DESIGN CRITERIA FOR A KNOWLEDGE-BASED ENGLISH LANGUAGE SYSTEM FOR MANAGEMENT: AN EXPERIMENTAL ANALYSIS

by

ASHOK MALHOTRA

B.Sc. (Honours) University of Delhi
(1962)

S.B. Massachusetts Institute of Technology
(1964)

S.M. Massachusetts Institute of Technology
(1966)

S.M. Massachusetts Institute of Technology
(1967)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF
PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF
TECHNOLOGY

February, 1975

Signature of Author ... ...
Alfred P. Sloan School of Management, January 15, 1975

Certified by ....
Thesis Supervisor

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

DESIGN CRITERIA FOR A KNOWLEDGE-BASED ENGLISH LANGUAGE SYSTEM FOR MANAGEMENT: AN EXPERIMENTAL ANALYSIS

Ashok Malhotra

ABSTRACT

Submitted to the Alfred P. Sloan School of Management

On January 15, 1974

In partial fulfillment of the requirements for

the degree of Doctor of Philosophy

This thesis investigates the utility and feasibility of a knowledge-based English language computer system to support management. An "ideal" system was designed to contain knowledge about a problem-domain and respond to questions and commands phrased in natural English. A prototype was implemented based upon the corporate data base of a hypothetical manufacturer of lead batteries.

To investigate actual system usage a "perfect" English language system was simulated with the assistance of the prototype. This was capable of responding to requests in free English typed in at a computer terminal. Twenty three subjects were asked to solve a problem involving the battery manufacturer using this system.

The experiment showed that managers were able to start quickly and work naturally with a system that could respond to requests phrased in English and could provide information about itself. Analysis of the words used in the sentences seems to indicate that a vocabulary of 1000 to 1500 words may be adequate for a domain-specific system. Some 78% of the sentences used by the managers fell into ten basic syntactic types and a moderately powerful parser would seem to be able to provide an adequate capability. To reach some understanding of the amount of knowledge required in a domain-specific system the subjects' requests were also analyzed for the knowledge that would be required to respond to them. We found that although the amount of knowledge required is large, it is feasible to incorporate it in a management-support system.

The problem-solving protocols obtained through the experiment were used to test a "frame oriented" paradigm of problem-solving which states that managers analyze problems by checking hierarchical lists of potentially contributing sub-problems. The data supports the paradigm with some evidence of exceptional behavior. This strengthens the generality of our results.

The final section of the thesis presents a design for an English language management-support system that is both technologically feasible and managerially useful.

Thesis Supervisor: William A. Martin

Title: Associate Professor of Electrical Engineering and Management

ACKNOWLEDGEMENTS

This thesis has benefitted from the advice, assistance and critisism of a large number of counsellors, colleagues and friends. They cannot all be named here but I want them to know that their contributions, large and small, are deeply appreciated.

The problem was initially conceived by Dr. John Weil and Dr. R. F. Clippinger of Honeywell. They, along with members of my thesis committee, helped narrow it down to reasonable proportions.

Prof. W. A. Martin and Prof. G. A. Gorry introduced me to the need for knowledge in intelligent systems. A large number of Prof. Martin's ideas are imbedded in this thesis; either philosophically or as realized in various pieces of software created by the Automatic Programming group at Project MAC. Dr. E.R. Banks wrote the morphological analysis program which was incorporated in the prototype system. Alexander Sunguroff and Rand B. Krumland were responsible for bringing up various versions of the OWL system that were used in the prototype system. Robert V. Baron wrote the Perfect English language simulation system. The Semantic Diffferential test was designed by Sudeep Anand.

Prof. Michael S. Scott Morton and Prof. Peter G. W. Keen were instrumental in setting the general direction of the thesis and in refining its arguments. They have been a continual source of ideas and inspiration.

It is difficult to overstress the contributions of my wife, Patricia, who suffered through the frustrations and irritations of the thesis process and provided the motivation and emotional support to keep going. She and the computer also deserve credit for the production of this document.

This work was supported by Honeywell Information Systems Inc., 200 Smith Street, Waltham, Mass., 02154

The prototype system incorporates software produced by the Automatic Programming group at Project MAC, M.I.T. which is supported by a grant from the Advanced Research Projects Agency of the Department of Defence.

This document was produced with the assistance of the PDP-10 computer acquired by Project MAC, M.I.T. with a grant from the Advanced Research Projects Agency of the Department of Defence.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

THE PROBLEM SETTING

1.1 INTRODUCTION

In the past few years it has become common to talk about how the computer will revolutionize management and how its powerful capabilities will be used to improve the process, and perhaps even the content, of human thought. While these prophecies cannot be challenged if projected into the dim mists of the indefinite future it must be admitted that the progress so far has been rather limited. It is a mattur of record that large numbers of computer installations are unsuccessful, by any definition of the term, and an even larger number of computer systems are failures in the sense that they do not meet the user's requirements and expectations and are soon abandoned.

