the trade program list may represent a single transaction order. (E.g., a single record list specifying "-100 IBM" would indicate a transaction to sell 100 shares of IBM.)

Each stock included in the master "approved list" would also have an associated string of descriptors, the values for variables such as price, price-earnings ratio, historical earnings growth rate, etc. Thus, one might also think of a fifth list, a list of the stock or holding descriptor variables in the data base.

All of these operands are used by the PM in his current decision process with varying problems in accessibility and currency. However, since all are completely quantifiable, they are amenable to transfer to the computer in the prototype MMDS.

Definition of Operators

An initial set of operators for the prototype MMDS are identified below by a consideration of specific problems or requirements noted in Section 5.4 within the decision phases of intelligence, design, choice, and implementation (the control phase is discussed in Section 5.9, which describes the design of the decision control system).

In the intelligence phase, one of the major problems identified was the fact that portfolio status reports were out-of-date and had to be updated by hand. This suggests the need for a STATUS operator which causes the display of the current status of a portfolio or other list.

Also in the intelligence phase, another problem cited was the fragmentation of the data base into two separate files, one on portfolio holdings and the other on stock performance history and forecasts. This suggests the need for a TABLE operator that allows for the juxtaposition of portfolio holding information and investment research information in one tabular display. That is, the PM should see explicit values for such variables as earnings growth rate and price-earnings ratio associated with each stock in a portfolio, rather than trying to remember them or compute in his head their current value.

Also in the intelligence phase, and elsewhere in the decision process, another problem noted was the dearth of aggregate or overall measures of portfolio status and structure. There are several possible approaches to this problem. One would be an operator that might be called AGGREGATE, which would produce overall statistics for the portfolio (e.g., the mean and standard deviation of the yield across all of the stocks in the portfolio). Another would be a function called HISTOGRAM, which would produce a graphic sketch of a profile or histogram representing the distribution of the holdings of an account along some dimension of interest (e.g., estimated total return). Another graphic function would be SCATTER, which would produce a two-dimensional scatter plot of the components of a list for any two variables selected from the data base (e.g., price-earnings ratio versus estimated future earnings growth rate).

A further problem noted in the intelligence phase was the lack of any formal means for comparison of portfolio status with goals, or even with some accepted standard. This problem might be reduced by a function called COMPARE, which allows comparison of two or more lists (e.g., a given portfolio versus a "model portfolio", selected by policy committee or even by a portfolio selection algorithm). The COMPARE function could operate in conjunction with one of the other functions, such as HISTOGRAM or AGGREGATE (e.g., compare aggregate statistics of several lists, or compare overlaid histograms).

Another problem noted in the intelligence phase was the fact that all reports in the current system were rigidly formatted, with holdings listed by industry group only. Industry group may be an important attribute of a stock and of portfolio diversification, but stocks have other attributes of significance. This suggests a need for a function called ORDER which produces a list status report where the components of the list are sorted into ascending or descending order of value of any selected attribute (e.g., a portfolio report in order of increasing estimated total return could help to highlight problem holdings).

In the decision phase of design, one problem noted was the tendency to search very locally for stock buy-sell candidates. I.e., search usually considered mainly the issues already in a portfolio -- almost never was the whole list of nearly 350 approved stocks considered explicitly. This search could be broadened and made more rigorous by the use of a function called FILTER, which filters through a specified list based on several given criteria and displays those components which satisfy the criteria. For example, one could ask for all stocks in the approved list which have a price-earnings ratio of less than 15 and an estimated earnings growth rate of over 20 per cent.

Another problem cited in the design phase was the lack of any formal mechanism for constructing and viewing alternative portfolios rather than just alternative stocks. This suggests a need for a CREATE function, which would allow for the creation of hypothetical alternative portfolios. These might be created by using FILTER and setting cost, size, and market value for the components of the resulting list, or by making a hypothetical trade in an existing portfolio by specifying a trade program list. With CREATE, there should be a counterpart function called DELETE.

A further problem noted in the design phase was the slowness of response of the current system for reporting holdings of a given stock across all portfolios. This could be alleviated by a fast-response function called STOCK which would carry out the same action but in real time.

In the choice phase of decision making, the primary concerns were with lack of workable mechanisms for comparing or evaluating alternative portfolios as opposed to alternative stocks. This problem could be reduced by use of the functions already defined with the proviso that all functions operate equally well upon CREATE'd hypothetical portfolios as upon actual portfolios.

In the implementation phase, no severe problems were noted in the current process, although the speed of order transmission to the trader might be increased. This could be accomplished through an EXECUTE function which activated a specified trade program and transmitted it directly to the trader.

Control on the specified trade might then be accomplished by applying the STATUS functions to the now activated and pending trade program list, to see if it has been either partially or fully executed. (Control on the decision process itself will be discussed in Section 5.9.)

Thus far the following twelve functions have been specified:

1. STATUS - displays contents of a portfolio or other list.
2. TABLE - displays list contents plus selected attribute values.
3. AGGREGATE - displays aggregate statistics of a list.
4. HISTOGRAM - displays distribution of a list on a selected dimension.
5. SCATTER - displays contents of a list distributed on two dimensions.
6. COMPARE - compares the status of two lists.
7. ORDER - displays a list ordered on one attribute.
8. FILTER - filters a list for components meeting specified criteria.
9. CREATE - allows creation of a new list.
10. DELETE - allows deletion of an existing list.
11. STOCK - displays the holdings of a single stock across portfolios.
12. EXECUTE - causes executions of a specified trade order.

These clearly do not yet represent a well-defined command language, but rather a general definition of capabilities aimed at alleviating some of the specific problems identified in the current decision system. The list is not exhaustive and could easily be extended.

It is a matter of judgment as to when one stops in creating operators, and it was judged here that these eleven operators represented a substantial response to the decision problems noted.

It is fully expected that this process of design of a MMDS will be evolutionary, with new problems and hence new formal operators being defined as a result of experience with the decision system. In this situation, the author elected to stop short of designing functions which would be useful specifically in developing and using more complex and rigorous models for stock and portfolio selection and evaluation. It is fully expected that such a capability is desirable and, in fact, that it will be demanded by the PM as one logical next step beyond the MMDS functions defined above.*

Although some considerations of technical and economic feasibility are already implicit in the way problems and functions have been defined, such considerations have yet to be raised explicitly. That is the purpose of the design phases of search and constraining, described in the next section.

---

*One simple example of one function that would contribute to this modeling capability would be an operator called DEFINE, which would allow for the definition of new variables as algebraic and/or logical functions of existing variables in the data base. These DEFINE'd variables could then be used in all reports as if they were part of the data base.

An example of a function which would represent a step toward an evolutionary decision programming capability is one that could be called MACRO. MACRO would allow for the creation of a sequence of other functions which could then be recalled as a single command, with arguments as required.

5.6 Search and Constraining

A number of desirable operators and operands for a PM decision system were defined in the previous section. A further step in design of the MMDS is the specification of desirable operator-operand combinations. The matrix representing all possible such combinations is shown on the next page. One could view this matrix as a representation of a "search space" for a MMDS design. It raises the question of which intersections are feasible and/or desirable for a portfolio management MMDS. The design exercise of "constraining" involves the screening of possible MMDS functions and operator-operand combinations in light of constraints of limited technology, budgets, time, and human capability.

This explicit exhaustive representation of all pairwise combinations of defined operators and operands is useful in that it may suggest particular combinations that may have been ignored otherwise. For example, the HISTOGRAM function came to mind in the previous section as useful for displaying the structure of portfolios -- the matrix suggests the further idea of a histogram across a directory of lists. There are two possible forms to such a directory histogram:

(1) a distribution of the individual holdings of all stocks in all lists;

(2) a distribution of the values of a summary statistic across all -- lists (e.g., a distribution of the mean yield of all portfolios in the department).

	Operand Lists			
Operators	Stock List	Portfolio	Directory	Trade Program
STATUS	X̲	X̲	X̲	X
TABLE	X̲	X̲		X
AGGREGATE	X	X		X
HISTOGRAM	X̲	X̲		X
SCATTER	X̲	X̲		X
COMPARE	X̲	X̲		X
ORDER	X̲	X̲		X
FILTER	X̲	X̲		X
CREATE	X̲	X		X
DELETE	X̲	X		X
STOCK			X̲	
EXECUTE				

Legend

X: Feasible and Desirable

X̲: Actually Implemented

FIGURE 5.3

Initial Operator-Operand Matrix for the Prototype MMDS

The first distribution could be useful as a standard for comparison for individual portfolios; the second could be useful for highlighting groups of portfolios in extreme positions by some status measure. This particular operator-operand combination may have come to mind anyway, but the matrix representation helps to ensure that the designer conducts a wide search for potential MMDS capabilities.

On the other hand, some particular operator-operand combinations may seem practically meaningless. For example, STOCK was defined as a function which searched across a directory of lists, identifying those lists which hold a specified stock. In this context, application of STOCK to a single portfolio does not appear to have much meaning. However, with portfolios having a large number of holdings, there may, in fact, be some need for a function to establish quickly whether or not a particular portfolio holds a particular stock. Again, the exhaustive matrix representation serves as a stimulus to creative design thinking and operator generalization that might not readily occur without such explicit mechanisms.

For the particular matrix shown here, it happens that none of the operator-operand intersections are totally meaningless with the exception of some operand combinations with EXECUTE. Some combinations, however, are less easily implemented or used by the PM than others. Exercising some judgment as to the particular combinations which could be implemented within the limitations of the prototype

MMDS project, the intersections marked with an "X" were selected as being both feasible and desirable. In the end, this screening proved too generous and only those intersections with an underlined "X" were fully implemented.

It should be emphasized again that the phases of this design process should not be expected to occur in an orderly, distinct sequence. In fact, there is (and should be) much iteration among phases and much blurring of the boundaries between phases. The methodology is described here as distinct phases for clarity in presentation, and to indicate the activities and heuristics that should be useful in the design of MMDS. There should be no implication that the methodology is to be executed as a programmed activity.

In this particular case of the design of a prototype MMDS, there was a very high degree of iteration and feedback among the phases of decision modeling, functional modeling, constraining, and design. The PM's who were to use the system were kept involved during the design process, and their reactions contributed to this iteration and feedback.

5.7 Conversational Graph Specification

The specification of a conversational graph representation of the MMDS design passed through several iterations before it was complete. It began at a very general level of representation based directly on the operators specified in the functional model.

The graph shown on the next page is an example of such an initial general representation, before all of the details of man-machine conversation are specified. Note that this graph includes almost all of the defined functions of the previous section. The missing function are AGGREGATE, COMPARE, EXECUTE and ORDER. The AGGREGATE and EXECUTE functions were not actually implemented due to limitations in time and resources and are therefore not shown in the graph.

The COMPARE function was implemented to a limited extent within the HISTO and ACCOUNT functions. The HISTO function allows for the overlaying and comparison of distributions of two different stock lists or portfolios. The ACCOUNT function summarizes comparative totals of cost, and market value for all of the portfolios in the system.

The ORDER function has actually been merged with the TABLE function. That is, the tabular report on contents of a given list or portfolio produced by TABLE can be ordered on any selected stock attribute. The specific command "STATUS DIRECTORY" was implemented as simply "DIRECTORY". This function produces a list of all lists in the system.

From a general beginning such as shown in the figure, the conversational graph was expanded and modified through many interations, adding much more structure to each of the functions shown. For example, a small section of the graph for the HISTO function is shown in the next figure.

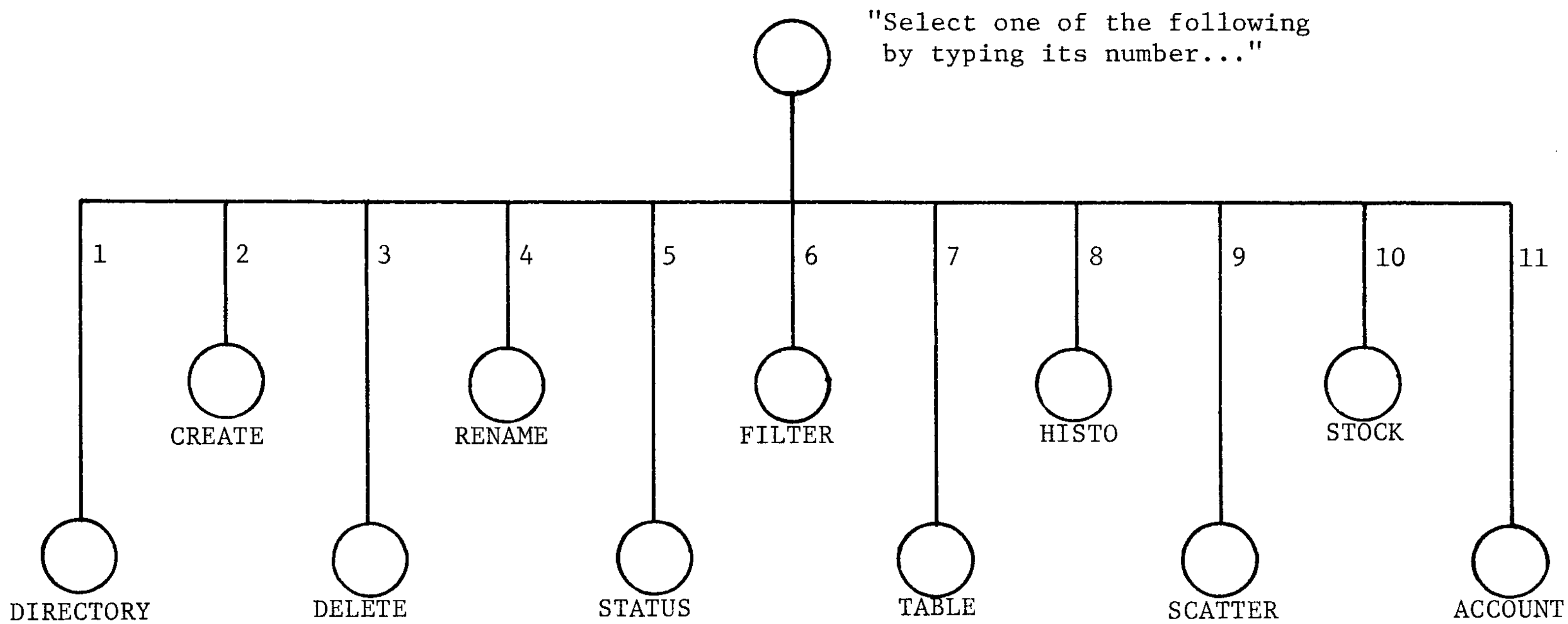


FIGURE 5.4

Top Node in the Prototype MMDS Conversational Graph

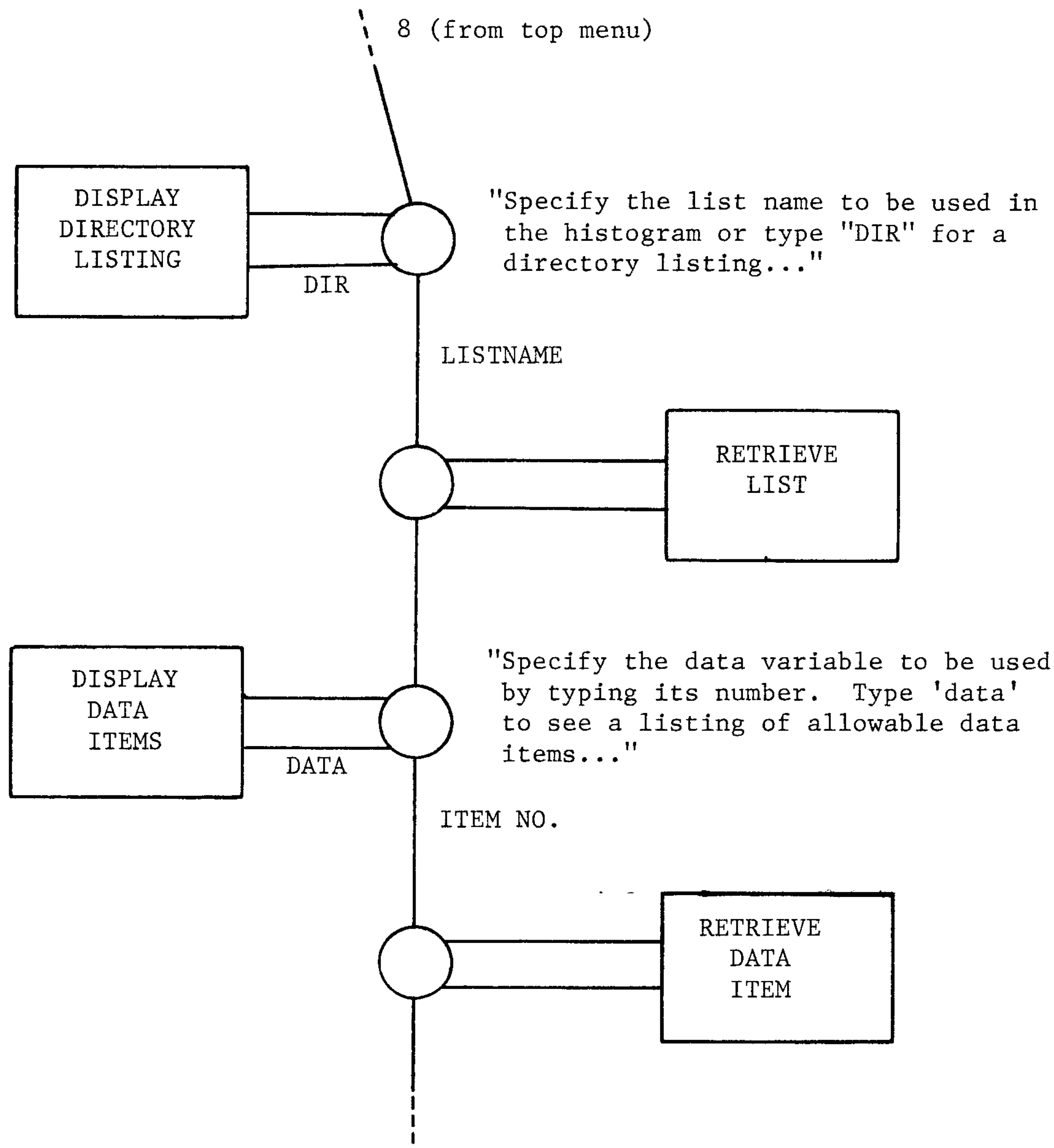


FIGURE 5.5

Sample Segment of Detailed Prototype Conversational Graph

The goal in developing the conversational graph was to structure a system that would allow the decision maker to do everything he might reasonably want to do with each function. The judgment as to just exactly what he "might reasonably want to do" was made by reference back to the goals expressed in the original normative model statement, the functional specification, and the descriptive model. Also the PM's themselves were consulted on some of these general design issues. Implicit in these detailed decisions were assumptions and expectations as to eventual MMDS behavior. Some of these assumptions are outlined in Section 5.10 and evaluated in Chapter 6.

The graph, as it developed, provided a convenient vehicle for mental simulation of MMDS use. That is, the designer could generate hypothetical man-machine scenarios to develop a feel for the logical structure and the mechanics involved in running the system. At this stage, rough drafts of display formats were also being designed, so that one could also begin to visualize the user's impression of the MMDS.

In this process of graph specification, there was a continual trade off being made between mechanical simplicity on the one hand and generality and complexity on the other: e.g., how many alternative actions can you offer a user at each stage of the man-machine conversation without hopelessly confusing him? There was an aim to keep the mechanics especially simple for functions whose use was expected to be high, but this was not always possible. There was a further aim to have the system somewhat adaptable to a variety of conversational paces, from the beginning user who may require considerable prompting to the experienced user who may want a direct and concise command language. There was also an aim to keep the system rather easy to modify, especially at the level of structure and content of the conversational graph, so that the mechanics of conversation could be "tuned" after some experience with system use. The way in which some of these partially conflicting aims were balanced is indicated in the next section summarizing the resulting MMDS design.

5.8 Detailed Design and Implementation

To reiterate, the aim in this project was to design and implement a prototype MMDS for portfolio management, not an operational system. The further purpose of the project was to exercise the MMDS design methodology proposed in the thesis, to demonstrate technological feasibility of such a MMDS, and to provide a vehicle for observation of experimental use of the prototype by a group of subject portfolio managers.

Some of the design aims were noted in the previous section. A further goal in the design process was to produce a prototype which was highly attractive in appearance to the user and which was capable of graphic information presentation. This suggests use of a cathode ray tube terminal as the man-machine interface. The terminal selected for the system was the ARDS (Advanced Remote Display Station), a storage tube graphic display device originally developed at M.I.T. The ARDS has full graphic capability, and the further advantages of high character capacity, quiet information presentation, and a relatively high character display rate even over voice-grade telephone lines. The ARDS had the further advantage of being fully supported under C.T.S.S., an M.I.T.-developed time-sharing system.

Another aim of the system was to be adaptable to various modes of conversational interaction. Hence the system provides for three distinct modes:

(1) Menu-selection, where the computer displays and explains all alternatives at each step in the conversation; the PM merely selects from the given alternatives.

(2) Stacked-selection, where the PM, after he knows the system well enough, can "stack" a sequence of selection commands on one line, thus skipping over all of the intermediate dialogue.

(3) Direct Command, where the PM can specify an operation and the relevant operands directly, with no prompting from the system, to cause immediate execution of the desired function.

The idea is that the user can operate at any of these levels, and that he would normally evolve from mode (1) to mode (3) with time and experience. The user does not take any particular action in switching from mode to mode; the system simply interprets any given user input line as being one of the three (or a combination). Thus the user can use direct and stacked commands for those parts of the system he knows well, but he can revert, in the same session, to the menu-selection mode while trying out an unfamiliar or new capability on the system.

Also, as a tactic to give the totally inexperienced PM something to work with immediately, the STATUS and STOCK functions were designed to be as simple as possible to control. They both produce portfolio or stock holding reports similar to ones the PM gets from his current information system, so it was felt that they would provide an easy transition into some of the more complex and less familiar functions.

Based on the detailed conversational graph system specification of the system, design proceeded through the programming and implementation phases. Programming was initiated in June, 1968, and the system was brought up for initial experimental use in July, 1969, after slippage of the original target date of December, 1968. The resources employed in the project involved approximately one man-year of analysis and general design and one man-year of detailed design and programming.

The appearance and operation of the resulting prototype MMDS are suggested by the scenario of seventeen displays shown on the following pages. These are reversed reproductions of photographs of conversational sequences on the ARDS screen. Each one of the displays is explained below. All human user input is shown in lower case characters; all system output is upper case.

Figure 5.6. The system has been called and it responds with a welcome message, the date and time, and a request for user identification. The user responds with his initials.

Figure 5.7. The top "menu" of alternative functions is displayed for initial selection by the user. In this case, alternative 1, DIRECTORY, is selected, resulting in the next display.

Figure 5.8. A directory of both portfolios and stock lists currently saved by the system is displayed. The stock lists were all created by an earlier user with initials DEM by filtering the full approved list on some specified criteria. The directory display also identifies the number of stocks in each list, the date the list contents were last modified on the system, and a short name for the list to be used in calling it for other operations. The user types a carriage return to return to the top node in the system conversational graph.

```
WELCOME TO THE COMPUTER AIDED PORTFOLIO MANAGEMENT SYSTEM.


TODAY'S DATE IS MAY 26, 1970
THE TIME IS 03:0  PM EST

PLEASE TYPE YOUR THREE INITIALS...

tpg
```

FIGURE 5.6

Initial Prototype "Sign On" Display

```
IF YOU WISH TO HAVE MORE INFORMATION ABOUT THE SYSTEM,
TYPE 'INFO', OTHERWISE...


SELECT ONE OF THE FOLLOWING BY TYPING ITS NUMBER OR THE NAME
WITHIN QUOTATION MARKS...


   1. PRINT 'DIRECTORY' OF LISTS      7. ORDERED 'TABLE' DISPLAY
   2. 'CREATE' A NEW LIST             8. FORM 'HISTO' GRAM
   3. 'DELETE' A LIST                 9. FORM 'SCATTER' PLOT
   4. 'RENAME' A LIST                10. 'STOCK' RUN
   5. REVIEW LIST 'STATUS'           11. 'ACCOUNT' MARKET VALUE
   6. 'FILTER' A LIST

... 1
```

FIGURE 5.7

The Top "Menu" of Alternative Functions

DIRECTORY OF CURRENT LISTS   DATE-   MAY 28, 1970

LIST		SYMBOL NAME	LAST MODIFIED	NUMBER OF ITEMS	INITIALS
ACCOUNT PORTFOLIOS					
1.	ABC FOOD CORPORATION PENSION T	ABCCOR	11 9 69	63	PM2
2.	ACME CHEMICAL COMPANY PENSION	ACME	11 9 69	50	PM1
3.	INTERSTATE MOVING COMPANY PENS	INTER	11 9 69	50	PM1
4.	SAMPLE PORTFOLIO ACCOUNT FOR S	SAMPLE	1 10 70	45	RAC
5.	UNITED PETROLEUM CORP PENSION	UNITED	11 9 69	63	PM2
STOCK LISTS					
1.	EARNINGS GROWTH OVER 15	EGRHI	1 12 70	20	DEM
2.	YIELD GREATER THAN 5 PERCENT	FILOUT	1 7 70	24	DEM
3.	HIGH EPS GROWTH IN ACME	HIEGRO	1 7 70	5	DEM
4.	HIGH DIVIDENDS, LOW PRICE	JAN5	1 5 70	16	DEM
5.	RANDOM LIST OF STOCKS	JAN7	1 7 70	7	DEM
6.	ERNIE'S TWO FILTER LIST	MIE	3 3 70	6	DEM
7.	SIX TEST STOCKS	OTHER	1 10 70	6	DEM
8.	HI EGR, LO PE	PE/EGR	1 10 70	10	DEM
9.	RHG TEST LIST		1 24 70	7	DEM

TYPE A CARRIAGE RETURN WHEN YOU ARE READY TO CONTINUE...

. . .

FIGURE 5.8

The Full Directory of Portfolios and Stock Lists

Figures 5.9 and 5.10. These displays present a status report on the Acme portfolio called from the top menu in the previous figure by the user's typing "5 acme". The report covers two pages of the ARDS screen and includes for each stock holding the shares, ticker symbol, name, recent price, cost, and market value, as well as a total value for the account on the last page. The user goes from one page to the next by striking an "erase" button on the ARDS keyboard. Note on the second page that the user has the option of (1) reviewing another list, (2) calling "DIR" to get a full directory report on all portfolios, or (3) hitting a carriage return to return to the first page of the same report. He may return to the top menu by typing "top".

CHEMICAL COMPANY PENSION TRUST
P/P.T 39999

PRICING DATE FEB 8, 1969
NUMBER OF ITEMS 50
PORTFOLIO MANAGER PM1

		COST	MARKET VALUE
1	CASH	404886	404885
2	CASH EQUIVALENT	2773300	87733000
3	PREFERRED STOCKS	2701034	4301350
4	BONDS	32030813	28226576
5	MISC ASSETS	10484005	8856086

1	20000	OT	OTIS ELEVATOR	47 0/8	668826	940800
2	45000	CTR	CATERPILLAR TRACTOR	39 6/8	456825	1788758
3	43050	GDC	GARDNER-DENVER	27 0/8	1009946	1162350
4	53716	MTC	MONSANTO	34 0/8	2314121	1826344
5	78000	STY	STERLING DRUG	43 2/8	1715816	3373580
6	25000	WLA	WARNER-LAMBERT	72 0/8	628197	1800000
7	34000	AB-	ANHEUSER-BUSCH	79 2/8	1509625	2694580
8	33334	KO	COCA-COLA	82 7/8	967345	2762555
9	51666	CFD	CONSOLIDATED FOODS	38 3/8	1616594	1982682
10	43000	GEB	GERBER PRODUCTS	39 1/8	1615844	1682375
11	55000	SB	STANDARD BRANDS	53 3/8	1797760	2935625
12	10000	TL	TIME INC	35 1/8	866887	351258
13	42120	EK	EASTMAN KODAK	79 7/8	1548100	3364335
14	16000	PRD	POLAROID	97 4/8	1671909	1560000
15	30000	PRX	PUREX	14 6/8	1064527	442588
16	40000	CLU	CLUETT PEABODY	25 4/8	779995	1028888
17	60000	KYR	KAYSER-ROTH	25 6/8	1993587	1545000
18	66000	GY	GENL TIRE	19 2/8	1691783	1278580
19	60000	TRW	TRW INC	30 6/8	1027344	1845888
20	72000	F	FORD MOTOR	42 3/8	3101676	3051888
21	43000	GM	GENL MOTOR	69 3/8	2560048	2983125
22	25000	EMR	EMERSON ELEC	63 4/8	966480	1587588
23	26000	GE	GENL ELECTRIC	70 4/8	2076869	1833888
24	34000	SMB	SUNBEAM	23 5/8	960095	803250
25	22000	HWP	HEWLETT-PACKARD	99 6/8	1418688	2194588
26	20500	LIT	LITTON INDUSTRIES	28 3/8	1485300	581687
27	7000	MOT	MOTOROLA	128 4/8	398541	899588
28	70000	RCA	RCA CORP	30 0/8	3408102	2100000
29	25000	TXN	TEXAS INSTR	126 5/8	2701589	3165625
30	31000	ZE	ZENITH RADIO	34 5/8	1204453	1073375
31	15000	CDA	CONTROL DATA	64 4/8	2052313	967500
32	22500	IBM	INTL BUSINESS MACHINE	340 2/8	3465795	7655625[illegible]

FIGURE 5.9

The Status Report on a Portfolio (Page 1)

33	20000	PBI	PITNEY-BOWES	36 7/8	695186	737500
34	15000	XRX	XEROX	91 0/8	3070494	1365000
35	53000	IPC	ILLINOIS PWR	34 0/8	1860097	1802000
36	91000	MSU	MIDDLE SOUTH UTILITIES	22 0/8	1349708	2002000
37	75000	NI	NORTHERN IND PUB SERV	27 2/8	1982173	2043750
38	55000	SCG	SO CAROLINA E + G	25 2/8	1020740	1408750
39	78000	SO	SOUTHERN CO	26 5/8	2125963	2076750
40	31000	TXU	TEXAS UTIL	58 5/8	1656000	1817375
41	46000	GAS	NORTHERN ILL GAS	29 3/8	1958563	1351250
42	32000	T	AMER TEL + TEL	51 2/8	1555270	1640000
43	10000	NWT	NORTHWEST INDS	15 4/8	1303909	155000
44	72000	AHS	AMER HOSPITAL SUPPLY	45 0/8	512063	3240000
45	42000	AMA	AMFAC	46 4/8	1502283	1953000
46	38684	CBS	COLUMBIA BROADCASTING	43 4/8	1546134	1682754
47	37000	DB-	DUN + BRADSTREET	55 3/8	1157280	2048875
48	90000	ITT	INTL TEL + TEL	59 0/8	4750857	5310000
49	32000	MMM	MINNESOTA MINING	109 0/8	1641266	3488000
50	41300	SNH	SPERRY + HUTCHISON	51 2/8	1880925	2116625

TOTAL - COMMON STOCKS	84197856	99562582
TOTAL - NON STOCKS	48394038	67521897
GRAND TOTAL	132591894	167084479

TYPE THE SHORT FORM NAME OF THE LIST WHOSE STATUS YOU WISH TO
REVIEW OR A CARRIAGE RETURN TO REVIEW THE CURRENT 'ACTIVE' LIST.
TYPE 'DIR' TO GET A DIRECTORY LISTING...

...

FIGURE 5.10

The Status Report on a Portfolio (Page 2)

Figure 5.11. Part of the dialogue necessary to CREATE a list is shown here. In the dialogue the user specifies that he would like to create a stock list, that he would like to build it by filtering, and that he would like to filter the approved list (called "CSL" here for Common Stock List). He is then asked which data items he would like to use in the filtering operation. Since he doesn't remember all of the data items, he types "data", which produces the next display.

Figure 5.12. A list of all of the stock-related data items in the data base plus associated item numbers are displayed. After refreshing his memory with this list, the user types "12" to indicate that he wants to filter on item 12, the three year best estimate of earnings growth rate.

Figure 5.13. The filtering operation continues with the user specifying filter limits on earnings growth of 15 to 1000 percent. He is told that 20 stocks pass this filter, and he asks to see what they are, thus producing the following display.

Figure 5.14. The stocks passing the filter are shown, plus the value of the data item being used in the filter (estimated earnings growth rate). The user is then asked if he wishes to save this filtered list for future reference, and he responds with "yes".

```
PLEASE SPECIFY THE TYPE OF LIST YOU WISH TO CREATE BY
SELECTING ONE OF THE FOLLOWING OR TYPE '4' FOR DEFINITIONS...
     1. HYPOTHETICAL PORTFOLIO
     2. STOCK LIST
     3. PORTFOLIO CHANGE PROGRAM
     4. DEFINITION OF TERMS

... 2

SELECT THE LIST BUILDING OPERATION YOU WISH TO APPLY...
     1. TYPE IN TICKER SYMBOLS
     2. FILTER AN EXISTING LIST OR PORTFOLIO OR THE CSL
     3. SELECT COMPONENTS FROM AN EXISTING LIST OR PORTFOLIO

... 2

TYPE THE SHORT FORM NAME OF THE EXISTING LIST YOU WISH TO
FILTER OR 'CSL' FOR THE COMMON STOCK LIST.  TYPE 'DIR' TO
GET A DIRECTORY LISTING...

... csl

SELECT UP TO 8 DATA ITEMS BY TYPING THEIR NUMBERS SEPARATED BY SPACES.
TYPE 'DATA' TO SEE A LISTING OF ALLOWABLE DATA ITEMS.

... data
```

FIGURE 5.11

Dialogue to Specify Creation of a New List

1 PRIMARY INDUSTRY CLASSIFICATION
2 SECONDARY INDUSTRY CLASSIFICATION
3 EXCHANGE CODE
4 STRATEGY CLASS (A=1, ETC)
5 VALUE CLASS
6 1968-9 PRICE HIGH
7 1968-9 PRICE LOW
8 LAST CLOSING PRICE
9 COMPANY NUMBER
10 EPS GROWTH - 3 YR. HIGH EST.
11 EPS GROWTH - 3 YR. LOW EST.
12 EPS GROWTH - 3 YR. BEST EST.
13 LONG RANGE GROWTH RATE EST.
14 EPS STD. DEV. 1959-68
15 EPS GROWTH AVE. 59-68
16 EPS GROWTH AVE. 60-69E
17 EPS GROWTH AVE. 64-68
18 EPS GROWTH AVE. 68-69E
19 EXPECTED DIVIDEND
20 NOT USED
21 NOT USED
22 INTERIM EPS LATEST 12 MOS.
23 NOT USED
24 FY EPS 1968
25 FY EPS 1969E
26 FY EPS 'NORMAL' 1969E
27 FY EPS TREND LINE 1969
28 NOT USED
29 10 YR. AVE. P/E 58-67
30 3 YR. P/E ESTIMATE
31 VOLATILITY INDEX
32 PERCENT OF SHARES OUTSTAND
33 RETURN ON EQUITY
34 NOT USED
35 NOT USED
36 EPS GROWTH 68E-69E
37 P/E - LATEST 12 MONTHS
38 P/E - 1968
39 P/E - 'NORMAL' 1969E
40 P/E - TREND LINE 1969
41 P/E 1969E
42 CURRENT YIELD
43 ROE VALUATION
44 RELATIVE ROE VALUATION
45 ANNUAL TOTAL RETURN EST.-3 YR.
46 MARKET VALUE
47 COST
48 NUMBER OF SHARES
49 PERCENT OF ACCOUNT
50 UNREALIZED GAIN (P.C.)

SELECT UP TO 8 DATA ITEMS BY TYPING THEIR NUMBERS SEPARATED BY SPACES.
TYPE 'DATA' TO SEE A LISTING OF ALLOWABLE DATA ITEMS.

... 12

FIGURE 5.12

Digression to Check the Data Items Available

```
FOR EACH OF THE FOLLOWING DATA ITEMS, SPECIFY A LOWER BOUND
AND A UPPER BOUND ON ACCEPTABLE VALUES SEPARATED BY SPACES.


IF YOU WISH TO SPECIFY PERCENTILE LIMITS, TYPE THE TWO LIMITS
FOLLOWED BY THE LETTER 'P'.
EXAMPLE., TYPE ' 10.0  90.0  P ' TO OBTAIN 10TH THROUGH 90TH PERCENTILE.


EPS GROWTH - 3 YR. BEST EST.          IN UNITS OF PERCENT
USING THE FORMAT XXX.X      XXX.X  FOR LOWER AND UPPER BOUNDS
FOLLOWED BY 'P' IF BOUNDS ARE PERCENTILES IN THE FORM XXX.X

...  15.0 1000.0

THE NUMBER OF STOCKS PASSING THROUGH ALL FILTERS SPECIFIED SO FAR IS...

 80

ALL FILTERS SPECIFIED HAVE BEEN COMPLETED...


DO YOU WISH TO REVIEW THE LIST OF STOCKS OBTAINED SO FAR...

...  yes
```

FIGURE 5.13

Continuing to Create a List by Filtering

THE LIST OF STOCKS PASSING ALL OF THE FILTERS SPECIFIED SO FAR
IS AS FOLLOWS:

COLUMN	DATA ITEM	UNITS
1.	EPS GROWTH - 3 YR. BEST EST.	PERCENT

1	AHS	AMER HOSPITAL SUPPLY	15.0
2	AVP	AVON PRODUCTS	17.0
3	BAX	BAXTER LABORATORIES	16.0
4	BDX	BECTON DICKINSON	17.0
5	BCC	BOISE CASCADE	15.0
6	BGH	BURROUGHS	17.0
7	CDA	CONTROL DATA	20.0
8	EFU	EASTERN GAS AND FUEL	15.0
9	ECK	ECKERD DRUG OF FLA	18.0
10	EAF	EMERY AIR FREIGHT	15.0
11	GP	GEORGIA-PACIFIC	15.0
12	HWP	HEWLETT-PACKARD	15.0
13	IBM	INTL BUSINESS MACHINE	15.0
14	KG	KRESGE S S	15.0
15	MES	MELVILLE SHOE	15.0
16	OXY	OCCIDENTAL PETROLEUM	15.0
17	PRD	POLAROID	16.0
18	TXN	TEXAS INSTR	15.0
19	UC-	UNIVERSITY COMPUTING	22.0
20	XRX	XEROX	20.0

DO YOU WISH TO GIVE THE INTERMEDIATE LIST JUST TYPED A NAME,
THEREBY SAVING IT FOR FUTURE REFERENCE...

... yes

FIGURE 5.14

The New List Resulting from Filtering

Figure 5.15. Starting from the top menu again, this time the user ignores the menu-selection form and types the direct command, "histo acme". This calls for a histogram display on the Acme portfolio. The user now reverts to the menu selection mode of interaction, and he specifies estimated earnings growth rate as the variable on which he wants the distribution. The system then asks how he would like to have the portfolio distributed: by percent of market value, etc. He responds in the stacked-selection form with "1 7", since he remembers that selection 7 in the following menu produces the histogram display.

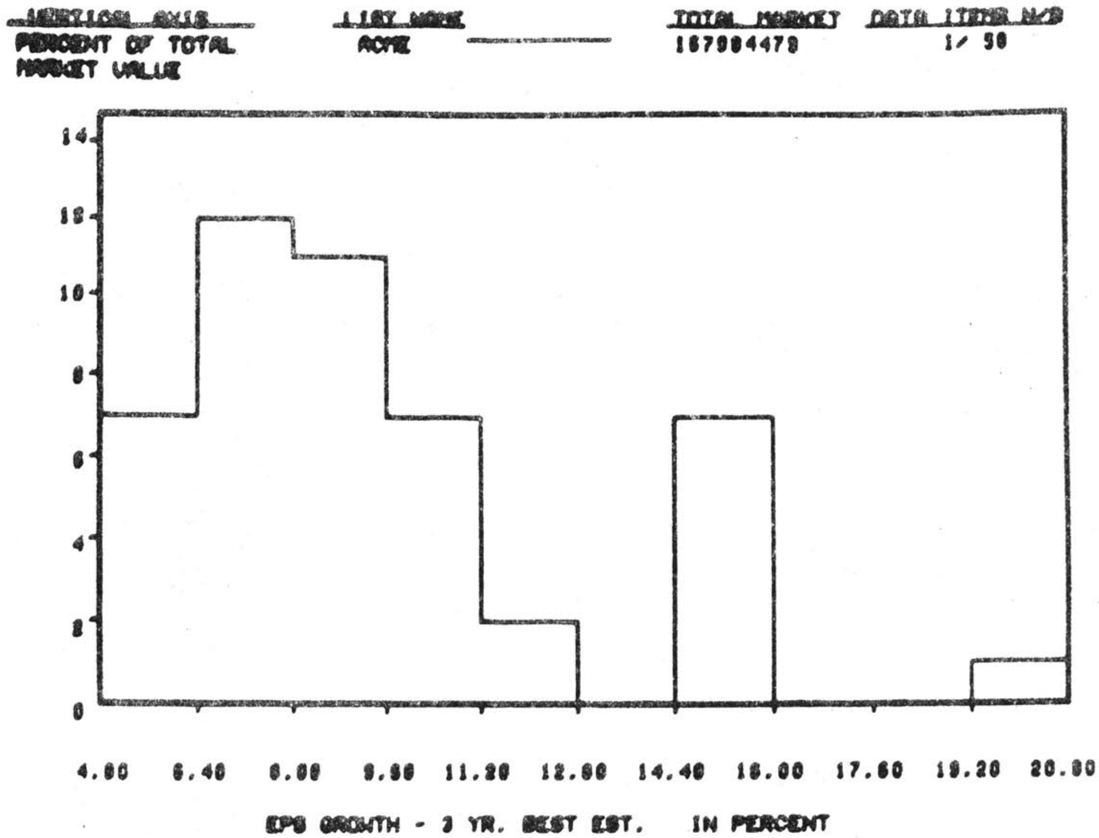
Figure 5.16. The resulting histogram display is shown for Acme, a portfolio with market value of $167M. The user has several ways available to modify the display, as shown in the menu at the bottom. He elects to specify different last names and, using the stacked command form, calls for a histogram on both Acme and Inter (a much smaller portfolio).

Figure 5.17. The comparative distributions of estimated growth rates for the Acme and Inter portfolios are shown here. The user now types the direct command "table" to get the following display.

```
SELECT ONE OF THE FOLLOWING BY TYPING ITS NUMBER OR THE NAME
WITHIN QUOTATION MARKS...

     1. PRINT 'DIRECTORY' OF LISTS   7. ORDERED 'TABLE' DISPLAY
     2. 'CREATE' A NEW LIST          8. FORM 'HISTO' GRAM
     3. 'DELETE' A LIST              9. FORM 'SCATTER' PLOT
     4. 'RENAME' A LIST             10. 'STOCK' RUN
     5. REVIEW LIST 'STATUS'        11. 'ACCOUNT' MARKET VALUE
     6. 'FILTER' A LIST

...  histo acme

     ACME        ACME CHEMICAL COMPANY PENSION TRUST

     THE ABOVE NAMED LISTS WILL BE USED IN THE HISTOGRAM...



     SPECIFY THE DATA VARIABLE TO BE USED BY TYPING ITS NUMBER.
     TYPE 'DATA' TO SEE A LISTING OF ALLOWABLE DATA ITEMS...

...  12

     THE FOLLOWING DATA VARIABLE WILL BE PLOTTED IN THE HISTOGRAM...

     EPS GROWTH - 3 YR. BEST EST.

     SELECT ONE OF THE FOLLOWING Y-VARIABLE OPTIONS...
          1. PERCENT OF MARKET VALUE     3. PERCENT OF STOCKS IN LIST
          2. MARKET VALUE IN $1000       4. NUMBER OF STOCKS IN LIST

...  1 ?
```

FIGURE 5.15

Dialogue to Specify a Histogram Display



SELECT ONE...

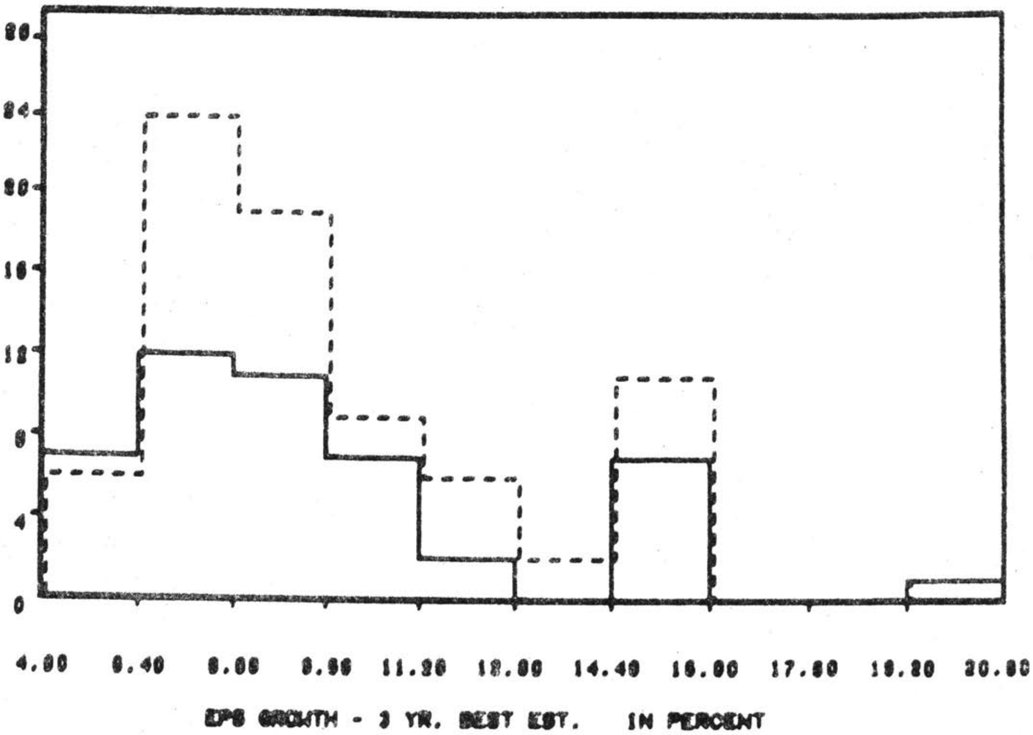
1. SPECIFY DIFFERENT LIST NAMES
2. SPECIFY DIFFERENT DATA VARIABLE
3. SPECIFY DIFFERENT Y MEASURE
4. SELECT DIFFERENT BOUNDS ON X
5. DISPLAY STATISTICS,DECILES
6. REVIEW CURRENT PARAMETERS
7. DISPLAY HISTOGRAM
8. QUIT HISTOGRAM MODE

... 1 acme inter 7

FIGURE 5.16

Histogram Display on One Portfolio

VERTICAL AXIS	LIST NAME		TOTAL MARKET	DATA ITEMS N/A
PERCENT OF TOTAL	ACME	————————	187084478	1/ 38
MARKET VALUE	INTER	- - - - - - -	84388826	4/ 38



SELECT ONE...