The reasons for these failures are numerous and complex. Some systems fail because of inadequate understanding of the nature of the management processes they try to assist [5,18,27]. Others fail because they overlook organizational realities [24,25,49] and because they are supported by poor data gathering systems. Still others are technological failures. At first blush, technology would seem to be the least of the offenders. In fact, it is clear that developments in hardware and computer facilities are considerably ahead of useful applications. Nevertheless, technology has led to two kinds of failings in the development of computer-based systems for management. First, the spectacular advances in technology have seduced some practitioners into building systems that were technically sophisticated but managerially naive. Second, despite technological advances, it is still rather difficult for the manager to communicate with computers without extensive initiation into programming languages and data storage techniques. Not only must he encode his requests in complex unnatural jargon but he must specify how the information he needs has to be prepared from the data in the system. Computer systems are procedural in nature; they have to be told "how" to answer a request as well as "what" is wanted. If a computer system is to "understand" a manager's request and respond to it without solution procedures being specified, it must contain knowledge about the problem domain and the corporate context within which it exists. It must also contain knowledge about the processes that are appropriate in the problem-domain along with knowledge about typical requests that managers may make of it and the kinds of answers they expect. Minsky [46] discusses this need for knowledge in computer systems and argues that the power of a system depends on the quantity and quality of domain-related knowledge it possesses. His argument is central to our thesis and we shall refer to it many times.

A little reflection shows that a large amount of knowledge is required to make a truly responsive system. This, however, will make the system slow and expensive. As computer costs continue to decrease with respect to human costs, however, a point will be reached when an arbitrarily expensive interface system will become cost-effective. In fact, even today, a system that provides facilities that the manager needs and truly supports his day to day decision-making would have to be very inefficient indeed to be too expensive.

A Knowledge-based system will provide other benefits as well. It will allow the manager to make ad hoc requests that existing systems do not support. Mintzberg [48] found that management tasks are characterized by brevity, fragmentation and variety and, thus, ad hoc requests are very important to the management process. In addition, he will get faster response to his enquiries. This will enable him to explore complex problems, albeit those with a certain general structure for which he may not be able to produce models but knows the problem-solving processes they require, quickly and efficiently at a single sitting. On another level, the knowledge that the manager imparts to intermediaries to assist in responding to his requests will now be incorporated into the computer system. Consequently, the system will become more powerful and by acquiring a large amount of knowledge about the business it will be able to assist the manager in a natural and effective manner. It will also serve as an objective repository of corporate knowledge.

1.2 THE PROPOSAL

In this chapter and the next we propose a knowledge based system that attempts to alleviate some of the problems that make it difficult for managers to make effective use of computers by allowing them to ask questions and state commands in English. The remainder of the thesis will describe the detailed design of the system and explore it's utility and feasibility.

Simon [62] maintains that the central feature of management is decision-making. In fact he equates the two. We prefer the term problem-solving i.e. the development of solutions to alleviate undesirable situations. For the most part, the manager is concerned with problems that are generally familiar but are neither routine nor monumental. They are known to have solutions and the solution criteria are known. The manager's level of aspiration is not important and innovation is rarely necessary. The system will be designed for individual, rather than group, problem-solving but will provide some additional benefits by allowing a number of managers convenient, common access to sets of data they may not all know well. It will facilitate communication between line managers and provide a vehicle for cooperative decision-making.

We conducted an early, informal survey during which we asked managers how they would like an ideal system to behave, what facilities would they like it to exhibit and how would they use it. We followed this up with an informal experiment in which we introduced subjects to a managerial problem situation and asked them to try to reach an understanding of it sufficient to formulate a plan of action. We found that subjects needed to ask unstructured questions to obtain information during the problem-solving process. They asked questions about the state of the organization and the environment and they asked questions to test assumptions and to evaluate the effect of proposed policies. It seemed appropriate, therefore, to design a management support system as an English language system that would answer questions about the data base and about its contents and capabilities. The results of the early experiments indicated the general and specific design criteria for the management support system that is described in the following chapters.

If the questions are of a routine nature, or if they can be anticipated, it is possible to build formal information systems to provide answers for them. On the other hand, as we go up higher in the management hierarchy, the questions tend to become progressively less predictable and standard information systems become less and less useful.

A company routinely gathers data pertinent to the control of its business in a transactional database. The data may be encoded in files that are used by report generating programs written in COBOL or PL/1 or it may be stored in files created by a database language such as IMS, IDS or CODASYL COBOL extensions. In either case it is a formidable problem to read and comprehend the data structures. As a result two or more levels of personnel stand between the manager and database. There is a slow evolution of batch programs that take care of the bread and butter work of data capture, editing, file updating and report generation. The manager soon learns not to dream about what he could do if he could get a question answered immediately and waits patiently until the next report that contains the answer buried in it can be perused and excerpted for his purpose. Unusual requests require special programs to be written and it may take weeks or months to get an answer.

The questions the manager would like answered depend on the problem and the environment within which it appears.