1. SPECIFY DIFFERENT LIST NAMES
2. SPECIFY DIFFERENT DATA VARIABLE
3. SPECIFY DIFFERENT Y MEASURE
4. SELECT DIFFERENT BOUNDS ON X
5. DISPLAY STATISTICS,DECILES
6. REVIEW CURRENT PARAMETERS
7. DISPLAY HISTOGRAM
8. QUIT HISTOGRAM MODE

... table

FIGURE 5.17

Histogram Display on Two Portfolios

Figure 5.18. The user begins a dialogue to specify the TABLE function. He selects the Acme portfolio as the operand and specifies that the report be sorted in order of increasing estimated earnings growth. He then specifies the four variables he would like to see in the report, which is shown in the next two displays.

Figures 5.19 and 5.20. The TABLE report on Acme is shown in these two displays with values for estimated earnings growth, 10 year average price-earnings ratio, latest 1969 price-earnings ratio, and estimated total return. The stocks in Acme are sorted into ascending order of estimated earnings growth; any stock for which no value of earnings growth was available is placed first in the list. Note that after viewing this display, the user types "scatter acme" to produce the scatter plot in the following figure.

```
THERE ARE THREE COMPONENTS REQUIRED IN ORDER TO SPECIFY AN ORDERED
TABULAR DISPLAY.  THEY ARE AS FOLLOWS...

     1. AN 'ACTIVE' LIST OF STOCKS FOR WHICH DATA WILL BE DISPLAYED.
     2. A DATA ITEM ON WHICH THE LIST WILL BE ORDERED  (E.G. 1987 P/E).
     3. UP TO 8 DATA ITEMS WHOSE VALUES ARE TO BE DISPLAYED FOR EACH
        STOCK IN THE ACTIVE LIST.


TYPE THE SHORT FORM NAME OF THE LIST YOU WISH TO USE IN THE TABLE
OR A C.R. FOR THE 'ACTIVE' LIST.  TYPE 'DIR' TO GET A DIRECTORY LISTING...

...  acme

     ACME         ACME CHEMICAL COMPANY PENSION TRUST


TYPE THE NUMBER OF THE DATA ITEM WHICH SHOULD BE USED IN ORDERING THE
ACTIVE LIST.
TYPE A CARRIAGE RETURN FOR ALPHABETICAL ORDER.
TYPE 'DATA' TO GET A LISTING OF ALLOWABLE DATA ITEMS.

...  12

SELECT UP TO 8 DATA ITEMS BY TYPING THEIR NUMBERS SEPARATED BY SPACES.
TYPE 'DATA' TO SEE A LISTING OF ALLOWABLE DATA ITEMS.

...  12 26 41 48
```

FIGURE 5.18

Dialogue to Specify a Table Display

COLUMN	DATA ITEM	UNITS
1.	EPS GROWTH - 3 YR. BEST EST.	PERCENT
2.	10 YR. AVE. P/E 58-67	NORMAL UNITS
3.	P/E 1969E	NORMAL UNITS
4.	ANNUAL TOTAL RETURN EST.-3 YR.	PERCENT

1	SNH	SPERRY + HUTCHISON	0.	0.	0.	0.
2	OT	OTIS ELEVATOR	5.0	17.9	10.2	16.2
3	F	FORD MOTOR	5.0	15.4	8.0	22.2
4	GM	GENL MOTOR	5.0	14.1	12.0	17.2
5	GAS	NORTHERN ILL GAS	5.5	10.6	11.7	21.3
6	T	AMER TEL + TEL	5.5	18.3	12.0	20.7
7	GDC	GARDNER-DENVER	6.0	12.6	10.0	15.1
8	SMB	SUNBEAM	6.0	18.9	14.3	15.2
9	IPC	ILLINOIS PWR	6.0	19.1	12.6	22.0
10	SCG	SO CAROLINA E + G	6.5	20.5	13.2	23.0
11	SO	SOUTHERN CO	6.5	21.0	15.1	18.0
12	MTC	MONSANTO	7.0	10.0	9.7	30.7
13	GEB	GERBER PRODUCTS	7.0	19.6	20.5	5.5
14	SB	STANDARD BRANDS	7.0	19.4	19.7	6.0
15	KYR	KAYSER-ROTH	7.0	12.3	8.1	27.0
16	GT	GENL TIRE	7.0	13.5	8.1	44.3
17	GE	GENL ELECTRIC	7.0	26.0	16.2	23.6
18	MSU	MIDDLE SOUTH UTILITIES	7.0	20.5	15.1	20.0
19	NI	NORTHERN IND PUB SERV	7.0	19.0	14.7	23.0
20	NWT	NORTHWEST INDS	7.0	0.	31.0	25.0
21	WLA	WARNER-LAMBERT	7.5	22.2	30.0	-7.0
22	TXU	TEXAS UTIL	7.5	25.6	19.8	13.0
23	CTR	CATERPILLAR TRACTOR	8.0	17.0	15.0	11.4
24	TL	TIME INC	8.0	17.2	11.7	40.2
25	CLU	CLUETT PEABODY	8.0	13.5	11.6	23.5
26	ZE	ZENITH RADIO	8.0	25.0	13.6	29.1
27	CBS	COLUMBIA BROADCASTING	8.0	18.0	16.7	12.6
28	DB-	DUN + BRADSTREET	8.0	21.6	27.6	-3.2
29	RCA	RCA CORP	8.5	23.0	11.7	32.7
30	AMA	AMFAC	8.5	16.4	20.6	9.2
31	CFD	CONSOLIDATED FOODS	9.0	18.4	19.6	14.2
32	EK	EASTMAN KODAK	9.0	30.4	30.7	6.2
33	TRW	TRW INC	9.0	16.2	13.0	25.2
34	PBI	PITNEY-BOWES	9.0	26.1	26.3	6.3
35	STY	STERLING DRUG	10.0	24.3	30.0	4.5
36	PRX	PUREX	10.6	25.4	9.6	43.9
37	EMR	EMERSON ELEC	10.0	19.9	30.9	-1.0
38	LIT	LITTON INDUSTRIES	10.0	40.3	11.6	55.3
39	MOT	MOTOROLA	10.0	25.0	22.3	10.25

FIGURE 5.19

Four-Variable Table Display on One Portfolio (Page 1)

40	MMM	MINNESOTA MINING	10.0	36.6	33.0	6.2
41	KO	COCA-COLA	11.0	27.7	30.5	3.0
42	ITT	INTL TEL + TEL	12.0	19.5	20.7	12.4
43	AB-	ANHEUSER-BUSCH	12.5	16.1	30.6	2.4
44	HWP	HEWLETT-PACKARD	13.0	41.0	51.1	.2
45	TXN	TEXAS INSTR	13.0	40.4	42.2	0.3
46	IBM	INTL BUSINESS MACHINE	15.0	40.0	39.5	13.5
47	AHS	AMER HOSPITAL SUPPLY	15.0	33.0	60.0	-1.0
48	PRD	POLAROID	16.0	62.2	46.4	17.7
49	CDA	CONTROL DATA	20.0	0.	16.1	62.3
50	XRX	XEROX	20.0	57.4	46.0	16.5

SELECT ONE OF THE FOLLOWING (YOU MAY CHOOSE OTHERS AFTER IT IS COMPLETED).

1. REVIEW THE SET OF COMPONENTS SPECIFYING THE TABULAR DISPLAY.
2. SELECT A NEW ACTIVE LIST.
3. SELECT A NEW DATA ITEM TO BE USED IN ORDERING THE LIST.
4. SELECT A NEW SET OF DATA VARIABLES TO BE TABULATED.
5. DISPLAY THE TABLE WHICH HAS MOST RECENTLY BEEN SPECIFIED.
6. QUIT TABLE MODE

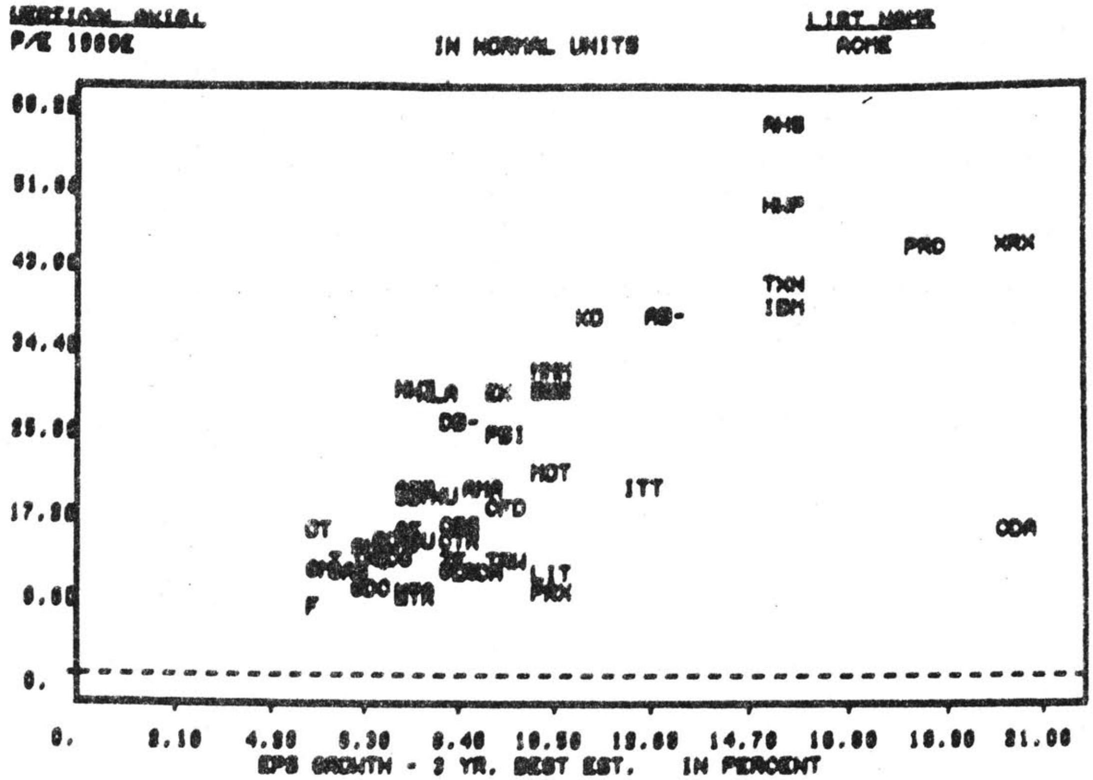
... scatter comp

FIGURE 5.20

Four-Variable Table Display on One Portfolio (Page 2)

Figure 5.21. This display is the result of the user asking for the SCATTER function applied to the Acme portfolio with holdings (indicated by ticker symbols) plotted in the two-dimensional space of earnings growth rate versus price-earnings ratio. Note that the user has a menu of alternative ways to modify the display, including the option to change scale and to blow up some of the currently cluttered regions of the display. Instead, he types "stock ibm" to produce the following display.

Figure 5.22. In this display, the STOCK function reports all accounts holding IBM. For each holding, the report specifies number of shares, cost, market value, per cent of the total portfolio, and per cent of the common stock portion of the portfolio. The user then types "xrx", the ticker symbol for Xerox, the next stock he wishes to review.



SELECT ONE...

1. SELECT A DIFFERENT NAME
2. SPECIFY DIFFERENT DATA VAR.
3. SPECIFY DIFFERENT X BOUNDS
4. SPECIFY DIFFERENT Y BOUNDS
5. DISPLAY STATISTICS, DECILES
6. REVIEW RECENT PARAMETERS
7. DISPLAY SCATTER DIAGRAM
8. QUIT SCATTER MODE

... stock ibm

FIGURE 5.21

Scatter Plot on One Portfolio

ACCOUNT HOLDINGS SUMMARY   STOCK PRICE 340 2/8
IBM   INTL BUSINESS MACHINE   AT CLOSING ON FEB 8, 1988

	ACCOUNT	SHARES	COST	MKR. VAL.	GAIN	PC TOT	PC STK
1.	ACME	22500	3465795	7655625	120.9	4.6	7.7
2.	ABCCOR	4200	674482	1429050	111.9	3.0	2.8
3.	INTER	3338	482330	1135754	135.5	4.7	5.1
4.	UNITED	2000	319404	680500	113.1	4.0	2.9
	TOTALS	32038	4941951	10900929	120.6		

TYPE THE TICKER SYMBOL OF THE STOCKS WHOSE HOLDINGS YOU WISH TO REVIEW...

... HPX

FIGURE 5.22

Stock Run Report on One Stock

These are a sampling of a few of the possible displays and man-machine dialogues possible with the prototype system. Some of the conversation shown may have seemed verbose; but on an ARDS with a 110 character per second writing rate, the exchange generally proceeds at a rapid pace. If this system were to operate on a teletype-like terminal at 10-14 characters per second, however, the slowness of output would be intolerable for most use.

One aspect of the prototype system that has not been discussed as yet is the design of controls on system usage. The control system design will be described next, followed by an outline of the designer's projections of expected prototype user behavior.

5.9 The MMDS Control System

The explicit control mechanisms designed for the prototype MMDS will be summarized in this section. However, the section will begin below with a digression to review problems encountered in the field experiment as it was originally conceived. The remainder of the section will review the planned instruments of the control system: the decision monitor facility, the Role Construct test, questionnaire and accounting data, and interview. Some of these instruments were dropped from the planned control system due to the problems described below.

Problems in the Original Study Design

The original aim of the study was to observe two groups of portfolio managers in parallel: one using the MMDS to manage several of their portfolios, the other using the current information supports and serving as a control group. There were three PM subjects and four controls for a total of seven PM's under study. Five portfolios each for the three subjects were to be monitored; each subject was to have three portfolios on the MMDS and two portfolios supported as before. The aim of monitoring the two portfolios off the system was to check for any transference of PM behavior or portfolio activity from the three accounts on the MMDS.

Each of the four controls was to have two portfolios each monitored. The object here was to attempt to control for section-wide or department-wide changes in PM behavior that might be confused with behavioral changes due to the MMDS.

Each PM subject was to be free to use the MMDS if and when he chose. Thus, any significant MMDS usage over a period of time would, in itself, represent a strong indication that the system was useful and attractive as a decision aid. In other words, the PM's were sufficiently busy with the pressures of their jobs that any "new toy" attractions of the system were expected to be very short-term.

The basic aim was for the MMDS to simulate a fully operational decision aid, insofar as possible and given the limited account coverage. Several factors which intervened to make these original aims infeasible are described below.

One, the development and programming of the prototype system slipped beyond the original target start-up date by seven months (December, 1968, to July, 1969). This increased the exposure of the original subject group to reorganization and transfer.

Two, during the period of MMDS delivery slippage and the experimental system usage that followed (a span of 15 months), three of the four controls and two of the three subjects either left the organization or were transferred for a variety of reasons. Most of this activity occurred around the time the MMDS was being put into initial use. Of the original seven portfolio managers, one control and one subject remained. The one subject who remained had two of his three accounts on the system transferred to another manager. Since all data collection prior to MMDS delivery had been focused upon these original seven portfolio managers and the specific accounts they managed, the net effect of all of these transfers was to make some of the planned before-and-after studies impossible.

Three, the ability to simulate an on-going operational MMDS was eliminated by unanticipated difficulties in gaining access to the prototype system, and with slow response time once connected to the system. These problems occurred in part because of the coincidence of initial use of this system with a significant increase in load on the CTSS system at M.I.T. This overload resulted when one I.B.M. 7094 was removed leaving only one remaining to handle the load originally supported by two systems. The heavy load meant that for much of the working day CTSS was loaded to its maximum of 30 simultaneous users; hence no new users could sign on. Even if one could get on the system, the response time was excessive. Typical response time during the day could be an average of 15 seconds with many responses taking several minutes. To say the least, a busy portfolio manager finds this sort of response unacceptable. There were additional factors affecting system access which compounded these response time problems: (1) there are only three high speed ports on the 7094 for ARDS usage; there are at least 10 ARDS terminals in use at M.I.T.; hence there was significant competition for and "hoarding" of those three ARDS lines to the computer; (2) there are a limited number of tie lines from the outside switchboard to M.I.T.'s internal computer PBX; these frequently were all busy when the subject users dialed in from outside. The net result was that the system could only be used effectively before 9:00 AM and after 6:00 PM. The subject portfolio managers found these effective hours highly inconvenient.

Thus, although some Trust Department users logged a considerable number of hours at the terminal, it was never viewed as an operational aid, but only an experimental demonstration of feasible capabilities in some future operational system. As such, however, it did catch the imagination of the user, and it proved to be a creative vehicle for gaining insight into early user behavior and MMDS control mechanisms that could be useful for design of operational MMDS.

It seems that such experience in field studies is not entirely unusual. Sackman (1969) recently concluded a massive study of the effects of man-machine interaction upon computer sciences education using 415 subjects, all students at the U.S. Air Force Academy. Unfortunately, the results of this very well designed experiment are of limited value because the time sharing system he employed was delivered late and was not fully debugged when it was put into use in the study. Hence the comparison turned out to be between a highly reliable batch system and an unreliable timesharing system. As Sackman notes:

> [This]...illustrates one of the pitfalls of real-world experimentation. Although we succeeded in controlling most critical variables toward an objective comparison of time-sharing and batch, and although we succeeded in designing and implementing the experiment right into the 'real-world' operation of computer science courses at the Academy, the timing of the experiment was such that one of the key control variables could not be rigorously maintained.
>
> (Sackman, 1969, p. 48)

The Decision Monitor Control Mechanism

The primary formal control mechanism for the prototype was a user monitor capability that was designed into the system. The system itself keeps a complete trace of all man-machine "conversations" including portfolios and data items accessed, as well as operations used. In an operational system this mechanism would provide a basis for analysis of evolution of MMDS behavior over time. In the case of prototype usage, the monitor "traces" allow for observation of the initial transient behavior of the subject portfolio managers.

The monitor system has at least two important practical applications in a MMDS: (1) as a debugging aid; (2) as a resource allocation guide. The recorded time trace of early user sessions can highlight disfunctional or incorrect use of the system and thus aid in the debugging of human factors aspects of the system, as well as actual bugs. For example, in the prototype system, users were often asked to "TYPE A C.R." (a carriage return) to designate a next step. Early monitor traces showed users typing the characters "A C.R." several times before finally typing "TOP" to escape their dilemma. The system, of course, had been responding that it did not understand the user input and that the user should try again. Simply spelling out "A CARRIAGE RETURN" in the text of much of the system dialogue eliminated the problem.

It was found that such a monitor trace was valuable even when a user called the attention of the designer to a bug that he had encountered. In the majority of such cases, the user could not remember enough of the specifics of the problem, nor of his actions leading up to the problem, to provide even a workable definition of the bug. Since the ARDS has no hard copy output, many of these problem situations would have been impossible to reconstruct without the trace record. The trace also revealed man-machine problems that were never reported to the designer. Since the behavior of a MMDS depends largely upon the highly impredictable actions of the human user, a monitor trace facility is an invaluable aid for finding and reconstructing system problems for debugging.

The monitor facility also has practical use in guiding system resource allocation. The monitor can accumulate historical statistics on function and file usage. These may suggest further investment in design, and programming effort, or even in hardware, to make highly utilized functions more efficient. The very simple monitor function built into the prototype system, however, did not collect such statistics directly; it merely maintained a complete trace of all user inputs and the sequence of modes traversed in the conversational graph. From this raw trace, all aspects of the man-machine decision protocol could be reconstructed except the specific values of items retrieved out of the system data base.

The Role Construct Repertory Test

The monitor facility was aimed at control over the outward aspects of man-machine decision behavior. This does not, however, aid directly in the detection and analysis of changes in the PM's understanding or perception of his decision task. The original plan was to use repeated applications of Kelly's Role Construct Test for the detection of changes in content and complexity of PM perception of stocks and portfolios with prototype system usage. As noted in Section 5.4, a baseline test was given to six PM's in early 1969 as a first step in this process. Unfortunately, the very limited system use actually achieved (due to the problems in system access and the loss of many subjects and controls mentioned earlier) meant that the system was unlikely to have much impact on task perception -- hence this control plan was dropped. The work of Wilcox (1970) and others, however, suggest that this test may prove a valuable tool in decision system design and control, and it deserves further evaluation in the context of similar studies.

Questionnaire and Accounting Data

As further controls on decision frequency and performance, the original intention was to continue collection and analysis of questionnaires and trust accounting data on the 23 portfolios in the study throughout the period of simulated operational usage of the system. Again, since simulated operational usage was never achieved, these control instruments also were dropped. The original intention was to use these instruments to monitor the following characteristics of the PM decision process for these 23 portfolios:

1. Decision Frequency -- the number of stock transactions per period.
2. Decision Magnitude -- the amount of stock trading per period, in number of holdings and market value, relative to the size of each portfolio.
3. Intelligence Type -- the proportion of need versus opportunity stimulated decisions.
4. Portfolio Review Character -- the frequency and duration of portfolio reviews.
5. Portfolio Performance -- the period-by-period time weighted rate of return on each portfolio.

With data on the above variables both before and during MMDS usage, the aim was to provide a further check on hypotheses about MMDS behavior. The intention is, in fact, that some of these variables will be monitored through the impending installation and use of an operational MMDS at the trust institution of this study.

Interviews and Direct Observation

Although most of the original formal control mechanisms designed for the system were dropped, the less formal mechanisms of periodic interviews and direct observation of MMDS use were continued through the period of limited prototype usage. The interviews focused primarily on the user's perception of prototype system effectiveness as a decision aid and his suggestions for changes and additions to improve the system. It is these interviews, some direct observation of PM usage of the prototype, and the monitor traces which form the base from which the observations on system behavior and impact of Chapter 6 are drawn.

Before discussing system use, however, the next section summarizes many of the design assumptions and behavioral hypotheses that underlay the detailed design decisions implied in Sections 5.5 - 5.8 that resulted in the prototype MMDS. These assumptions and hypotheses, in fact, represent the designer's model of expected MMDS behavior. The next step, in Chapter 6, will be to compare these expectations with observations on actual usage.

5.10 Expected Decision System Behavior

During the course of decision modeling and design of the prototype MMDS, many detailed design decisions were made based on a growing body of implicit design assumptions and expectations as to eventual system behavior. This is typical of the process of design. However, in this case, the designer took the somewhat unusual step of trying to make explicit as many of these assumptions as possible. That is, prior to and during the process of design, explicit hypotheses and assumptions were recorded for comparison with actual prototype system behavior.

One component of the philosophy that lies behind the MMDS design methodology proposed in Chapter 4 is the belief that a formal "memory" for past design assumptions is critically important if designs and design methods are to improve. In asserting the value of formal control based on design assumptions and models, our arguments parallel those of Carroll and Zannetos (1966) in advocating "operating process control". Similarly, we argue for "design process control" which is analogous to their "planning process control"; the design methodology proposed in this thesis represents an initial working model on which to base such design process control.

A formal memory for design assumptions is especially important where a high rate of evolution in system structure is expected, for then the design mechanism must learn most rapidly from design experience. Hence a formal memory for expected design behavior is particularly valuable for MMDS, which we fully expect to be in a state of continual redesign, insofar as it is aimed at complex and unstructured decision problems.

The expectations that developed during the course of decision analysis and early design of the prototype MMDS are summarized below. They are grouped within the following general headings, which are the elements of the general decision system framework developed in Chapter 3:

A. Decision Process Phases

1. Intelligence
2. Design
3. Choice
4. Implementation
5. Control
6. Decision Structure

B. Decision System Components

1. Memory
2. Operators
3. Plans

C. Characteristics of Man-Machine Interaction

1. Mode of Interaction
2. Structure of Interaction
3. Language and Form of Interaction
4. Flexibility of Interaction

D. Decision System Adaption

1. Learning and Training
2. Decision Programming

Each of the following sections contains a general discussion of the relevant design assumptions and expectations, followed by a concise list of the hypotheses implicit in the discussion.

Intelligence

One aim of the prototype system was to give the PM more flexible and powerful ways to examine portfolio status, as opposed to stock status. Hence one expectation was that portfolios on the system would be reviewed more often, employing a greater variety of tools and representations than before, when the PM was severely limited by his information system. It is recognized, however, that a higher review frequency may be a short-run phenomenon, due to initial discovery of a great many new problems in portfolios with the analytic tools of the system. A related expectation was that the PM would make particularly high use of those MMDS facilities which gave him an overall structural or aggregate view of his portfolio (e.g., HISTO and SCATTER), as distinct from the detailed component-by-component status reports he now receives. As one result of this greater power in portfolio status analysis, it was expected that later applications of the Role Construct Test would reveal increases in complexity of discrimination among portfolios, as well as a shift toward more status-related dimensions of discrimination from the goal-related dimensions used before.

Given a richer perception of portfolio status, it was also expected that there would be a long run pressure by the PM on the client for more detailed and explicit definition of goals.

It was further expected that explicit and formal inter-portfolio comparison would occur in portfolio status evaluation (e.g., using the HISTO function with overlaid portfolios). To the extent that explicit standards for comparison (e.g., the Dow Jones Industrials) were available in the system, it was expected that they would be employed in status evaluation.

As a result of the above expected increase in intelligence activity and power, plus the increased focus on overall portfolio status expected, it was expected that PM stock transactions would become more need-stimulated as opposed to opportunity-stimulated than before. It should be noted, however, that if the PM used the system to focus even further on individual stock evaluation rather than portfolios, their problems might become even more opportunity-stimulated instead.

The hypotheses imbedded in this discussion of design assumptions and expectations about the Intelligence phase of MMDS with the prototype are summarized below. Some of these hypotheses are less operational than others, as will become obvious in the discussion of actual results in Chapter 6. It is held to be important for a designer to write them down and remember them nonetheless, even if the resulting control process is somewhat subjective in nature:

H-1. Portfolios on the system will be reviewed more frequently than before.

H-2. Portfolios will be reviewed employing a greater variety of analytic tools and representations than before.

H-3. Functions which provide an overall view of portfolio status will be utilized especially heavily.

H-4. The PM's complexity of discrimination among portfolios will increase with use of the prototype system.

H-5. The dimensions used by the PM in discrimination among portfolios will be relatively more status-related (as opposed to goal-related) than before.

H-6. The PM using the prototype will begin to attempt to define portfolio goals with more detail and formality than a PM not using the prototype.

H-7. Explicit comparison of a portfolio with other portfolios and with standards will begin to occur as a part of status evaluation.

H-8. PM buy-sell decisions will become relatively more need-stimulated (as opposed to opportunity-stimulated) than before.

Design

The FILTER mechanism in the prototype was designed with an aim to provide for formal, exhaustive stock search across the approved list, something the PM almost never did previously. Thus, there was an expectation that FILTER would be useful in the design phase of decision making, and that formal, exhaustive security search would occur. It was further expected that the PM would begin to screen alternative securities on criteria related directly to portfolio needs, rather than upon their own inherent attributes. (For example, a PM may note in the HISTO display that one portfolio has a distribution of estimated total return that is rather low on the scale; he may then employ FILTER both to examine the low total return holdings in an account for sell candidates and to search the approved list for high total return buy candidates.)

Further, it was expected that the PM will begin to design explicit alternative portfolios for consideration, rather than focusing entirely on choice among individual securities as before. It was also expected that alternative portfolios examined will be further from actual status (in terms of differences in holdings) than alternatives implicitly being considered before; i.e., search for "solution" portfolios was expected to be more global than before.

H-9. The PM will conduct formal, exhaustive searches of the available securities for stocks meeting specified conditions.

H-10. Stock searches will tend to become guided by specific portfolio needs rather than by individual stock qualities.

H-11. The PM will design explicit hypothetical portfolios for consideration as possible alternative portfolios.

H-12. The hypothetical portfolios that are designed will tend to be farther from current status than the rather local alternatives implicitly considered previously.

Choice

One of the expectations concerning solution search carries over to solution choice as well. There was a general expectation that, as the PM's perception of his available "portfolio space" improved through use of the system, his search and choice would range wider from current status than before. The hypothesis underlying this expectation is that the PM currently tends to make relatively local and incremental modifications to a portfolio at any one time because he has a very limited view of alternative portfolios restricted to a perception of alternative stocks, with relatively little explicit sense for the potential consequences of any decision for the portfolio in the aggregate. As in the Intelligence phase, in the Choice phase the PM will employ explicit mechanisms for comparison of alternative portfolios with each other and with available standards. It was also expected that dimensions examined and used in comparison and choice would relate more to aggregate portfolio characteristics than previously, when the focus was largely on individual security characteristics. In other words, facilities such as HISTO, SCATTER, and AGGREGATE should find considerable use in comparison prior to choice.

H-13. The PM will tend to choose alternative portfolios which represent a larger departure from current status than choices made previously.

H-14. The PM will employ explicit mechanisms for alternative portfolio comparison for Choice.

H-15. The variables examined during Choice will tend to be more related to aggregate portfolio characteristics than before.

Implementation and Control

The prototype system has been designed with no facility to aid directly in the process of implementation of a selected trade. Since there is no hard copy generation capability, in fact, the PM must write down on an order form any trade program which he has selected from the ARDS screen.

Similarly, there is no direct facility designed into the prototype to aid in control on trade execution, beyond the fact that the system could retrieve an old trade program for him. Expectations relating to higher levels of control, such as control on the decision process and models, are discussed under the sections on decision programming and learning.

Plans

The models and procedures currently used by the PM are almost exclusively informal and mental. In particular, the Role Construct Test suggested that the PM had highly simplified mental models for portfolio status relative to those for stock status. Through use of the prototype system, with its variety of aggregate portfolio status measures, it was expected that the PM's model for portfolio status would grow more complex.

In addition, the SCATTER facility was expected to provide the PM an ability to visualize correlations between various stock attributes, as one step toward developing more rigorous models of stock status and performance.

Note that it was an explicit design tactic here not to impose prestructured and complex models upon the PM. The author, following McKenney (1967), Amstutz (1968), Little (1970), and others, feels that the decision maker must in general understand fully and participate in the development of any model which he is expected to use effectively. The tactic chosen was to provide facilities that represent steps toward more complex models, with the notion that the system and its user will evolve in that direction over time. The prototype system, though without such complex models in its initial version, provides a general framework within which a PM eventually can interact with a security valuation, portfolio selection, or performance evaluation model where desirable.

H-16. From use of the SCATTER function, the PM will begin to develop and remember graphic relationships between security attributes.

<u>Memory</u>

It was expected that the PM would begin to make use of computer memory of stock lists and hypothetical portfolios only slowly, since CREATE is one of the more complex functions to control in the system.

It was not expected that he would view the prototype system as his "working memory" for portfolio status either, since it would only cover a small fraction of his portfolios and it would not always be immediately accessible.

> H-17. The PM only slowly will begin to use the computer memory for stock lists and hypothetical portfolios.
>
> H-18. The PM will not use the prototype as his "working memory" for portfolios, but will continue to use the old reports as well.

Operators

Most of the expectations concerning use of operations in the decision system have been outlined under the various phases in the decision process above. The expectations regarding learning and flexibility of use of the system operators will be discussed later.

Decision Structure

Previous researchers have observed that MMDS tends to make decision making more agile and flexible, with what were before rigidly separate decision phases blending together (e.g., Scott Morton, 1967). In this particular case of portfolio management, the expectation was that the greatly reduced "set-up cost" of conducting a portfolio review (largely the clerical operations necessary to bring the old report up-to-date) would lead to more review activity, with relatively easy switching of attention from one portfolio to another.

In fact, it was expected that cross-portfolio functions such as STOCK might actually encourage rapid transfers of attention from one portfolio to another. Otherwise, the current PM decision phases of design and choice are already quite blurred; in fact, more formality in those phases might be expected, given the new formal functions available in the prototype. In other words, rather than sequential examination and screening of individual stock trade possibilities in a satisficing manner, the PM may now design complex multiple-trade alternatives before entering the choice phase, where these complex alternatives will be evaluated.

H-19. The PM will tend to switch attention more readily from portfolio to portfolio in the process of review.

H-20. The PM will exhibit less satisficing behavior (considering each stock independently in sequence) and instead will design and choose among one or more complex multiple-trade alternatives.

<u>Mode of Interaction</u>

It was fully expected in the design of the prototype, that at least two modes or paces of interaction would be present:

(1) a staccato, several-second user response was expected to characterize the dialogue necessary to specify a given function and all of its parameters; it was expected that this mode would be broken by

(2) the occasional minute-or-more response cycle, which would characterize points at which the user was studying in detail the final output of a particular complex function and deciding what to do next.

This pattern of user response was felt to be quite predictable given the design of the man-machine conversational graph. Note that this response distribution is different from the poisson distribution generally assumed in most models of time-shared computer systems.

Control over interactive mode, however, was not completely in the hands of the user, for response time delays on the part of the system could be a function of other load on the time-sharing system. Hence, some user dissatisfaction with lack of control over interactive mode or pace could be expected.

H-21. The interactive pace will exhibit two primary modes: a short (about five seconds) cycle, and a longer (about one minute) cycle.

H-22. The PM's dissatisfaction with his lack of complete control over pace will increase as system response time degrades.

Structure of Interaction

The prototype system was designed with enough prompting and tutorial devices that it was expected that the so-called "direct" interactive structure would be attained rapidly. The direct structure, discussed in Chapter 3, involves decision maker and machine interacting directly and closely, with only occasional interactions with the interpreter-designer, usually to recommend some design change or improvement.

H-23. The PM will rapidly attain the "direct" structure of interaction with the prototype.

Language and Form of Interaction

The graphic forms of display were incorporated into the design as an attractive and concise way to represent overall portfolio structure to the PM. It was expected that he would make substantial use of these displays as totally unique and revealing ways of looking at his portfolios.

The keyboard was elected to be the exclusive input medium. Although light pen or "mouse" input might have been adequate for menu-selection conversation, they would have been unsatisfactory for general direct command input. Since the keyboard seemed essential, therefore, it was decided to use it exclusively for the sake of consistency. The expectation was that the PM would find keyboard input quite acceptable. This expectation was based primarily upon the designer's bias that the keyboard provides the most positive as well as general input medium; it clearly runs counter to popular wisdom which says that a manager will not use a keyboard because of its association with clerical roles.

H-24. The PM's will find the graphic displays attractive and make heavy use of them.

H-25. Keyboard input will be quite acceptable to the PM.

Flexibility of Interaction

The three forms of conversation described earlier (menu-selection, stacked-selection, and direct command) were provided to allow for flexibility in expression on the part of the user. The design allows for the user to employ any mix of the forms within one conversation, and the forms may be mixed within one command line (e.g., the command "DIRECTORY 1 3" takes the user directly to the DIRECTORY function and then responds to the next two menus with 1 and 3, thus selecting a full directory to be displayed in order of size). It was expected that users would use this flexibility in form of expression.

H-26. The PM will make full use of the flexibility provided in form of expression.

Learning and Training

It was expected that the PM would have relatively little difficulty in adapting to the mechanics of the prototype system. The basic tutorial, or prompting mode of the system was aimed at keeping the mechanics simple. Basically, the only options that a user need know about that are not always displayed for him are (1) that typing "TOP" returns him to the top node in the conversational graph from wherever he is, and (2) that he must occasionally hit an "erase" key when the screen is full to go on to the next page of output.

It was also expected that the PM would learn most quickly to use those operators which are easiest to control and closest to his normal information system output (e.g., STATUS and STOCK). It was expected that he would, however, begin to explore other, more complex facilities immediately and learn to use many quite readily.

Similarly, it was expected that the early user will begin with the menu-selection form of interaction and will only learn to use more concise forms with experience. In addition, the PM was expected to use the direct or stacked forms of interaction primarily for those functions which he uses the most or which are very simple in specification; he will rely largely on menu-selection for unfamiliar or very complex functions.

The PM was expected to pass through phases analogous to the enactive, iconic, and symbolic stages of cognitive growth in learning to use the prototype system. The enactive stage will be mastery of the mechanics of system use; as noted above, this is expected quite quickly.

The iconic stage involves the internalization of a simple representation of each individual function (e.g., the PM learns to associate HISTO with a graphic distribution display, STOCK with a report of holdings across portfolios, etc.). At the iconic stage, the PM has a rather rigid though complete mental picture of the individual functions in the system as independent pieces in a mosaic from which he may choose.

The symbolic stage comes when the PM begins dynamically to link individual functions into logical sequences or procedures for the purposes of decision-making. The PM can now manipulate and link the individual functions mentally in planning a full decision procedure. Over time he will build up a repetoire of a number of such procedures or plans. Eventually, these plans themselves may become programmed and transferred to the machine.

It was this designer's expectation that the enactive stage would be reached almost immediately, within an hour's session, and that the iconic stage would be reached shortly thereafter, within the first several hours of use. On the other hand, it was expected that the symbolic stage, the full integration of system functions into the PM's decision process, would take substantially longer, perhaps several weeks of continual usage (McKenney, 1968). In other words, it was expected that the step would be large from (1) seeing the histogram as just an interesting representation of a portfolio to (2) seeing it as displaying a structural aberration in a portfolio, which one would expect to follow with a TABLE display to pinpoint the cause of the aberration, and then a STOCK run to see what other portfolios hold the problem stock or stocks, etc. It is at the symbolic stage where the interactive system begins to assume full decision power. Thus the expectation is that there may be a significant transient period of learning before a complex MMDS is fully functional at the symbolic stage.

The basis for many of the expectations expressed here is the heuristic search model of decision-making developed and elaborated by Simon, March, Cyert, and others. The basic notion is that a human decision maker will search heuristically for ways to allocate his limited decision making resources and capabilities so as to achieve satisfactory performance. By introducing a MMDS for portfolio management decision making, the apparent economics of this decision resource allocation for the PM has been shifted greatly -- avenues that previously yielded very low return relative to decision effort invested are now apparently capable of much greater relative return.* Thus these expectations reflect the idea that the PM will heuristically explore new capabilities and gradually shift the allocation of his decision energy and resources to accommodate the new economics of decision making.

---

*The word "apparently" is used here to indicate our focus only upon the return on incremental decision effort by the decision maker. This view neglects the fact that the process is now more capital-intensive through introduction of the computer. However, as far as the individual decision maker is concerned, his short-run decision behavior may be relatively unaffected by these long run economic considerations.

H-27. The PM will easily and quickly maintain the enactive stage, or mastery of the mechanics of the prototype.

H-28. The PM will also readily attain the iconic stage, the understanding of the prototype as a mosaic of independent functions.

H-29. The PM will only very slowly attain the symbolic stage, the integration of planned sequences of system functions into his decision process.

H-30. The PM will learn and use the familiar functions most readily.

Decision Programming

It was expected that the extensive use of the prototype system would make steps in the decision process more formal and would result in making the decision process as a whole more visible to the PM. It was felt that this would lead to an awareness of programmable aspects of the process and thus generate another pressure for a macro-building facility, which would allow the PM to begin steps toward decision programming. Since the prototype was designed without such a macro-building facility, it was expected that this decision programming pressure would manifest itself in terms of PM suggestions to this designer as to further decision operators, operands, or plans which might be programmed.

H-31. Extensive use of the system will lead the PM to a more explicit and structured model of his decision process, which will lead in turn to suggestions for programming further phases or functions in the process.

Other Expectations

There are a myriad of design assumptions imbedded in the prototype design process beyond these listed above. However, as an initial working list for the purpose of control, these were judged by this investigator to encompass the most substantial expectations.

5.11 Summary

This chapter has described the application of the MMDS design methodology of Chapter 4 to the development of a prototype MMDS for portfolio management. The research tactic of an in-depth field study was selected over the alternative of a laboratory experiment, since it was felt that the current limited state of the field was served better by a comprehensive study aimed at developing and refining of new hypotheses rather than the focused testing of a few given hypotheses.

The context of the design exercise was the pension fund section of a major bank. The eventual users of the prototype system were portfolio managers who typically manage 25-75 portfolios each, with sizes of $2-350 million each. The focus of the design was upon support of the portfolio revision decision for the common stock portion of the portfolio.

Both a normative and descriptive model of the portfolio management decision process were developed from March, 1968, to January, 1969, using the general decision model framework of

Chapter 3 as a guide. The normative model was a statement of desired characteristics of the decision process, whereas the descriptive model was a detailed outline of the process developed from interviews, questionnaires, reports, decision protocols, and psychological tests.

The problems defined by a comparison of normative and descriptive models lead to a simple functional model of the potential MMDS, consisting of functions defined to alleviate each specific problem. This set of "ideal" functions was reduced to a more workable design specification in the constraining phase, when the real limitations of time, technology, organization, and funds were first seriously introduced.

The functional design specifications were then elaborated into a conversational graph specification, which allows for early mental stimulation by the designer of the mechanics of the expected man-machine dialogue. This graph representation was carried through to implementation, and the prototype system was made available for experimental use in July, 1969.

In the later phases of design, a MMDS control system was designed. However, only part of this control system, consisting primarily of a programmed decision monitor system, was actually implemented due to unanticipated difficulties in the experimental situation.

The chapter concludes with a description of the most salient assumptions and expectations about eventual system behavior that developed through the process of design. These expectations are summarized in the form of concise hypotheses, which are compared with actual usage experience in the next chapter.

CHAPTER 6

RESULTS OF PROTOTYPE SYSTEM DESIGN

6.1 Introduction

This chapter describes the results of experimental use of the prototype MMDS for portfolio management whose design was described in the previous chapter. The results and observations are compared with the expectations of the designer outlined in Section 5.10, and implications for the theory of MMDS are drawn. Finally, observations on this application of the design methodology proposed in Chapter 4 are summarized. Before the results are described, however, certain qualifications on the experimental situation must be noted.

6.2 Qualifications on the Results

In interpreting the results of prototype system use, the reader should be aware of several qualifications. First, the prototype was in use for only a relatively short period, from July, 1969, to February, 1970. During that period, no PM logged more than seven hours cumulative at the terminal. The total of cumulative time logged for all eleven investment people who used the system was only 29 hours. Thus, most, if not all of the phenomena observed must be considered transient. There was no indication that any of the PM's had achieved a steady-state pattern of use of the system.

Also, there is a possibility that the fact of their being observed affected the behavior of the PM's. This possibility exists despite the fact that the author was actually present during only a small proportion of PM system usage, and he provided relatively little active direction, serving primarily to observe the process and to answer questions. Nonetheless, the PM's were well aware that the project was an experiment, and that their usage of the system was a subject of interest.

In addition, six of the eleven Trust users actually had none of their own portfolios on the system. Thus their use of the system was more an introduction to and exercise of its functional capabilities than a simulated "live" decision situation.

Also, access to the system was limited, as noted in Section 5.9. Thus a PM generally had to plan ahead to schedule a time either very early or very late in the day to use the system. Consequently the prototype was not used as a readily accessible and responsive device, as an operational system should be.

As a result of this limited prototype use and of the loss of subjects and controls mentioned earlier, many of the measures and instruments planned in the original experimental design were abandoned. Thus for some hypotheses, no direct evidence was collected, and the discussion of this chapter simply describes the measures planned in the original experimental design. (This applies to hypotheses 4, 5, 6, 10, 13, 14 and 15.) For thirteen of the remaining 24 hypotheses, direct observational evidence is available; whereas for eleven hypotheses both monitor trace and observational evidence is cited.

It should be noted once again that the trust institution under study is implementing an operational extension of this prototype MMDS. Thus the planned measures for the original experimental design still have considerable value in that they will be applied in the process of control of operational MMDS behavior with an even broader base of over 20 PM-users. In other words, some of the hypotheses merely proposed here will receive thorough testing in this anticipated next phase.

6.3 General Results

During the period that the prototype system was available, it was used by eleven different investment professionals for cumulative periods ranging from 1/2 hours to seven hours each, for a cumulative total of 29 console hours. These professionals used the system for from one to eight sessions each, for a cumulative total of 31 console sessions.

The system was also used by four other individuals inexperienced in professional investment management for cumulative periods of one to 17 hours each, in a total of 18 console sessions. All four individuals were computer systems professionals.

For the purposes of this experiment there are two primary differences of importance between these two groups, the investment professionals and the non-investment users. One, the non-investment users could be classed as naive investors, since all of them had an expressed interest in the stock market but none of them had a personal portfolio of any complexity nor any significant investment experience. Thus differences in decision function usage by the two groups could be considered representative of differences in decision style and needs between sophisticated and naive (though not totally ignorant) investors.

The second major difference is that the investment users could be considered naive in the use of computers, whereas the others were all computer professionals with from five to eight years of training and experience in computer system usage and development. Thus any greater facility with prototype system usage and control would suggest the value of prior computer training in MMDS usage by the latter group.

The monitor system in the prototype maintained a complete "trace" of the man-machine dialogue conducted by each of the above users. These trace data are used to support some of the observations on hypotheses in the following sections. A partial sample of this trace is shown in Figure 6.1. The three columns on the left in the figure are the time of day of the adjacent response in hours, minutes, and seconds. The fourth column indicates the user input, and the messages on the right are the general prototype queries. Note that the trace does not record the actual details of any given display produced, but

TIME	USER RESPONSE	SYSTEM RESPONSE
18 13 16		SELECT ONE OF THE FOLLOWING BY TYPING ITS NUMBER OR THE NAME WITHIN QUOTATION MARKS... 1. PRINT 'DIRECTORY' OF LISTS 2. 'CREATE' A NEW LIST 3. 'DELETE' A LIST 4. 'RENAME' A LIST 5. REVIEW LIST 'STATUS' 6. 'FILTER' A LIST 7. ORDERED 'TABLE' DISPLAY 8. FORM 'HISTO' GRAM 9. FORM 'SCATTER' PLOT 10. 'STOCK' RUN 11. 'ACCOUNT' MARKET VALUE
18 13 24	2	
18 13 36		PLEASE SPECIFY THE TYPE OF LIST YOU WISH TO CREATE BY SELECTING ONE OF THE FOLLOWING OR TYPE '4' FOR DEFINITIONS... 1. HYPOTHETICAL PORTFOLIO 2. STOCK LIST 3. PORTFOLIO CHANGE PROGRAM 4. DEFINITION OF TERMS
18 13 53	2	
18 14 47		SELECT THE LIST BUILDING OPERATION YOU WISH TO APPLY... 1. TYPE IN TICKER SYMBOLS 2. FILTER AN EXISTING LIST OR PORTFOLIO OR THE CSL 3. SELECT COMPONENTS FROM AN EXISTING LIST OR PORTFOLIO
18 14 54	2	
18 14 55		TYPE THE SHORT FORM NAME OF THE EXISTING LIST YOU WISH TO FILTER OR 'CSL' FOR THE COMMON STOCK LIST. TYPE 'CIR' TO GET A DIRECTORY LISTING...
18 16 51	CSL	

FIGURE 6.1: Sample of Prototype Man-Machine Monitor Trace

only reproduces the dialogue which led to display specification. The decision not to save specific displays was made in order to conserve disk storage space and computer time.

The other evidence brought to bear on the hypotheses is derived from direct observation of prototype use by the author, plus after-the-fact discussion with users of the system.

Perhaps the most important and general observation is that virtually all of the experimental users of the prototype, both PM's and others, liked the system and saw its potential value in portfolio management as great. Each user qualified that reaction, of course, in a variety of ways that are discussed further below. In particular, they wanted a system like the prototype, but only if it were easily accessible upon demand during the day and if it provided reasonable response times (i.e., averaging 5 seconds or less). But their net reaction was very positive, to the extent that the experimental institution has initiated development of a full-scale, operational MMDS for portfolio management. The system will support over twenty portfolio managers in all areas of trust investment management.

More specific observations on the results are outlined in the following sections, organized under the same headings as the design assumptions and expectations of Section 5.10. Within each heading, the previously stated hypotheses will be taken one by one, each being followed by a description of actual results where relevant. The chapter will conclude with several additional observations of interest not anticipated in the hypotheses.

6.4 Intelligence

H-1. Portfolios on the system will be reviewed more frequently than before.

There were no direct measures used to test this hypothesis. In any case, since the prototype was never used as an operational tool, it is unlikely that it had an effect on normal PM review behavior. In the original experimental design, this hypothesis would have been tested using questionnaire data, which monitored both the frequency and duration of portfolio reviews, both for the three PM's scheduled to use the prototype operationally and also for the four PM controls not using the prototype. Since two portfolios managed by subject PM's yet not on the prototype were also to be monitored, any transference of increased review activity to these manually analyzed accounts would have been tracked as well.

Despite the lack of direct evidence, however, there are surrogate indicators relevant to the hypothesis. For example, the trace data did indicate a high degree of portfolio review activity on the part of the investment group, both absolutely and relative to the non-investment users. The prototype user could apply a function to a stock, a stock list, the fully approved list of stocks and a portfolio. Table 6.1 indicates that the investment professionals using the system focused a majority of their attention upon portfolios. The next most important item of attention in their use was the single

Operand Type	Professional Investment Users: Number of References	Professional Investment Users: %	Non-Investment Users: Number of References	Non-Investment Users: %
Portfolio	211	69.	68	34.
Stock	68	22.	25	13.
Stock List	8	3.	93	47.
Approved List	18	6.	13	6.
TOTAL	305	100.	199	100.

TABLE 6.1

Number of References to Operand
Types by Prototype Users

stock, analyzed by calling for holdings across portfolios; therefore, this too is a portfolio review-related action. The non-investment users focused a substantially smaller amount of attention on these two items and concentrated much more upon simple stock lists; this is not surprising given their lack of familiarity with the portfolios and their potential interest in stocks for their own personal portfolios. In addition, Table 6.2 indicates that the investment group devoted very little of their time to functions such as CREATE and DELETE, which are aimed at manipulating stock lists, not portfolios.

These differences in function and data type usage between the investment and non-investment users are significant. The test for homogeneity of the two samples in Table 6.1, for example, shows the null hypothesis rejected strongly (Chi square = 149.9, DF = 3, 2-tail significance < .0001). Similarly, the test applied to the two function usage frequency patterns of Table 6.2 also rejects the null hypothesis strongly (Chi square = 41.166, DF = 10, 2-tail significance < .0001).

H-2. Portfolios will be reviewed employing a greater variety of tools and representations than before.

The portfolios on the prototype system were reviewed by the investment group using the full spectrum of analytic options. They in fact used these new portfolio-oriented analytic functions (i.e., TABLE, HISTO, SCATTER, and STOCK) 66% of the time versus only 6% for

Prototype Functions	Professional Investment Users: Number of Applications	%	Non-Investment Users: Number of Applications	%
DIRECTORY	45	11.	28	11.
CREATE	5	1.	20	8.
DELETE	3	1.	9	3.
RENAME	0	0.	3	1.
STATUS	22	6.	22	8.
FILTER	48	12.	29	11.
TABLE	24	6.	22	8.
HISTO	82	21.	54	20.
SCATTER	83	21.	49	18.
STOCK	69	18.	25	10.
ACCOUNT	12	3.	5	2.
TOTAL	393	100.	266	100.

TABLE 6.2

Frequency of Application
of Prototype Functions by Users

the familiar portfolio review function (i.e., STATUS). This result suggests that the new analytic tools and representations received a significant amount of attention, substantially more than the one report form already provided in their current information system (even if STATUS did provide the added attraction of current prices and holdings). Thus the prototype users did broaden their view of portfolio status substantially. Whether such behavior would stabilize over the long run or whether it represents only temporary exploration, however, has not been shown by this limited experience.

> H-3. Functions which provide an overall view of portfolio status will be utilized especially heavily.

The functions which tend to summarize portfolio status in a concise aggregate view are HISTO, SCATTER, and ACCOUNT. As indicated in Table 6.2, these functions were used 45% of the time by investment group and 40% of the time by the non-investment group. In fact, these aggregate view functions seem considerably more popular with the investment group than STATUS and TABLE, both of which produce very detailed, asset-by-asset reports. These latter, detailed analyses were only used 12% of the time by the investment group versus 45% for the aggregate view functions. These results do indeed suggest that a majority of user attention has been focused on gaining an aggregate perception of status. This is in sharp contrast to their use of their original information system, which provides only the standard, detailed portfolio review report as an indicator of portfolio status.

> H-4. The PM's complexity of discrimination among portfolios will increase with use of the prototype system.
>
> H-5. The dimensions used by the PM in discrimination among portfolios will be relatively more status-related than before.
>
> H-6. The PM using the prototype will begin to attempt to define portfolio goals with more detail and formality than a PM not using the prototype.

Given the very limited and non-operational use of the prototype system which actually resulted, no substantial change along the lines of these three hypotheses was expected. Thus, no attempt was made to collect any evidence bearing directly on them as originally planned.

In the original experimental design, the intention was to test H-4 and H-5 by later applications of the Role Construct Test for both subject and control PM's, aimed at eliciting the dimensions and complexity of their discrimination among stocks and portfolios. The reader will recall that the initial application of the Role Construct Test showed (1) that the majority of PM's showed more complexity in their discrimination among stocks than they did among portfolios (i.e., 5 out of 6 tested), and (2) that the dimensions used to discriminate among portfolios were far more goal-related than status-related. This designer's early hypothesis was that the PM's lack of adequate perception of portfolio status, especially in the aggregate, and a consequent focus of attention, by default, on stock status and portfolio goals would lead to these test outcomes. The new access to aggregate portfolio status measures in the prototype was expected to change these results.

Hypothesis H-6, on the other hand, was going to be tested by interview only. It was expected that, with a clearer and more detailed view of portfolio status, the PM would begin to define his portfolio goals (and sub-goals or criteria), more in the operational terms of the status measures available than in the relatively non-operational terms he uses now (e.g., high total return, high risk, etc.). However, all of these hypotheses H-4 through H-6 must now await use of an operational MMDS before testing.

> H-7. Explicit comparison of a portfolio with other portfolios and with standards will begin to occur as a part of status evaluation.

Although no explicit standards were available in the prototype data base, there was substantial evidence of interportfolio comparison observed. This evidence lies in the relatively high usage of prototype functions which allow for explicit portfolio-to-portfolio comparison, as well as in the rapidity of attention shift from portfolio to portfolio noted in the later discussion of hypothesis H-19. Although the only formal comparison mechanism was the HISTO function, which allows overlay of two portfolio distributions, the investment users also compared accounts at the detailed holding level by use of the STOCK function.

The HISTO function was used by the investment group 21% of the time, and the STOCK function was used 18% of the time (see Table 6.2). This is in sharp contrast to pre-MMDS behavior where the PM's were never observed making an explicit, formal comparison between two portfolios. (Although there was some informal comparison of the sort, "This account has not performed as well as portfolio Y.")

Several of the users were even observed attempting to compare portfolios with the SCATTER function, by rapid paging from one portfolio to the next, maintaining the same two dimensions of the scatter plot as they went. Both direct observation and the monitor traces clearly showed this effort to compare portfolios by any means available. There is an implication here of a desire for even more formal comparison mechanisms in the system.

Incidentally, as Table 6.2 indicates, the STOCK function became one of the most popular with the investment user group. The primary reasons for its high usage would seem to be (1) its usefulness in comparing investment timing and consistency in the same stock across many portfolios, and (2) its very simple mechanics of use (e.g., one need only type "STOCK IBM" to get a report on all holdings of IBM).

H-8. PM buy-sell decisions will become relatively more need-stimulated (as opposed to opportunity-stimulated) than before.

Again, since no substantial change was expected along the lines of this hypothesis due to the limited use of the prototype, only observational evidence was collected. The original design plan was to monitor the proportion of need-stimulated versus opportunity-stimulated decisions for both subject and control PM's through the course of prototype usage, using the questionnaire that they filled out on each portfolio review or revision. The expectation was that need-stimulated trading would increase as a function of the expected increase in portfolio status evaluation noted above. Of course, it could have turned out that the users would have applied the prototype primarily as a stock analysis system instead (e.g., making high use of stock search and FILTER capabilities), and then opportunity-stimulated trading would have been expected to increase. This was not the case, in fact, as noted earlier.

Even though there was no direct questionnaire evidence on this hypothesis, however, direct observation of PM users of the prototype did indicate that they discovered "problems" in portfolios as a result of use of the system. Some of these problems might not have been discovered otherwise. For example, using successive scatter plots, one PM discovered a major structural difference between two portfolios that he felt had similar objectives. He hadn't seen the difference before in part because his normal status reports for these two portfolios had entirely different formats for historical reasons. Once he saw the structural discrepancy he took trading steps to eliminate the difference.

Similarly, a PM was observed discovering inconsistencies in purchases, using the STOCK function to examine holdings across portfolios that he had wanted to manage in a similar way. He made the necessary trades in those portfolios which had not purchased a stock he had intended for them.

These observations tend to support the idea that the PM, using such an MMDS, will discover problems and needs in portfolios that he might not otherwise recognize. The implication is that his decisions may indeed become more need-stimulated than before.

6.5 Design

> H-9. The PM will conduct formal, exhaustive searches for stocks meeting specified conditions.

The measure used to test this hypothesis is the frequency of use of the FILTER function, which allows for exhaustive search of the full approved list of stocks. Table 6.2 indicates moderate usage (i.e., 12%) of FILTER in searching for stocks that meet specified criteria. Table 6.1, however, indicates that the full approved list was used as an operand relatively less often than FILTER was used as an operator (e.g., only 6% of the time), indicating that many searches were not exhaustive but were applied to a specified portfolio as a step in portfolio analysis, not stock search. Thus, although the user did apply the stock search function across his full security universe, this activity was slight relative to his portfolio analysis activity.

One possible explanation for this low usage is that FILTER may not be as attractive a function as originally supposed in the early stages of design. Certainly, when this investigator actually used the FILTER function in the final debugging stages before system delivery, his own reaction to the function was somewhat negative. Upon introspection, it was concluded that the problems seemed to be the following:

(1) Filtering on multiple criteria simultaneously could easily prove overly constraining, with the disappointing result that no stocks would pass the filter;

(2) The sharp cutoff of a specified filter criterion left this investigator with an uncomfortable uncertainty in not knowing what stocks had barely missed passing the explicit limit.

After such early use of FILTER, this investigator developed an expectation that, in fact, functions such as SCATTER or TABLE would prove more attractive to the PM than FILTER for stock search, because they displayed a complete ordered array of the stocks in a specified list, thus allowing for a very flexible subjective filtering of the stock universe (i.e., TABLE displayed stocks ordered on one dimension; SCATTER displayed them ordered on two dimensions). Results relevant to this latter expectation about TABLE and SCATTER usage are discussed in Section 6.17.

> H-10. Stock searches will tend to become guided by specific portfolio needs rather than by individual stock qualities.

There was no clear indication that any user progressed to the point of consistently linking formal portfolio analysis and problem-finding with formal stock search. This is not entirely surprising given the limited prototype usage experience, and the notion in hypothesis H-30 that such effective, integrated use of system functions takes considerable time to learn.

In the original experimental design the intention was to gather data relevant to this hypothesis via the user trace. In particular, a pattern such as a HISTO applied to a portfolio, followed by a FILTER of the whole stock universe on the same data variable used in the HISTO would be taken to indicate that the stock search had been guided by a portfolio need discovered in HISTO.

H-11. The PM will design explicit hypothetical portfolios for consideration as possible alternative portfolios.

H-12. The hypothetical portfolios that are designed will tend to be farther from current status than the rather local alternatives implicitly considered previously.

Since the facility to create hypothetical portfolios was not implemented (CREATE only builds stock lists) due to constraints in time and resources, there was no direct opportunity to collect evidence relative to these hypotheses. Table 6.2 does show clearly that a CREATE function that does _not_ build hypothetical portfolios was not highly favored by the investment group (not that one which did would necessarily receive any more attention). This is in contrast to the non-investment users who show substantially higher use of CREATE. During console sessions, there was, however, some observed discussion by investment users of hypothetical alternative portfolio graphical _shapes_ in the context of the graphic HISTO and SCATTER functions, even if such hypothetical forms could not be constructed formally on the prototype. This would seem to indicate a propensity to design abstract representations of alternative portfolios, in response to the new aggregate status displays, that might lead to usage of a formal design facility.

In the original experimental design, data relevant to H-11 were to have been derived from the monitor trace showing frequency of hypothetical portfolio creation and use. Data relevant to H-12 were to have been obtained from the same source by analyzing specific hypothetical portfolios created by users to see how great a departure they represented from actual status (in terms of number of holdings changed, market value of necessary trades to accomplish change, etc.).

6.6 Choice

H-13. The PM will tend to choose alternative portfolios which represent a larger departure from current status than choices made previously.

H-14. The PM will employ explicit mechanisms for alternative portfolio comparison for choice.

H-15. The variables examined during choice will tend to be more related to aggregate portfolio characteristics than before.

There is no direct evidence for any of these hypotheses beyond that discussed in the previous two sections concerning comparison and aggregate measures. It seems clear, in retrospect, that the prototype system without a facility for building hypothetical alternative portfolios provides no real formal aids for the choice phase and very little help for solution design. Thus the prototype system supports primarily the Intelligence phase of decision making with formal operators. This was not the original design plan, of course, but resulted from constraints of time and design resources.

In the original design plan, data relevant to H-13 were to be obtained from trust accounting records showing the absolute and relative volume of stock trading for a portfolio both by period and by single trading action. These accounting data were to have been analyzed for both subject and control PM portfolios, to see if portfolios on the prototype were experiencing more turnover and more dramatic changes. Data relevant to H-14 and H-15 were to have been gathered from monitor traces, showing the hypothetical comparison activity and the level of such comparison, whether aggregate or detailed.

6.7 Plans

> H-16. From use of the SCATTER function, the PM will begin to develop and remember graphic relationships between security attributes.

Again, there was only observational evidence collected to support this hypothesis, given the limited usage of the prototype. Several investment users, however, were directly observed by the author checking graphically such rough hypotheses as the following:

(1) Current price-earnings ratio should correlate with estimated earnings growth rate,

(2) Current price-earnings ratio should correlate with estimated future total return; etc.

Such activity could be viewed as a beginning step toward building and validating more formal models on the part of the decision maker.

One PM was observed to indulge in still another form of rough hypothesis evaluation. He had three portfolios placed on the system. One had performed very well historically, another moderately well, and the third rather poorly. He expected that the three would show substantial structural differences. In fact, by sequencing through successive SCATTER plots, he noted a consistent difference in distribution of holdings between the three portfolios. In a plot of price-earnings versus estimated earnings growth rate, the poor-performing portfolio was composed of relatively high P/E-low growth stocks compared to the top-performing portfolio, with the third account roughly in the middle. Again, this sort of activity represents a struggle to model more rigorously the relationship between portfolio structure and performance. However, this is another hypothesis that requires long term MMDS usage for reasonable evaluation, and it is one that is difficult to test with more than interview or observational evidence.

6.8 Memory

H-17. The PM will begin to use the computer memory for stock lists and hypothetical portfolios only slowly and after some experience.

The investment users did make relatively little use of the CREATE and FILTER facilities for storing away in computer memory particular stock lists. Table 6.2 indicates that they only used

CREATE and FILTER roughly 13% of the time, versus 19% for the non-investment users. In fact, the fraction of the time that the investment user actually asked for the STOCK list first CREATEd or FILTERed to be stored was considerably less than 13%. The user is always asked specifically by the system if he wants to store a list away with a name, and the traces indicate that the users generally responded "no".

It is expected that this relatively low use of FILTER or CREATE was only partly because they are relatively complicated functions to specify (i.e., they require a dialogue of several lines). Rather, it is largely because hypothetical portfolios could not be created, but only stock lists.

H-18. The PM will not use the prototype as his "working memory" for portfolios, but will continue to use the old reports as well.

The prototype, of course, never had the operational character to begin to test this hypothesis, and the PM's naturally continued to use their old reports. If the prototype had been more operational, it was still expected to be sufficiently inaccessible (i.e., the PM had to leave his desk and go to another room) to the PM that he would only use it for substantial portfolio analysis and review sessions, and that he would rely on his old review reports for quick answers to ad hoc questions. Data on this hypothesis would have been collected by interview with the PM's to get their perception of their relative usage of the prototype versus the old reports.

This last hypothesis does suggest the larger question as to the necessary characteristics of a MMDS for the machine to be used as active working memory by the decision maker. It is to be expected that extremely high standards of accessibility, responsiveness, accuracy, and reliability must be met before the decision maker will part with his hard copy back-up report.

6.9 Decision Structure

H-19. The PM will tend to switch attention more readily from portfolio to portfolio in the process of review.

Observations and the monitor traces indicate a clear tendency for investment users to switch readily from portfolio to portfolio and back again in the course of analysis with the prototype. In particular, the number of portfolio switches per session was abstracted from the traces. A single portfolio switch is counted every time the user applies an operation to a new portfolio, or one that is different from the portfolio last referenced. In other words, if a user only analyzed one portfolio, even if he accessed it repeatedly using various displays, he would only be counted for one portfolio switch. The investment group alone made 131 such explicit portfolio switches in the 1,716 minutes they logged on the prototype. This means that they referenced a different portfolio every 13.1 minutes on the average. When investment users who logged more than three hours each are compared with those who logged less than three hours, the more experienced group is found to have made a portfolio switch every 11.4 minutes on the average, as compared to every 19.9 minutes for the less experienced group. This last result may indicate an increase in flexibility, as measured by portfolio switches, with user experience, although there is some danger that these data also just indicate a general increase in usage pace with experience. To check this last possibility, the number of portfolio switches were normalized by the total number of user responses and commands issued per session. The results show one portfolio switch for every 9.7 user inputs by the more experienced investment users, whereas they show one switch per 15.6 user inputs for the less experienced users. Thus the difference is still substantial.

This behavior is significantly different from the normal portfolio review process which typically involves focused attention on one portfolio until a complete program of trades is generated. The procedure with the prototype quite often involved specifying a particular report or graph format, then applying this format to several portfolios in succession, then modifying the display specification, applying it to more portfolios, etc. This is a very different and a more comparative approach to portfolio status review than the PM's usual procedure. Whether such a different structure of decision behavior would stabilize over long-term use of an operational MMDS remains to be seen. Nonetheless, it represents an approach to portfolio analysis that is practically infeasible in all but the most rudimentary fashion given the traditional information system supports of the PM.

> H-20. The PM will exhibit less satisficing behavior (considering each stock independently in sequence) and instead will design and choose among one or more complex multiple-trade alternatives.

Again, without a hypothetical portfolio creation facility in the prototype, the "complex alternatives" cannot be generated formally, and this hypothesis cannot be tested. In the original experimental design, data would have been brought to bear on this hypothesis from the monitor trace and further decision protocol collection. In other words, the trace would have been examined for more hypothetical portfolio design activity than individual stock screening and analysis; and the number of hypothetical portfolios created and compared would have been monitored. Decision protocols would have helped to show whether trading decisions still appeared to be simply an accumulation of relatively independent individual security buy-sell decisions, or whether the PM saw himself as choosing among alternative portfolios, where the impact of a potential set of trades was considered explicitly in terms of its aggregate effect on overall portfolio status.

6.10 Mode of Interaction

> H-21. The interactive pace will exhibit two primary modes: a short (about five seconds) cycle, and a longer (about one minutes) cycle.

The monitor trace maintains the clock time to the second of every complete action taken by system and user. However, since the original experimental design was dropped, no formal mechanism for analysis of this mode information was constructed. Under the original plan, however, the trace of action times would have been maintained in machine-readable form, from which it could easily have been converted first to a list of inter-arrival times of user actions and finally to a frequency distribution of such times for each user session. It was expected that these distributions would have shown a bimodal form consistent with the bimodal response time distributions discovered by Fox (1970) in his recent study of managerial use of interactive terminals.

These distributions, plus associated statistics, would also have been useful to get a partial indication from the trace of degradation in machine response times, to be correlated with user behavior in different response time environments. Of course, the number of users on the central time-sharing system during each session would have been another indicator of load and hence poor system response.

In retrospect, the response time frequency collection system might well have been more complex. It could have collected data for two distributions: (1) the distribution of times between user steps and when the system starts to display a reaction; (2) the distribution of times between the system's beginning a response and the user's next action. The first would be a more direct function of machine response time degradation; the second would be a more direct function of user reaction times. (Although there would still be some confounding due to the fact that different system displays take varying amounts of time to write on the screen, and yet all would be counted as user response time.) To the MMDS designer, an analysis of human response times within each function may provide some indication of function complexity; i.e., an unusually slow average response for a function might indicate an unnecessarily confusing design.

Note that if user response times do tend to follow a bimodal distribution, as suggested above, this has implications for machine-centered modeling and design optimization of time-shared or conversational systems. In other words, the general assumption of a Poisson distribution for user response time adopted by most of these models (Schrage, 1969) may be invalid.

> H-22. The PM's dissatisfaction with his lack of complete control over pace will increase as system response time degrades.

The evidence gathered relevant to this hypothesis was purely observational. As noted above, however, the monitor trace could have provided relatively unambiguous data on machine response time degradation, as well as any changes in system usage patterns that correlate with such degradation. Under the circumstances of limited prototype use, however, it was decided not to invest in such a detailed analysis.

On the other hand, the very heavy build-up of load on the central time-sharing system during the major portions of the day did afford the opportunity to observe directly the effects of slow response time on user satisfaction. In the several early morning sessions observed by this investigator, the system response time appeared to be typically in the range of 1-10 seconds, and users expressed no particular dissatisfaction with the pace. Later in the morning, however, as load on the central system at M.I.T. increased, growing user frustration was observed. The users seemed to become uncertain and often irritated at delays much beyond 10 seconds, wondering whether or not the system had received the last input. At some point in response degradation, when the response time began to exceed approximately 10-15 seconds for a significant number of system actions, the user would typically lose all interest in the system and would begin to do other things (e.g., conversing with this investigator, reading the paper, etc.). The PM did not feel at all comfortable staring at a static display screen for more than a few seconds waiting for something to happen.

It was also noted, however, that reduction in uncertainty about the potential length of the delay did appear to relieve some frustration. For example, the PM soon learned that HISTO and SCATTER were complex functions that, given a heavy load, could involve significant delays -- hence his apprehension (if not his irritation) over the delay was lessened.

One other approach taken in an attempt to reduce this apprehension was to have the prototype print "I am thinking" on the screen periodically during processing steps known to incur relatively long delays. Although this device may have lessened apprehension as to whether the system was down, or whether it had received the latest input, it did not appear to affect the major apprehension as to when a substantive response might be expected. Hence the device was dropped.

6.11 Structure of Interaction

H-23. The PM will rapidly attain the "direct" structure of interaction with the prototype.

This hypothesis is supported by observations of prototype use. Although the designer was present at a number of user sessions and often answered questions, the typical user learned very rapidly to handle prototype mechanics on his own. Three of the four non-investment users, in fact, began use of the system alone with essentially no tutorial help at all, after having read a brief note on the use of the system. All of the investment group had someone available for their first session to answer questions, but there was very little observed need for such help after the first hour or so. The implication is that the direct structure of interaction in a MMDS is feasible for a non-computer oriented decision maker. This feasibility, of course, depends greatly on the language and form of the MMDS, and how it maintains a simple yet flexible appearance to the user.

6.12 Form and Language of Interaction

H-24. The PM's will find the graphic displays attractive and make heavy use of them.

Table 6.2 indicates clearly that all users, investment professionals and others alike, made heavy use of the graphic functions, HISTO and SCATTER. These two functions constituted 42% of prototype usage by the investment group and 38% for the others.

Also, this high use of graphics seemed to hold for more experienced users as well as inexperienced ones. In particular, Table 6.7 compares prototype function usage for investment users who logged more than three hours on the prototype versus investment users logging less than three hours. It indicates that inexperienced investment users applied the graphic functions, HISTO and SCATTER, 37% of the time versus 45% for the more experienced users. This suggests that the graphic displays may have a continuing attraction for the decision maker beyond just a glamorous initial appearance.

H-25. Keyboard input will be quite acceptable to the PM.

Although typing errors were made by all users, there was never any indication of serious frustration with the keyboard input medium. This was true despite the fact that some types of erroneous input could cause the system to "crash", requiring a minute or so delay to restart and return to the same point again. More comprehensive input controls would have reduced this user irritation with such "fatal" errors even further. This finding is directly contrary to the prevailing wisdom in some circles which says that a manager will not use a typewriter keyboard to communicate with a computer. Once again, however, this hypothesis requires the test of time and operational usage. The result here was simply that there were no unsolicited complaints about the keyboard input form, but there was no comprehensive attempt made to get user reactions to the keyboard.

This result may have been helped in part by the fact that the terminal was in a room by itself. Thus, in general, the users were not widely observed while typing. If the terminal had been located in the middle of the open trust department floor, the results might have been different. That is, the possibility of a felt role conflict (e.g., a "manager" using a "clerk's" tool) might have been stronger had the PM been more widely visible to others. On the other hand, the PM's were already somewhat conditioned to keyboard use in that there was a Telequote terminal on the trust department floor that was heavily (and visibly) used by PM's to retrieve the latest price and volume figures on selected stocks.

Note that alternative input media were not tried, and therefore there is no necessary implication that the users would not have found light pen or "mouse" input attractive as well, or that a mixed-media configuration might not have worked well also.

6.13 Flexibility of Interaction

H-26. The PM will make full use of the flexibility provided in form of expression.

There is no question that the users exercised the stacked-command and direct command forms of expression, as well as the standard menu-selection mode. This is shown clearly in Tables 6.3 and 6.4. Table 6.3 gives the proportion of input lines falling into the classes of normal menu-selection response, direct command, or request for information (e.g., such commands as DIR or DATA, which ask for supplementary information on the directory or data variables in the list). Either the direct command or the menu-selection responses may be stacked. Table 6.4 shows the degree of such stacking employed by the user (e.g., "HISTO ACME" is a stacked direct command of length 2). This table shows that investment users stacked commands 8% of the time and non-investment users 5% of the time.

There do not appear to be any substantial differences between investment and non-investment users in their frequencies of direct or stacked-command usage. In particular, the null hypothesis test

Form of User Input	Professional Investment Users: Number of Input Lines	%	Non-Investment Users: Number of Input Lines	%
Menu-Selection	1318	92.2	1047	92.1
Direct Command	77	5.4	52	4.6
Information Request	35	2.4	38	3.3
Total	1430	100.0	1137	100.0

TABLE 6.3: Distribution of Type of User Input

Number of Stacked Commands per Line	Professional Investment Users: Frequency	%	Non-Investment Users: Frequency	%
1	1317	92.0	1066	94.8
2	48	3.4	36	3.1
3	45	3.2	23	2.0
4	17	1.2	9	.8
5	3	.2	3	.3
Total	1430	100.0	1137	100.0

TABLE 6.4: Distribution of Degree of Command Stacking

for homogeneity of two samples of direct versus menu-selection usage of Table 6.3 yields no significant difference between investment and non-investment users (Chi square = .626, DF = 1, 2-tail significance = .4288). This result is still true when the test is extended to cover the full pattern of use of menu-selection, direct command, and information request forms (Chi square = 2.612, DF = 2, 2-tail significance = .2709). Similarly, the test applied to the two samples of stacked command usage of Table 6.4 shows no significant basis for rejecting the null hypothesis (Chi square = 4.344, DF = 4, 2-tail significance = .3614). Note that the frequency of stacked command use decreases monotonically with increasing stack length, as one would expect.

Thus the basic result here is that users will apply accelerated language facilities in combination with the more primitive menu-selection form. In other words, the feasibility has been demonstrated of the language design tactic of maintaining a primary and simple language form in combination with more powerful forms that remain invisible to the beginning user. The idea is that the beginner can use only the menu-selection form without being confused by the complexity of alternative forms, and yet the transition is easy and direct when he does wish to move faster.

Table 6.5 shows evidence of this transition to the more accelerated language forms by comparing investment users who have logged more than three console-hours on the prototype with those

Type of User Input	Users Logging Over 3 Console-Hours		Users Logging Under 3 Console-Hours	
	Frequency	%	Frequency	%
Menu-Selection	1735	92.0	630	92.2
Direct Command	106	5.7	23	3.4
Information Request	43	2.3	30	4.4
Total	1884	100.0	683	100.0
Number of Stacked Commands per Line	**Frequency**	**%**	**Frequency**	**%**
1	1714	91.0	669	98.0
2	74	3.9	10	1.5
3	65	3.5	3	.4
4	25	1.3	1	.1
5	6	.3	0	.0
Total	1884	100.0	683	100.0

TABLE 6.5

Frequency of Use of Command Forms
for Long-Term Versus Short-Term Users

who logged less. The net result is a substantial increase in proportion of direct command use (from 3.4 to 5.7%) for the more experienced users. Applying the test for homogeneity in menu-selection versus direct command usage of the two samples of Table 6.5 show the null hypothesis rejected at the 5% level (Chi square = 4.466, DF = 1, 2-tail significance = .0346). There is also a substantial increase in the stacking of commands. Applying the test for homogeneity to the two samples of stacking patterns in Table 6.5 show the null hypothesis rejected very strongly (Chi square = 38.151, DF = 4, 2-tail significance $<$ .001).

Note also that the less experienced users made relatively heavier use of the direct command form than the stacked form. One possible explanation for this pattern is that the menu prompted use of direct commands by associating the appropriate direct command mnemonic with each available selection. (See, for example, Figure 5.7 in Section 5.8, which shows how the prototype menu places all direct command mnemonics within apostrophes.) Another possible explanation is that the direct command is easier to learn, being a mnemonic, than is the stacked command, which requires remembering the next question and its numerical answer in a particular dialogue before it is displayed. In any case, it should be noted that even the more experienced users continue to rely heavily on the unstacked menu selection form of expression.

Incidentally, note that the less experienced users made substantially greater use of requests for explanatory information, 4.4% versus 2.3% of the time for experienced users. The homogeneity test applied to the two samples of information requests versus all other inputs shows the null hypothesis rejected at the 1% level (Chi square = 7.332, DF = 1, 2-tail significance = .0068). This is to be expected, since more experienced users will begin to remember code names for portfolios, data items, etc., and will not require so much prompting from the system.

One question for the MMDS designer suggested by the increase in direct and stacked command usage with experience is, what will the long run result be? It might be expected that even the combined direct command and stacked command facility may become too verbose for experienced users. In particular, the stacked command form as realized on the prototype is neither infinitely extensible nor general purpose (e.g., in some situations, the prototype would reject all stacked input beyond a second or third item); the most experienced users had already begun to reach beyond its limits. One answer may be a general, user-controlled macro-building facility such as that discussed in Chapter 4. In the prototype application the likeliest candidates for such a facility would be the graphic functions, both of which were heavily used and both of which are relatively complex to specify. Having several preprogrammed and specified formats callable by a single name may be particularly attractive here.

6.14 Learning and Training

H-27. The PM will easily and quickly attain the enactive stage, or mastery of the mechanics of the prototype.

H-28. The PM more slowly will attain the iconic state, the understanding of the prototype as a mosaic of independent functions.

H-29. The PM will only very slowly attain the symbolic stage, the integration of planned sequences of system functions into his decision process.

The only evidence which is brought to bear on these hypotheses is observational. As for H-27, it was noted that no user appeared to experience any substantial difficulty in mastering system mechanics by the first hour of usage. This is consistent with results of another recent MMDS experiment (Gerrity and Black, 1970), in which a substantial majority of a subject group of 51 managers adapted to and made effective use of an aggregate production scheduling MMDS within one hour of beginning system use. Thus H-27 is somewhat supported both by direct observation of user experience with mechanics of the prototype, and by scanning the usage traces.

For hypotheses H-28 and H-29, the general impression was that most users did not attain a comprehensive understanding of all of the functions on the system during their limited period of prototype usage and that virtually none of them attained the so-called symbolic stage where they showed an ability to weave functions together in a planned and logical sequence toward a decision goal. In some situations, the users expressed a certain logic in going from one function to the next, but that linkage was seldom apparently more than two deep (i.e., function switching behavior might have been described by a Markov process, at best).

Part of this apparent slowness in full learning might be explained by the fact that many of the facilities provided in the prototype represented the possibility of a major departure from the PM's earlier decision procedure. As such, it may be reasonable to expect a longer learning period than one to seven hours before the PM demonstrates substantial control over the MMDS functions in integrating them into his decision process. Another reason for the slowness, of course, might be the non-operational character of prototype usage; i.e., the exercise had little "live" quality to it, and hence the motivation to learn rapidly might well have been less than in operational use. A final reason, of course, might be the fact that the planned facilities for hypothetical portfolio design were not implemented, and thus the prototype did not provide full continuity in decision support.

In any case, the detection of the usage patterns which might indicate cognitive growth beyond the enactive stage is not easy. No rigorous operational measures had been developed in the original design, and none were applied here. It is expected, however, that continual protocol collection may be the appropriate instrument.

Otherwise the researcher is left with the extremely difficult and not very rigorous approach of making inferences as to the logic behind a particular series of steps shown in a user trace.

In summary, the three hypotheses above simply say that a user of an MMDS will pass through three stages of cognitive growth -- enactive, iconic, and symbolic -- in developing an understanding of the system. The implication for the MMDS designer is that he should not misread a very early mechanical facility with the system on the part of the user as a fully successful integration of man and machine -- it is only a first step in what could be an extended period of evolution.

H-30. The PM will learn and use the familiar functions most readily.

This hypothesis appears not to be supported by the evidence. At least Table 6.2 shows the investment users applying the STATUS function, the most familiar to them, only 6% of the time. Also, Table 6.6 does not show any pattern of heavier usage of STATUS by less experienced users. In fact, the table shows the relative usage to be the same at 6% for both groups. One explanation might be the non-operational nature of the system usage combined with the attraction of the graphic functions. Evidence from long-term operational usage is required here.

6.15 Decision Programming

> H-31. Extensive use of the system will lead the PM to a more explicit and structured model of his decision process, which will lead in turn to suggestions for programming further phases or functions in the process.

The first part of this hypothesis is clearly much more difficult to test than the latter part. In fact, no attempt was made to derive the instruments to test the first part, partly because of the difficulty of the task and partly because of the very limited use of the prototype. Even if the hypothesis were true, one would expect that a significant amount of operational MMDS usage would have to occur before significant changes in decision process perception result.

On the other hand, there appears to be some support for the latter half of the hypothesis. The investment users, particularly those with the most prototype experience, did provide spontaneously a variety of suggestions of other functions that might be programmed. These suggestions were often about functions which the PM now tried to perform mentally and subjectively, and which he now saw (by analogy with demonstrated prototype functions) could be partly or fully programmed. For example, during use of STOCK in one session, the PM remarked that he would really like a function that scanned holdings of groups of securities as well as single securities. That is, he would like to have a search mechanism to tell him which portfolios hold, for example, aerospace stocks and how much they hold. He found that he often tried to perform operations like this manually anyway, and the analogy to the STOCK function suggested the possibility of programming it. Note that this observation should also mean some feedback to the original descriptive model, because this particular type of search operation was not observed there. Similarly, such observations may have implications for revision of the normative model as well.

Over a larger period of MMDS usage and with an operational system, one would expect to observe more of such suggestions being made, insofar as the user can discover new and attractive programmable functions within his decision process. If a macro-building facility like that discussed earlier were available to the user, there would be a potential for more rigorous monitoring of this decision programming process than the observational approach above. In other words, the monitor trace could show quite clearly the pattern of use of the macro-building facility as the user began to transfer more and more procedural information to the machine.

6.16 Other Observations on System Usage

The previous sections have outlined results and observations as they related to specific hypotheses. This section will describe other general observations on the experimental use of the prototype. In particular, observations will be made on the following:

(1) problems in the user's perception of overall prototype structure and his immediate context in the man-machine conversation;

(2) the relatively high investment user focus on portfolios rather than stocks, as opposed to non-investment users;

(3) an apparent lack of correlation between prototype function complexity and low function usage;

(4) an apparent shift in function usage over time toward SCATTER and away from HISTO and FILTER;

(5) strongly expressed user sensitivity to problems in system or data quality.

Some inferences will be drawn regarding possible explanations for these observations.

One observation concerns the user's perception of the prototype system. Although the users adapted readily to the mechanics of the system, they did appear to have an initial sense of "feeling lost" in the total system context, despite the everpresent "menu" which told them what their immediate alternatives were. This uncertainty gradually decreased for more experienced users, but it suggests a need for a broader presentation of total system structure to early users. One technique might be to provide access to a moving conversational graph display, showing the user's position and immediate context in the graph at several possible levels of detail (e.g., similar to the Chesler and Turn system at RAND, or the AESOP system at Lincoln Labs). Thus the user could stop at any point and ask for a graphic picture of where he was within the context of the total system. Another approach, of course, might be to do more thorough pre-training of system users, explaining the conversational graph structure in detail and providing each with a reference copy. Note that this problem of inadequate user perception of his information and decision system aids is not limited to highly interactive systems but is quite general in incidence. The many anecdotes about laborious and redundant manual processes going on right down the corridor from a computer system designed to perform the same function are symptoms of the problem of inadequate formal information system perception. It is an area that demands further research into more effective techniques of system documentation, representation, and communication.

A further examination of investment user versus non-investment user behavior serves to emphasize the very high interest of investment users in portfolio analysis as opposed to stock analysis. Table 6.1 shows that investment users use operands that indicate either individual portfolio or cross-portfolio analysis (portfolio or stock operands) 91% of the time, with only 9% allocated to stock list operands. Non-investment users, on the other hand, used stock list operands 53% of the time, portfolio-related operands 47% of the time. This difference in focus is further emphasized in Table 6.2 which shows investment users calling functions that only manipulate stock lists (CREATE, DELETE, RENAME) 2% of the time, versus 12% for the non-investment users. Tables 6.3 and 6.4, on the other hand, showed that both user groups have similar language usage "profiles". Thus the primary differences in prototype usage behavior between the two groups seem to revolve about the focus of the investment professionals on portfolios as opposed to individual stocks and their descriptors. This behavior, of course, is counter to previously observed PM behavior outlined in the descriptive model of Section 5.4 which shows much PM attention directed toward individual stocks. It may well be that this previous behavior is largely a function of limitations in their current information system, since evidence here shows that, given portfolio analysis mechanisms, the PM will tend to shift more attention to portfolios.

One result which was somewhat unexpected was the relatively low usage of the TABLE function, only 6% for investment users (see Table 6.2). It was felt by the designer that this was a relatively powerful tool for bringing together portfolio information and stock information in one report. There are at least the following three obvious possible explanations for this usage pattern:

1) the function is simply not attractive or useful to the user;

2) the TABLE function was implemented later than other functions, in September, 1969, and this may have biased later usage downward, as well as having prevented use in July - August;

3) the TABLE function is probably the most complex function to specify and to use, and this may have helped to inhibit use.

On this latter possibility, some analysis seems to indicate the somewhat obvious point that simplicity alone does not lead to high function usage, nor does complexity necessarily lead to low usage. Table 6.6 lists the functions of the prototype ranked in ascending order of complexity of use, as perceived by the author. Complexity is treated here as corresponding roughly to the number of lines of dialogue and the number of different parameters that must be specified to apply a function. The table indicates no clear correlation between function simplicity and degree of use.

The first possible explanation listed above, that TABLE was simply not attractive or useful to the user, was not investigated directly by interview or questionnaire. On the other hand, in the few console sessions observed by this investigator, there was no indication that users showed any of the same level of excitement or interest over the TABLE function as they did with the HISTO or SCATTER functions.

Prototype Functions (in ascending order of complexity)	Frequency of Function Usage (in percent)	
	Professional Investment Users	Non-Investment Users
ACCOUNT	3. %	2. %
STOCK	18.	10.
STATUS	6.	8.
DIRECTORY	11.	11.
DELETE	1.	3.
RENAME	0.	1.
SCATTER	21.	18.
HISTO	21.	20.
FILTER	12.	11.
CREATE	1.	8.
TABLE	6.	8.
Total	100.	100.

TABLE 6.6

Frequency of Function Usage Versus Function Complexity

The second possible explanation, that the late implementation of the TABLE function may have artificially reduced its usage did appear to have some substance. When function usage statistics were analyzed for prototype use only after September, 1969 (rather than July, 1969, when first usage began), it was found that TABLE was used 9% of the time by investment users and 14% of the time by others, a substantial increase over the full period statistics of 6% and 8% respectively.

A further comparison of short-term versus long-term prototype users, showing their relative use of system functions, is outlined in Table 6.7. The difference between these two samples is significant. The homogeneity test (grouping CREATE, DELETE, and RENAME into one category to avoid an overly distorted sample) shows the null hypothesis rejected at the 5 percent level (Chi square = 16.074, DF = 8, 2-tail significance = .0413). Looking at the details of the function usage patterns, the only substantial differences appear to be in usage of FILTER, HISTO, and SCATTER. It seems that FILTER and HISTO were relatively less attractive for the long-term users and SCATTER more attractive. This may be because SCATTER, in a rough way, provides some of the same information as FILTER and HISTO. It graphically shows the ranking of all stocks in a list along any two selected dimensions, and it gives a graphic sense for distribution of holdings (though not by market value) along those

Prototype Functions	EXPERIENCED USERS: Investment Users Logging Over 3 Console-Hours Frequency	%	LESS EXPERIENCED USERS: Investment Users Logging Under 3 Console-Hours Frequency	%
DIRECTORY	31	11.	14	13.
CREATE	4	1.	1	1.
DELETE	3	1.	0	0.
RENAME	0	0.	0	0.
STATUS	16	6.	6	6.
FILTER	29	10.	19	18.
TABLE	17	6.	7	7.
HISTO	55	19.	27	25.
SCATTER	73	26.	10	9.
STOCK	50	17.	19	18.
ACCOUNT	9	3.	3	3.
Total	287	100.	106	100.

TABLE 6.7

Frequency of Use of Prototype Functions
for Long-Term Versus Short-Term Users

same two dimensions. It also has one advantage over HISTO in that it identifies the specific stocks in the distribution by their ticker symbols. It has a similar advantage over FILTER in that it allows the user to examine the full range of list values, not just those that passed the filter limit.

Another phenomenon that was observed was a general user sensitivity to data quality and accuracy. The data base was essentially the same one that the investment professionals had available in a variety of forms in their original information system, with the same occasional errors that characterized that original data base. Yet on-line access to that same data via a conversational computer system seemed to heighten user sensitivity to errors. The reaction was typically one of irritation and complaint. It seems that the increased rigor of the functions the user had available may have made problems in data quality more salient. John Little notes that managerial use of management science models has a similar sort of effect (Little, 1970).

This sensitivity to data quality seems to be just one aspect of a more general phenomenon: MMDS user sensitivity to any problems in system or data quality, accuracy, currency, response time, or ease of access. In other words, it may be expected that a decision maker will demand more from a MMDS in terms of fast response, freedom from bugs, easy access and control, etc., than he normally demands from his batch computer or manual information system aids. An analogy may be drawn here with the problems of scheduling a production system globally and in real time. With control that is highly decentralized and periodic in application, a production organization (or any organization, for that matter) will tend to develop "buffers" such as extra inventory and slack resources so as to decrease each unit's sensitivity to the behavior of the rest of the organization. When such an organization is controlled centrally and in real time, part of the payoff of this tight control derives from elimination of these costly buffers. However, removal of the buffers make each unit highly interdependent and much more sensitive to the behavior and timing of others. What was before a tolerable degradation in timing or responsiveness now becomes intolerable. An analogous situation exists for a decision system. Part of the payoff of a MMDS may come from tighter timing and more global application of the decision functions. One cost of such a system may be, however, heightened user sensitivity to degradations in service that may have been tolerable when the user had adapted his decision making style to a less responsive system.

6.17 Summary

This chapter has described the experimental use of the prototype MMDS and made some observations on the application of the design methodology of Chapter 4. The period of use of the prototype was from July, 1969, to February, 1970, and it involved eleven investment professionals and four non-investment users, all four of whom were computer professionals. The subject use of the prototype system involved 49 console sessions logging a total of 50 console-hours. The results of this experimental use were compared with the design assumptions and expectations described in Section 5.10 and were organized in the same outline as the MMDS theory presented in Chapter 3. The machine monitor traces plus direct observation provided the basis for the described results.

One of the most salient observations on system use was that the subjects generally liked the prototype and saw its potential as a decision aid as great. The subjects found the mechanics of the system relatively easy to handle, were not apparently inhibited by keyboard input, and could generally operate the system effectively after less than an hour of use. Effective integration of system functions into the user's portfolio decision process, however, seems to be a much longer process, and no user appears to have achieved it to any substantial degree.

One major difference observed between investment and non-investment users was that the former focused substantially greater attention on portfolio analysis as opposed to stock analysis. Since this behavior is somewhat counter to pre-MMDS behavior, it suggests a need for portfolio analysis that has not been satisfied by the current portfolio management information system. In addition, the investment users made especially high use of those functions which provided an aggregate rather than a detailed view of portfolio status; in particular, the graphic functions were heavily used. There was also a great deal of attention to comparison of different portfolios, as well as a rapid switching of attention from portfolio to portfolio -- neither of these activities were characteristic of pre-MMDS decision behavior and they suggest a need for aggregate portfolio analysis and comparison that is not satisfied or feasible in their traditional information system.

Another characteristic of the prototype users was an extremely high sensitivity to any degradation in system response, accessibility, reliability, or data quality. This sensitivity seems substantially higher than that of research users of interactive systems in academic environments that the author has observed. It also seems considerably higher than prototype users' sensitivity to similar problems in their existing batch or manual information systems.

The prototype users appeared to learn rapidly and to demand continually more of the prototype. This was indicated in a tendency to evolve toward more direct and concise forms of command expression with system usage. It was also revealed in a tendency for the experienced users in particular to suggest additional functions to be added to the system as well as extensions of functions already present.

The principal observation on the design methodology was that it requires more comprehensive testing beyond this limited application, particularly since some aspects of the methodology were developed in the course of the design. There is a need for much more experience and even controlled experimentation with design methodologies before generalizations can be made. Nonetheless, the single design experience here does indicate that the methodology is workable, and that it can lead to identification of problems in decision system behavior, association of programmable operators with those problems, and eventually to some change in the problem behavior largely in the way expected by the MMDS designer.

CHAPTER 7

CONCLUSIONS AND DIRECTIONS FOR FURTHER WORK

This dissertation has asserted the importance of research into the behavior and design of man-machine decision systems (MMDS). It has contributed to this research by developing a general decision system framework to serve as a guide to representation of MMDS, as well as a link to relevant behavioral science and information systems literature, for both the MMDS researcher and designer. This general framework was used as a key component in the development of a model-based MMDS design methodology and in the elaboration of a set of MMDS design principles. Finally, the design methodology and general decision system framework were exercised and evaluated by the design and use of a prototype MMDS for portfolio management in an actual field situation.

The general results of this thesis have raised a number of issues for MMDS theory and design. This chapter discusses these issues and the next directions they suggest in developing MMDS theory, research instruments, and design methodology.

MMDS Theory

The general decision system framework developed here has focused on the interrelationship of man, machine, and decision task to produce decision making behavior. There was no formal attempt to link any particular form of decision behavior to an objective function for the decision. In other words, the judgment as to what constituted "good" or "desirable" decision behavior was left relatively unstructured. Thus the relationship between general characteristics of decision behavior and overall decision performance in a specific non-programmed decision task remains to be fully explored in the development of theory of MMDS.

In the context of the prototype MMDS design, a number of specific hypotheses and designer assumptions concerning MMDS behavior were raised here. Although most of these hypotheses were either supported or qualified in light of actual prototype usage experience, none of them were fully tested here. Therefore, many individual hypotheses proposed here deserve further in-depth investigation both in the laboratory and in the field.

One particular observation of the prototype usage experience was that user behavior patterns appeared to continue to evolve and change even after the first two or three hours of system use (Hypotheses H-26 and H-31). This evolutionary behavior has been predicted by several researchers, but it has received strikingly little direct investigation to date. In fact, beyond the work here, it appears that only McKenney (1967) and Scott Morton (1967) have followed the changes in the behavior of their subject decision makers over any appreciable period of time. If evolutionary growth is expected to be a prime characteristic of MMDS, then it deserves much further study in the form of longitudinal observation of MMDS behavior (focusing especially on the relationship between MMDS flexibility and evolution). To some extent, of course, the lack of such studies may be explained by the difficulties of maintaining the interest and attention of subjects for laboratory MMDS experiments over long periods of continual usage (e.g., over months). Field studies have the advantage of employing decision makers who are motivated to do their job (and thus, presumably, to use the MMDS) without need for artificial rewards, but they have the disadvantage of difficulty in control. An additional difficulty at present is that this investigator knows of no existing field situations where management decision makers are interacting in a substantial way with a complex MMDS in order to make unstructured decisions (although use of computer-aided-design facilities for engineering problem-solvers may be of interest here)--thus the researcher may have to build his own MMDS, which was the approach taken here with the prototype system.

Another consistent observation on the prototype usage experience was the apparent marked change of focus of decision maker attention from very local aspects of the problem (individual stock holdings) to a more global view (whole portfolios and cross-portfolio analyses). This observation arose in several individual hypotheses (H-1, H-2, H-3, H-8, H-19, H-24). It arose in a slightly different way in the apparent global comparison activity (portfolio comparison) on the part of the decision makers (H-7 and H-16). It seems clear that the prototype provided an attractive overview of the problem which had been impossible to achieve previously and whose absence had left the decision maker to apply a series of local heuristic strategies in order to do his job. This global view is effectively a new representation of the problem space with several levels of hierarchy in form as advocated by many (Newell, 1966; Oettinger, 1965; Simon, 1969). The impact of such a global MMDS view requires further longitudinal investigation both in portfolio management and in other decision situations.

Another general observation on the prototype usage was the apparent greater attractiveness of transformation or recoding operations as opposed to filtering or searching operations (H-9, H-24). This relates directly to the experimental findings of Newman and Rogers (1966) and Sweetland (1964). Note that this may be somewhat counter-intuitive in the sense that many writers about man-machine systems seem to view rote information retrieval or data base filtering as the two primary operations for such systems. Again, more longitudinal studies over a broader set of problem types is required.

Instruments in MMDS Research

One striking result of the prototype MMDS experience was the apparent effectiveness of the machine monitor in producing rich traces of MMDS behavior. A capability has been demonstrated to link detailed man-machine function usage to detailed decision behavior, thus going far beyond mere "user accounting," the primary purpose of such monitor systems to date. The unobtrusiveness and relative efficiency of this monitor mechanism may make it a highly effective research tool, especially in field studies where longitudinal observation can be so cumbersome and costly otherwise.

By scanning such a trace, it may be possible for the decision systems analyst or researcher to recreate to a large extent the objectives, logic, and general "feel" of the user session. This is possible because the form of the trace displays the full user input message as well as all standard machine response dialogue. The only part of the "conversation" that is missing is the precise content of information displays seen by the user, which are a function of the state of the prototype data base at the time of the session. Thus, in the prototype system, for example, the trace analyst can follow the full dialogue which specifies a HISTO display, but he does not see the final display itself.

It is feasible to build a further system to actually replay the terminal output and user input of the session on the terminal and with the appropriate timing, based on the monitor trace. The displays produced would, of course, be based on current status of the MMDS data base rather than its status when the trace was generated. Otherwise, with few exceptions, such a system could interpret and execute the sequence of user command inputs just as the prototype did at the original session. The few exceptions would invoke command sequences which were, in fact, data base dependent. For example, in the prototype, the user might have asked to "DELETE ACME," to erase a stock list called Acme. On the replay of the trace, of course, Acme would have been deleted already and the system would ordinarily switch to a different response procedure than it did at the original session. This "trace interpreter" system would have to have a capability to recognize and recover from such exceptions while still reproducing the bulk of the original session.

A further MMDS monitor alternative would be to tape record all of the man-machine dialogue written on the terminal screen. (At least one storage tube terminal manufacturer provides this facility.) In fact, this display recording taken in synchronization with a verbal tape of the user's decision protocol could provide a powerful combined tool. Such a dual tape recording would not allow for much complex processing beyond simple playback, but it might provide a very rich representation of MMDS behavior for the early, exploratory stages of a research project.

A further observation on use of the machine-readable trace is that a special purpose MMDS for trace analysis might be useful. The trace is one representation of some rather complex decision behavior--the number of possible interesting analyses of the trace are huge, and the variety in approaches to the raw trace data manipulation and aggregation is large. In addition, the kind of questions the analyst would ask of the trace data base are difficult to structure in advance to any great degree.

For example, one idea for an interesting ad hoc analysis that came to mind very late in this thesis research was to look at the frequency of use of direct and stacked commands across various prototype functions. The object of the analysis would be to see if high usage of more concise forms of expression correlated with functions that were heavily used--if so, this would suggest an attempt on the part of the user to recode his decision tools so that high usage procedures could be called more concisely. However, given that the trace was in the form of several inches of computer printout, rather than in machine-readable form, the prospect of performing such a special analysis becomes distinctly unpalatable. Had the trace been maintained in a system for man-machine analysis, however, such an unplanned and exploratory analysis would have been more attractive.

Essentially, the trace analyst is involved in decision modelling--the building and testing of hypotheses about MMDS decision behavior. This is a highly unstructured problem in itself, yet one involving a large data base and a number of programmable statistical and logical operations--in other words, decision trace analysis has characteristics which make it amenable to MMDS support.

Another instrument that was used in this research was the Role Construct Repertory test of Kelly (1955). Although the test was used only on a tentative basis here, it deserves further investigation as a possible measure both of the dimensionality and complexity of the decision maker's discrimination ability within a given decision task and of changes in that ability with time and MMDS usage. Recently, Riesing (1970) has proposed evaluation of the test as extended by Wilcox (1970) as a general measure of level of information processing activity by managers as well.

MMDS Design Methodology

The MMDS design methodology (as well as the general design methodology) proposed here has some unique characteristics which deserve further testing. For example, the methodology requires use of explicit models of human decision systems to guide current decision system analysis, as well as specification of desired and predicted MMDS behavior. One weak test of the approach is that it appears to produce an attractive, workable MMDS; that is, variations on this decision-centered approach have worked for this investigator, as well as for Scott Morton (1967) and Newman and Rogers (1966). Stronger, comparative tests of this and other particular characteristics of the proposed design methodology will be difficult to conduct. Nonetheless, comparative experiments and field surveys of the effectiveness of various MMDS design approaches should be attempted. At the very least, one could do a much better job than the literature to date of simply *describing* actual design methodologies used in practice, especially those which seem most effective.

In the application of the design methodology for portfolio management, it was noted that the prototype proved to be a very useful vehicle for stimulating involvement and design creativity on the part of the investment users. Design ideas and reactions had been solicited continuously through the project, from initial conceptualization through conversational graph specification, but the dialogue over the prototype was far the most fruitful. The users could be far more explicit and direct about the capabilities they desired with the prototype in front of them than they ever found possible with "paper" designs. As noted earlier, the prototype seemed to act as a catalyst in setting off this self-modeling activity by the users. The implication here is that it may be valuable in a MMDS design effort to bring the eventual users face-to-face with a simulated or prototype version of the expected system as early as possible in the process. If user involvement is valued highly in such a design effort, then this early prototype approach seems to be one effective way to get it.

Another specific MMDS design approach developed and applied here involved the conversational graph. As noted before, the form of the conversational graph representation of the man-machine interface suggests a variety of issues involved in the trade-off between generality and power versus simplicity of use in design of the language of interaction. One particular balance was selected for the prototype--a hybrid menu-selection, stacked-command, and direct command approach. This and other approaches to attaining flexibility and power in a MMDS need to be developed and evaluated. In other words, the prototype design only examined one "point" in the spectrum of possible language forms. What is required is a comprehensive approach to exploration of this MMDS "design space" and its mapping into MMDS behavior. The general decision system framework developed here is intended at least to structure the major dimensions of MMDS design parameters and behavioral characteristics.

In addition, MMDS researchers and designers need to experiment with the programming of more complex decision operators and plans than those attempted here. For example, only a very limited solution design capability was programmed in the prototype, and it was used relatively little. One possibility is that some decision makers (or perhaps managers as a class) do not like to design actively, but consider review and evaluation of alternatives already generated as more fitting to their decision style or role. In that case, the development and evaluation of programmed alternative generators within a MMDS might be worthwhile. Such a facility could be attractive for decision makers who consider it more their role to review and select, rather than to search and build.

The design methodology effectively asks the designer to invest extra resources in several tasks that don't seem directly related to "getting the job done" in a traditional sense. For example, the methodology emphasizes extra effort in terms of decision modeling over and above the traditional information system modeling, even when the designer may think he knows what functions the system ought to have. Finally, it emphasizes investment in design of a decision monitoring and control system that may seem like an unnecessary frill, especially under the pressure of the final phases of implementation. The logic behind these "extra" expenditures of design resources has been outlined in Chapter 4--what is needed is more experience and evidence of the kind described in Chapters 5 and 6 regarding the prototype MMDS. The feeling of the author after design of the prototype is that these additional design activities are useful, that they contribute to a more effective design, and that they are worth the extra effort. Obviously, much more experience, and even experiments with the design methodology are going to be necessary before this "feeling" is supported by substantial evidence.

Finally, the field of MMDS requires much more theory-building. The general decision system framework developed here needs further elaboration and refinement. Links between this framework and its base of behavioral science and information systems research need to be made stronger and more rigorous. The framework itself may benefit from modification and expansion. Theoretical results need to be integrated into a MMDS design methodology.

In short, there is much speculation, experimentation, theorizing, and laborious integration ahead of us in the building of a theory of MMDS design and behavior. This investigator hopes that the work here has contributed to that end.

BIBLIOGRAPHY

Ackoff, Russell L., "Management Misinformation Systems," Management Science, December, 1967, pp. B-147-56.

Allen, Thomas J., "Studies of the Problem-Solving Process in Engineering Design," I.E.E.E. Trans. Eng. Mgmt., June, 1966, pp. 72-83.

Allen, T. J., and Marquis, D. G., "Positive and Negative Biasing Sets: The Effects of Prior Experience on Research Performance," I.E.E.E. Trans. Eng. Mgmt., December, 1964, pp. 158-61.

Amstutz, Arnold, "The Computer: A New Partner in Investment Management," Alfred P. Sloan School of Management at M.I.T., Working Paper 283-67, September, 1967.

Amstutz, Arnold, "A Basic Market-Oriented Management Information System," Working Paper 307-68, Alfred P. Sloan School of Management at M.I.T., 1968.

Anshen, Melvin, "Managerial Decisions," from Automation and Technological Change, John T. Dunlop, Editor, Prentice-Hall, 1962.

Baker, James D., "From the Diet of Worms to the Bucket of Worms: A Protest Concerning Existing Display Dogma for Information Systems," Information System Science and Technology, Spartan Books, 1964.

Bauer, Walter F., "The Economics of On-Line Systems," from On-Line Computing: Time Shared Man-Computer Systems, Editor, Walter J. Karplus, McGraw-Hill, 1967, pp. 74-106.

Bavelas, A., "Communication Patterns in Task Oriented Groups," Journal of the Acoustical Society of America, 22 (1950), pp. 725-730.

Becker, Joseph and Hayes, R. M., Information Storage and Retrieval, Wiley, 1963.

Bell, Thomas E., "Graphical Analysis Procedures for Simulation and Scheduling," unpublished Ph.D. dissertation, U.C.L.A. School of Business Administration, 1968.

Betaque, N. E., and Gorry, G. A., "Automated Judgmental Decision-Making," Working Paper 416-69, Alfred P. Sloan School of Management at M.I.T., September, 1969.

Bieri, James, "Cognitive Complexity-Simplicity and Predictive Behavior," J. Abnormal and Social Psychology, 1955, 51, pp. 263-68.

Blume, Marshall E. II, "The Assessment of Portfolio Performance - An Application of Portfolio Theory," unpublished Ph.D. Dissertation, Graduate School of Business, University of Chicago, March 1968.

Blumenthal, Sherman C., Management Information Systems: A Framework for Planning and Development, Prentice-Hall, 1969.

Boehm, Barry W., "Keeping the Upper Hand in the Man-Computer Partnership," P-3533, The RAND Corporation, April, 1967.

Borko, Harold, "Utilization of On-Line Interactive Displays," Information System Science and Technology, Spartan Books, 1966, pp. 327-334.

Boutell, Wayne, Computer-Oriented Business Systems, Prentice-Hall, 1968.

Bowman, E. H., "Consistency and Optimality in Managerial Decision Making," in Industrial Scheduling, J. L. Muth and G. L. Thompson, Eds., Prentice-Hall, 1963.

Brady, Rodney H., "Computers in Top-Level Decision Making," Harvard Business Review, July-August, 1967, pp. 67-76.

Braybrooke, David, and Lindblom, Charles, A Strategy of Decision, The Free Press, 1963.

Bruner, Jerome, Goodnow, Jacqueline, and Austin, George, A Study of Thinking, Wiley, 1956.

Bruner, J. S., "The Course of Cognitive Growth," Journal of the American Psychology Association, Vol. 19, 1964, pp. 1-15.

Bryan, G. E., "JOSS: Accounting and Performance Measurement," RM-5217-PR, The RAND Corporation, Santa Monica, Calif., June, 1967.

Carbonell, James R., "On Man-Computer Interaction: A Model and Some Related Issues," I.E.E.E. Trans. Systems Science and Cybernetics, January, 1969, pp. 16-20.

Carpenter, Richard A., "The Interactive Computing Graph as a Tool in the Analysis of Conversational Computing Systems," unpublished Ph.D. Dissertation (in process), Massachusetts Institute of Technology, 1970.

Carroll, Donald C., "Implications of On-Line, Real-Time Systems for Managerial Decision-Making," from The Management of Aerospace Programs, Vol. 12, Science and Technology Series, 1967, American Astonautical Society, pp. 345-367.

Carroll, D. C., and Zannetos, Z. S., "Toward the Realization of Intelligent Management Information Systems," Working Paper No. 218-66, Alfred P. Sloan School of Management at M.I.T., Cambridge, Mass., October, 1966.

Charnes, A. and Cooper, W. W., "Data, Modeling, and Decisions," Management Sciences Research Report No. 43, Carnegie Institute of Technology, June, 1965.

Chesler, L., and Turn, R., "The Application of On-Line Graphical Techniques for Programming and Operating a 'Moving Network' Monitoring Display," RM-5183-PR, The RAND Corporation, January, 1967 (1967a).

Chesler, L., and Turn, R., "Some Aspects of Man-Computer Communication in Active Monitoring of Automated Checkout," P-3522, The RAND Corporation, March, 1967 (1967b).

Churchman, C. W., Prediction and Optimal Decision, Prentice-Hall, 1961.

Clarkson, G. P., Portfolio Selection: A Simulation of Trust Investment, Englewood Cliffs, New Jersey, Prentice-Hall, 1962.

Clarkson, G. P. E., and Pounds, W. F., "Theory and Method in the Exploration of Human Decision Behavior," Industrial Management Review, Fall, 1963.

Cohen, Kalman J., and Pogue, Jerry A., "An Empirical Evaluation of Alternative Portfolio-Selection Models," Journal of Business, Vol. 40, No. 2, pp. 166-193, April 1967.

Cyert, R. M., and March, J. G., A Behavioral Theory of the Firm, Prentice-Hall, 1963.

Davis, Ruth, "Classification and Evaluation of Information System Design Techniques," Information System Science and Technology, Spartan Books, 1964.

Davis, Ruth M., "Man-Machine Communication," Annual Review of Information Science and Technology, Editor, C. A. Cuadra, Vol. 1, Interscience Publishers, 1966, pp. 221-254.

Dearden, John, "Can Management Information be Automated?" Harvard Business Review, Vol. 42, No. 2, March-April, 1964, pp. 128-135.

Dearden, John, "Myth of Real-Time Management Information," Harvard Business Review, May-June, 1966, pp. 123-131.

Dill, W. R. in H. J. Leavitt and L. R. Pondy, Eds., Readings in Managerial Psychology, University of Chicago Press, 1964, pp. 459-60.

Dow Jones and Company, "Management and the Computer," Wall Street Journal Study, 1969.

Edwards, Ward, "Dynamic Decision Theory and Probabilistic Information Processing," Human Factors, Vol. 4, pp. 59-73, 1962.

Edwards, Ward, "Probabilistic Information Processing System for Diagnosis and Action Selection," Information System Science and Technology, Spartan Books, 1964, pp. 141-155.

Edwards, Ward, "Decision-Making is Too Simple for Executives to Waste Time on It," Innovation, No. 5, 1969, pp. 34-41.

Edwards, Ward, et al., "Emerging Technology for Making Decisions," New Directions in Psychology II, Holt, Rinehart and Winston, 1964, pp. 260-325.

Elithorn, Alick, and Telford, Alex, "Computer Analysis of Intellectual Skills," Int. Journal of Man-Machine Studies, Vol. 1, 1969, pp. 189-209.

Emery, James C., "Organizational Planning and Control: Theory and Technology," unpublished Ph.D. Dissertation, M.I.T., September, 1965.

Enthoven, A., "Systems Analysis and the Navy," Naval Review, U. S. Naval Institute, Annapolis, Maryland, 1965.

Erikson, Warren J., "An Analytical Cost Comparison of Computer Operating Systems," TM-3525, System Development Corporation, Santa Monica, California, 30 June 1967.

Ferguson, Robert L., and Jones, Curtis H., "A Computer Aided Decision System," Management Science, June, 1969, pp. B-550-61.

Festinger, Leon, A Theory of Cognitive Dissonance, Stanford University Press, 1957.

Festinger, Leon, Conflict, Decision and Dissonance, Stanford University Press, 1964.

Fox, Lawrence E., "The Chief Executive as a User of a Terminal Network," paper delivered at the Winter Symposium on Man-Machine Interaction of the Institute of Management Sciences, Gainesville, Florida, February 20, 1970.

Frischmuth, Daniel, and Allen, Thomas, "A Model for the Description and Evaluation of Technical Problem Solving," Working Paper 316-68, Alfred P. Sloan School of Management, M.I.T., March, 1968.

Gal, Joseph J., "Man-machine Interactive Systems and Their Application to Portfolio Management," Financial Analysis Journal, May-June, 1966.

Galbraith, Jay R., "Organization Design: An Information Processing View," Working Paper 425-69, Alfred P. Sloan School of Management at M.I.T., October, 1969.

Garrity, John T., "Getting the Most Out of Your Computer," McKinsey & Co., Inc., New York, 1963.

Gerrity, Thomas, and Black, Robert, "An Experiment in Man-Machine Decision Making," Working Paper (in process), Alfred P. Sloan School of Management at M.I.T., 1970.

Glans, Thomas B., et al, Management Systems, Holt, Rinehart and Winston, 1968.

Gold, Michael M., "A Methodology for Evaluating Time-Shared Computer System Usage," unpublished paper, Carnegie-Mellon University, August 1967.

Gold, Michael M., "Time-Sharing and Batch-Processing: An Experimental Comparison of Their Values in a Problem-Solving Situation," Communications of the ACM, May, 1969, pp. 249-259.

Gorry, G. A., "A System for Computer-Aided Diagnosis," MAC-TR-44, Project MAC, M.I.T., September, 1967.

Gorry, G. A., "Modelling the Diagnostic Process," Working Paper No. 370-69, Alfred P. Sloan School of Management at M.I.T., February, 1969.

Grant, E. E., and Sackman, H., "An Exploratory Investigation of Programmer Performance under On-Line and Off-Line Conditions," I.E.E.E. Transactions on Human Factors in Electronics, March, 1967, pp. 33-48.

Gregory, Robert, and Van Horne, Richard, Automatic Data Processing Systems, Wadsworth Publishing Company, 1963.

Guetzkow, H., and Simon, H., "The Impact of Certain Communication Patterns on Task Oriented Groups," Management Science, Vol. 1, Nos. 3-4 (August 1955), pp. 233-250.

Head, Robert, Real-Time Business Systems, Holt, Rinehart, and Winston, 1964.

Heilbroner, Robert L., "The Impact of Technology: The Historic Debate," from Automation and Technological Change, Editor, John T. Dunlop, Prentice-Hall, 1962, pp. 7-25.

Hill, Lawrence S., "Some Implications of Automated Design on the Engineering Process," P-3625, The RAND Corporation, July, 1967.

Hormann, Aiko M., "Designing a Machine Partner," Datamation, pp. 29-33, February 1967.

Hunt, Earl B., Marin, Janet, and Stone, Philip J., Experiments in Induction, Academic Press, 1966.

"I.B.M. Study Organization Plan," I.B.M. Corporation, Data Processing Division, White Plains, New York, 1963.

Johnson, Donald M., "Serial Analysis of Thinking," Annals of the New York Academy of Sciences, December 23, 1960.

Kaplow, R., Strong, S., and Brackett, J., "A System for On-Line Mathematical Analysis," MAC-TR-24, Project MAC, M.I.T., January, 1966.

Kelly, George, The Psychology of Personal Constructs, Norton, 1955.

Kotler, P., Marketing Management, Prentice-Hall, Englewood Cliffs, New Jersey, 1967, pp. 176-177.

Kriebel, C. H., "Operations Research in the Design of Management Information Systems," Chapter 22 in Operations Research in Design of Management Information Systems, Ed., John F. Pierce, Tech. Assn. of the Pulp and Paper Industry, 1967.

Kriebel, Charles H., "Perspectives on Information Processing in Management Information Systems," Management Science Research Report No. 171, Carnegie-Mellon University, August, 1969a.

Kriebel, Charles, "Information Processing and Programmed Decision Systems," Management Science, November, 1969b, pp. 149-164.

Lee, Robert S., "The Computer and the Concept of Man," from Conversational Computers, Editor, William D. Orr, John Wiley and Sons, 1968, pp. 218-223.

Licklider, J. C. R., "Man-Computer Partnership," _International Science and Technology_, May, 1965, (1965a), pp. 18-26.

Licklider, J. C. R., "Man-Computer Interaction in Information Systems," _Toward a National Information System_, Editor, A. Rubinoff, 1965 (1965b).

Licklider, J. C. R., "Principles and Problems of Console Design," _Proceedings of the I.F.I.P.S. Congress_, 1965c, pp. 508-9.

Licklider, J. C. R., and Clark, Welden E., "On-Line Man-Computer Communication," _Proceedings of the 1962 Spring Joint Computer Conference_, pp. 113.

Licklider, J. C. R., and Taylor, Robert W., "The Computer as a Communication Device," _Science and Technology_, April, 1968, pp. 21-31.

Little, John D. C., "Models and Managers: The Concept of a Decision Calculus," _Management Science_, April, 1970, pp. 466-85.

McEachern, A. W., and Newman, J. R., "A System for Computer-Aided Probation Decision Making," unpublished paper, 1969 (to be published in the _Journal of Research in Crime and Delinquency_).

McIntosh, S. D., and Griffel, D. M., "The Admins Primer--November 1968," Center for International Studies, M.I.T., November, 1968.

McKenney, James L., "A Clinical Study of the Use of a Simulation Model," _Journal of Industrial Engineering_, January, 1967, pp. 30-35.

McKenney, James L., "Computer Based Models as Adaptive Communicators Between Different Cognitive Styles," Working Paper No. 141, Graduate School of Business, Stanford University, June, 1968.

McKenney, James L., "An Approach to Simulation Model Development for Improved Planning," _Proceedings of the Fall Joint Computer Conference_, 1968b, pp. 411-16.

McKinsey and Company, "Unlocking the Computer's Profit Potential," Research Report to Management, 1968.

McLaughlin, Curtis P., "Computer Aids for Aerospace Design and Engineering," RM-5264-PR, The RAND Corporation, Santa Monica, California, July, 1967.

McLuhan, H. M., _Understanding Media: The Extensions of Man_, McGraw-Hill, 1965.

McWhinney, William, "Aspiration Levels and Utility Theory," in _The Psychology of Management Decision_, Editor, George Fisk, 1967.

March, J. G., and Simon, H. A., _Organizations_, Wiley, 1958.

Markowitz, Harry, _Portfolio Selection_, John Wiley and Sons, 1959.

Marquis, Donald G., "Individual and Group Decisions Involving Risk," Working Paper 322-68, Alfred P. Sloan School of Management at M.I.T., March, 1968.

Marquis, Donald G., and Reitz, Joseph, "Uncertainty and Risk Taking in Individual and Group Decisions," Alfred P. Sloan School of Management at M.I.T., March, 1968.

Martin, James, _Design of Real-Time Computer Systems_, Prentice-Hall, 1967.

Martin, William A., "Symbolic Mathematical Laboratory," MAC-TR-36, Project MAC, M.I.T., January 1967.

Meister, David, and Rabideau, G. F., _Human Factors Evaluation in System Development_, John Wiley and Sons, 1965.

Miller, G. A., "The Magical Number Seven, Plus or Minus Two," _Psychology Review_, 1956, _63_, pp. 81-97.

Miller, G. A., Galanter, E., and Pribram, K. H., _Plans and the Structure of Behavior_, New York, Henry Holt & Company, 1960.

Miller, James R., "On-Line Analysis for Social Scientists," MAC-TR-40, Project MAC, M.I.T., May, 1967.

Miller, R. B., "Task Description and Analysis," in R. M. Gagne (Ed.), _Psychological Principles in System Development_, New York, Holt, Rinehart and Winston, 1962.

Miller, R. B., "Psychology for a Man-machine Problem-Solving System," IBM Technical Report TR 00.1246, February 19, 1965.

Miller, R. B., "Archetypes in Man-Computer Problem Solving," _I.E.E.E. Trans. on Man-Machine Systems_, December, 1969, pp. 219-41.

Mock, Theodore J., "Comparative Values of Information Structures," _Empirical Research in Accounting: Selected Studies, 1969_, Graduate School of Business, University of Chicago, 1969.

Morton, Michael S. S., "Interactive Visual Display Systems and Management Problem Solving," Industrial Management Review, Fall, 1967, pp. 69-81.

Morton, Michael S. S., and McCosh, Andrew M., "Terminal Costing for Better Decisions," Harvard Business Review, May-June, 1968, pp. 147-156.

Myers, Sumner, and Marquis, Donald G., Successful Industrial Innovations: A Study of Factors Underlying Innovation in Selected Firms, National Science Foundation Report NSF 69-17, U. S. Government Printing Office, 1969.

Nadler, Gerald, "An Investigation of Design Methodology," Management Science, June, 1967a, pp. 642-655.

Nadler, Gerald, Work Systems Design: The IDEALS Concept, Richard D. Irwin, 1967b.

Neisser, Ulric, "MAC and its Users," MAC-M-185, Project MAC, M.I.T., September 29, 1964.

Neisser, Ulric, "Computers as Tools and Metaphors," from Conversational Computers, Editor, William D. Orr, John Wiley and Sons, 1968, pp. 206-217.

Ness, David N., "Interactive Budgeting Models: An Example," Working Paper No. 345-68, Alfred P. Sloan School of Management at M.I.T., 1968.

Newell, Alan, "On the Representation of Problems," Computer Science Research Review, Carnegie Institute of Technology, 1966, pp. 19-33.

Newell, A., Shaw, J. C., and Simon, H. A., "Empirical Explorations of the Logic Theory Machine," Proceedings of the Western Joint Computer Conference (February 1957), pp. 218-230.

Newell, A., Shaw, J. C., and Simon, H. A., "Elements of a Theory of Human Problem Solving," Psychological Review, 65, May, 1958.

Newell, A., Shaw, J. C., and Simon, H. A., "Report on a General Problem-Solving Program," Proceedings of the ICIP (June 1960). Reprinted in: Computers and Automation (July 1960), as "A General Problem-Solving Program for a Computer."

Newman, J. Robert, "Extension of Human Capability Through Information Processing and Display Systems," SP-2560, System Development Corporation, December 7, 1966.

Newman, J. R., and Rogers, M. S., "Experiments in Computer-Aided Inductive Reasoning," TM-3227, System Development Corporation, December 6, 1966.

Oettinger, Anthony G., "Linguistic Problems of Man-Computer Interaction," Proceedings of the IBM Scientific Computing Symposium on Man-Machine Communication, Yorktown Heights, New York, May 3-5, 1965.

Optner, Stanford, Systems Analysis for Business Management, Prentice-Hall, 1968.

Parkhill, D. F., The Challenge of the Computer Utility, Addison-Wesley, 1966.

Parnas, David L., "On the Use of Transition Diagrams in the Design of a User Interface for an Interactive Computer System," Proceedings of the National ACM Convention, August, 1969.

Patrick, Robert L., "Time-Sharing Tally Sheet," Datamation, pp. 42-47, November 1967.

Piaget, Jean, Six Psychological Studies, Random House, 1968.

Pogue, Jerry A., "An Adaptive Model for Investment Management," unpublished Working Paper, Alfred P. Sloan School of Management, M.I.T., January 1968.

Polya, G., How to Solve It, Princeton University Press, 1945.

Pounds, William F., "The Process of Problem Finding," Industrial Management Review, Fall, 1969, pp. 1-19.

Prywes, Noah S., "Associative Memory in Heuristic Problem-Solving for Man-Machine Decisions," from The Psychology of Management Decision, Editor, George Fisk, CWK Gleerup Publishers, Lund, Sweden, 1967, pp. 152-166.

Quade, E. S., "Systems Analysis Techniques for Planning-Programming-Budgeting," P-3322, The RAND Corporation, March, 1966.

Riesing, Thomas, "Management Control Systems: Management Performance and the Levels of Information Processing," unpublished Ph.D. Dissertation (in process), Alfred P. Sloan School of Management at M.I.T., 1970.

Rockart, J. F., "Model-Based Systems Analysis--A Methodology and Case Study," Industrial Management Review, Winter, 1969.

Rubinoff, Morris; Bergman, Samuel; Franks, Winifred; and Rubinoff, Elayne R., "Experimental Evaluation of Information Retrieval Through a Teletypewriter," Communications of the ACM, Vol. II No. 9, pp. 598-604, September, 1968.

Sackman, H., "Experimental Investigation of User Performance in Time-Shared Computing Systems: Retrospect, Prospect, and the Public Interest," SP-2846, Systems Development Corporation, May 5, 1967.

Sackman, H., "Online vs. Offline Programming," Data Processing Digest, Vol. 14, No. 5, pp. 1-14, July, 1968a.

Sackman, H., Computers, System Science and Evolving Society: The Challenge of Man-Machine Digital Systems, Wiley, 1968b.

Sackman, H., "Experimental Evaluation of Time-Sharing and Batch Processing in Teaching Computer Science," SP-3411, Systems Development Corporation, 24 October 1969.

Sackman, H., Erickson, W. J., and Grant, E. E., "Exploratory Experimental Studies Comparing Online and Offline Programming Performance," Communications of the ACM, Vol. II, No. 1, pp. 3-11, January 1968.

Sackman, H., and Gold, Michael M., "Time-Sharing Versus Batch Processing: An Experimental Enquiry into Human Problem-Solving," SP-3110, System Development Corporation, June 14, 1968.

Schatzoff, M., Tsao, R., and Wiig, R., "An Experimental Comparison of Time-sharing and Batch Processing," Communications of the ACM, Vol. 10, No. 5, pp. 261-265, May 1967.

Scherr, A. L., "An Analysis of Time-Shared Computer Systems," Project MAC Report MAC-TR-18 (Thesis), Massachusetts Institute of Technology, Cambridge, Mass., June 1965, 177 pp.

Scherr, A. L., "Time-sharing Measurement," Datamation, pp. 22-26, April 1966.

Schrage, Linus, "The Modeling of Man-Machine Interactive Systems," Report No. 6942, Center for Mathematical Studies in Business and Economics, University of Chicago, September, 1969.

Schroder, Harold, Driver, Michael, and Streufert, Siegfried, Human Information Processing, Holt, Rinehart and Winston, 1967.

Schultz, G. P., and Whisler, T. L., Editors, _Management Organization and the Computer_, Free Press of Glencoe, 1960.

Schum, David, "Concerning the Evaluation and Aggregation of Probabilistic Evidence by Man-Machine Systems," _Information System Science and Technology_, Spartan Books, 1966, pp. 337-347.

Scott Morton, Michael S., "Computer-Driven Visual Display Devices--Their Impact on the Management Decision-Making Process," unpublished Ph.D. Dissertation, Graduate School of Business, Harvard University, June 1967.

Scott Morton, Michael S., "Strategic Planning and Management Information Systems--Some Implications of Present Research," Working Paper No. 338-68, Alfred P. Sloan School of Management at M.I.T., Cambridge, Mass., June 1968.

Scott Morton, Michael S., "Strategy for the Design and Evaluation of an Interactive Display System for Management Planning," Working Paper No. 439-70, Alfred P. Sloan School of Management at M.I.T., February, 1970.

Selltiz, Claire, Jahoda, M., Deutsch, M. and Cook, S., _Research Methods in Social Relations_, Holt, Rinehart and Winston, 1966.

Shackel, B., "Man-Computer Interaction--The Contribution of the Human Sciences," _I.E.E.E. Trans. on Man-Machine Systems_, December, 1969, pp. 149-163.

Sharpe, William F., "A Linear Programming Algorithm for Mutual Fund Portfolio Selection," _Management Science_, Vol. 13, No. 7, pp. 499-510, March, 1967.

Shaw, J. C., "JOSS: Experience with an Experimental Computing Service for Users at Remote Typewriter Consoles," P - 3149, The RAND Corporation, Santa Monica, Calif., May, 1965.

Shuford, Emir, "A Computer-Based System for Aiding Decision Making," _Information System Science and Technology_, Spartan Books, 1964, pp. 157-168.

Simon, H. A., "A Behavioral Model of Rational Choice," _Quarterly Journal of Economics_, Vol. 69, 1955, pp. 99-118.

Simon, H. A., _Models of Man_, Wiley, 1957.

Simon, H. A., _The New Science of Management Decision_, New York, Harper and Brothers, 1960.

Simon, H. A., "On the Concept of Organizational Goal," Administrative Science Quarterly, 1964, pp. 1-22.

Simon, H. A., "Reflections on Time-Sharing from a User's Point of View," Computer Science Research Review, Carnegie Institute of Technology, 1966, pp. 43-52.

Simon, H. A., "The Future of Information Processing Technology," Management Science, May 1968, pp. 619-24.

Simon, H. A., The Sciences of the Artificial, M.I.T. Press, 1969.

Smith, Lyle B., "A Comparison of Batch Processing and Instant Turnaround," Communications of the ACM, Vol. 10, No. 8, pp. 495-500, August 1967.

Soelberg, Peer, "A Study of Decision Making: Job Choice," unpublished Ph.D. Dissertation, Carnegie-Mellon University, 1967a.

Soelberg, Peer, "Unprogrammed Decision Making," Industrial Management Review, Spring, 1967b, pp. 19-30.

Soelberg, Peer, "The Structure of Individual Goals: Implications for Organization Theory," in The Psychology of Management Decision, Editor, George Fisk, 1967c.

Sprague, Richard E., "Personalized Data Systems," Business Automation, October, 1969, pp. 43-51.

Sprague, Richard E., "The Browsing Era," Business Automation, June, 1967, pp. 53-70.

Stern, Harry, "Information Systems in Management Science," Management Science, October, 1967, pp. B-848-51, Vol. 13, No. 12.

Stimler, Saul, Real Time Data Processing Systems, McGraw-Hill, 1969.

Sutherland, Ivan E., "Sketchpad: A Man-Machine Graphical Communication System," TR-296, Lincoln Laboratory, M.I.T., January 30, 1963.

Swanson, Andrew G., "Selection of Project Managers in a Government Research Laboratory: A Study of the Decision Process," unpublished Masters Thesis, Alfred P. Sloan School of Management at M.I.T., 1964.

Sweetland, A., "The Use of Computers in Air Force Maintenance Management and Analysis," The RAND Corporation, RM-4228-PR, October 1964.

Utterback, James M., "A Model of Innovation in the Industrial Firm," unpublished Ph.D. Dissertation proposal, Research Program on the Management of Science and Technology, M.I.T., April 7, 1968.

Vazsonyi, Andrew, "Computer-Aided Planning, Command and Control," from The Psychology of Management Decision, Editor, George Fisk, CWK Gleerup Publishers, Lund, Sweden, 1967, pp. 102-151.

Vitz, Paul C., "Preference for Different Amounts of Visual Complexity," Behavioral Science, 1965.

Weil, John W., "The Impact of Time-Sharing on Data Processing Management," Data Processing, Vol. IX, D.P.M.A., 1965, pp. 52-67.

Whisler, Thomas L., "The Impact of Advanced Technology on Managerial Decision-Making," paper delivered at the Conference on Employment Problems of Automation and Advanced Technology, International Institute for Labor Studies, July, 1964.

Whitehead, Thomas C., "Uses and Limitations of Systems Analysis," unpublished Ph.D. Dissertation, M.I.T., September, 1967.

Wilkins, P. R., "An Ergonomic Study of a Man-Machine Interface," Report No. 13, Computing Science Section, Centre for Computing and Automation, Imperial College, University of London, October, 1968.

Wilcox, Jarrod, "Market Participant Cognitive Maps for Individual Common Stocks," unpublished Ph.D. Dissertation, Alfred P. Sloan School of Management at M.I.T., January, 1970.

Will, H. J., "Definition of Management Information Systems," excerpted in "Information Systems in Management Science," Management Science, pp. B-391-95, February, 1968.

Wolek, Francis W., "The Complexity of Messages in Science and Engineering: An Influence on Patterns of Communication," Working Paper No. 110, Department of Industry, Wharton School of Finance and Industry, February, 1970.

Yntema, D. B., "The Software Problem," Lincoln Laboratory, Grp. 25, Report 1964-5, Massachusetts Institute of Technology, 4 September 1964.

Zannetos, Zenon S., "Toward a Functional Accounting System: Accounting Variances and Statistical Variance Analysis," Industrial Management Review, Spring, 1966a, pp. 71-83.

Zannetos, Zenon S., "Managerial Information Systems for Planning and Control," Working Paper No. 210-66, Alfred P. Sloan School of Management at M.I.T., August, 1966b.

Zannetos, Zenon S., "Management Information Systems and the Management Process: New Directions," Working Paper No. 243-67, Alfred P. Sloan School of Management at M.I.T., March, 1967.

BIOGRAPHICAL NOTE ON THE AUTHOR

Thomas P. Gerrity, Jr., was born on July 13, 1941, in Savannah, Georgia, the son of General and Mrs. Thomas P. Gerrity.

He attended secondary schools in Ohio, Texas, Virginia, and Oklahoma, and graduate first in his class from Midwest City High School, Oklahoma, in June 1959.

He entered the Massachusetts Institute of Technology as an Alfred P. Sloan Scholar in September 1959, where he studied Electrical Engineering, receiving the S.B. degree in June 1963. As an undergraduate, he was elected to Tau Beta Pi, and he received the Eastern Collegiate Athletic Conference Medal for 1963. He continued graduate study of Electrical Engineering at M.I.T. on a National Science Foundation Fellowship and received the S.M. degree in September 1964. He worked as a research assistant in M.I.T.'s Research Laboratory of Electronics during the summer of 1964 developing communication network simulation models on Project MAC's newly developed time-sharing computer.

In December 1963 Mr. Gerrity was awarded a Rhodes Scholarship for study at Oxford University. He studied Economics at Merton and Nuffield Colleges at Oxford under the tutelage of Professor Terence Gorman.

Mr. Gerrity entered the Alfred P. Sloan School of Management at M.I.T. on a National Defense Education Act Fellowship in September 1965 as a Ph.D. candidate in Management Information Systems. He began teaching at the Sloan School as a teaching assistant in February 1968 and was appointed an Instructor in July 1968. He teaches courses in management information systems, planning and control, and quantitative methods. He has accepted a position as an Assistant Professor in the Sloan School beginning June 1970.

Mr. Gerrity is a member of Tau Beta Pi, Sigma Xi, Eta Kappa Nu, the Association of Computing Machinery, the Institute of Management Sciences, the Operations Research Society of America, and the Society for Management Information Systems. He is a consultant to the RAND Corporation and has served as Lecturer to the U.S. Naval War College and to the U.S. Civil Service Commission Middle Management Institute.