To obtain answers from existing computer systems, however, he has to encode them in a complex jargon and specify in detail the operations to be performed on the data. To do this he has to learn a programming language and understand the intricacies of the organization of the relevant data files. Managers, however, do not like to learn computer languages and work with file structures. Even if they wanted to, most of them would not have the time to do so. While it can be argued that if managers need to use the computer badly enough, they will learn to program, this leaves out the borderline users and those who have not defined their problems and requirements clearly enough. Moreover, it imposes unnecessary demands on them. Thus, we recommend that computer systems should be accessible in English and their characteristics should allow the manager to use them as naturally as possible. Not only should they assist decision-making but they should also assist in the problem definition and formulation phase. The manager should be able to browse comfortably within a database, looking at data, computing functions, testing models until he feels he has reached an understanding of the problem situation.

We are therefore, attempting to describe a front-end or an interface to normal corporate data bases that will allow managers to use them in truly rewarding ways. The updating of the data bases and the veracity of their contents do not concern us in this thesis. We argue that such an interface will increase problem-solving effectiveness in situations that are structured but not routine. They will allow the manager to perform conveniently and quickly the data retrieval and data manipulation that he knows he needs. This will involve him more deeply in the problem situation and allow him to gain greater familiarity with the problem environment, look at it from a larger number of perspectives and investigate a larger range of solutions.

Since English is very rich and powerful it may seem too difficult to allow it as an input language. Managerial questions, however, do not span the English language. Analysis of typical questions makes us confident that it is possible to define a comfortable subset of English which will provide adequate fluency and completeness for the manager but will be limited enough to allow efficient comprehension by the system. In practice, it should not be calamitous if the question cannot be understood right away. If an intelligent response can be made, a dialog can be started which will lead to comprehension by the system.

The plan of this thesis is to start by defining the facilities required for an ideal system to support managers. This is based on our preliminary experiments and is described in Chapter 2. Later chapters discuss the validity of this ideal design in light of the results of a problem-solving experiment and discuss the feasibility of implementing such a system. The initial design is, thus, tested both technically and behaviorally. The final chapter presents a modified system design that is expected to be useful as well as realizable.

Chapter 3 describes a prototype system that was implemented in order to come to grips with some of the hard problems that need to be solved. Building the prototype system was the largest single part of the thesis and we learnt some important lessons in the process. These are also described in Chapter 3. The prototype system was incorporated into a simulator and used to conduct an experiment to test the validity of the system design. In this experiment, subjects were asked to solve a management problem using a system that, essentially, allowed them to ask for any information and processing they desired in free English. The details of the experiment and its primary results are summarized in Chapter 4. Appendix I describes the experimental materials used and Appendix II enumerates the requests made by the subjects to the system.

The experimental data is analyzed in greater detail in Chapters 5, 6, 7 and 8. Chapter 5 analyzes the words used by the subjects in framing their requests and discusses the size of the vocabulary necessary for a domain-specific English language management support system. Chapter 6 analyzes the subjects' requests in terms of their sentence types and discusses the capabilities of the parser that would be required to front such a system. Chapter 7 analyzes the various types of knowledge that would be required to support such a system. The knowledge required to respond to the requests made by the subjects is listed in Appendix III. A detailed paradigm of coming to grips with problem situations is described in Chapter 8. It is contrasted to existing theories of problem-solving and tested against the problem-solving protocols obtained from the subjects.

Chapter 9 presents the design of a management support system that will assist managers in a natural manner and is feasible to implement with current technology. It also touches on the problems of implementing such a system and suggests directions for future research.

1.3 EXISTING SYSTEMS

We shall describe two classes of computer-based systems in this section. The first class consists of query systems or "end user facilities" that attempt to simplify access to a data base. Some of these use an English-like command language. The existence of such systems indicates that the need we are addressing has been recognized although, perhaps influenced by available technology, in a limited way. Experience with such systems has been rather poor, however.

In the words of an insider "People play with them for some time and then abandon them". We shall try to analyze why this is so.

The second class of systems that we will describe are "knowledge-based" systems that have provided English language interfaces in some areas with some success. These systems incorporate knowledge pertaining to a particular area and use it to understand requests and solve problems. The success of these systems provides confidence that a knowledge-based system for management is now technically feasible.

1.3.1 End User Facilities

The managerial need to investigate the contents of a data base had led to the implementation of a number of systems called query languages or "end user facilities" such as TDMS [5], ICL [28] and CHARLIE [18]. These incorporate knowledge about the structure of the data base (some assume particular structures such as trees) and are able to respond to English-like queries. Typically, the allowable queries provide templates from which a user select certain options and fills slots with names of records and fields. A typical example, taken from I-D-S DATA QUERY [29] is:

> The first word of an Input Query must either be be DISPLAY, CREATE or PRINT followed by the Dictionary datanames or the record name the user wants to query....

DISPLAY directs the resultant output to the user's terminal for immediate display.
CREATE directs the resultant output to a permanent file.
PRINT directs the resultant output to an on-line printer.

To query records, the key word RECORD must precede the record name. This results in the display of all data fields in the record.

These systems really provide a set of instructions with fixed formats. They may look like English but there is little flexibility in the way requests have tc be formulated.Learning the formats is much like learning a programming language. The number of instruction formats is usually small. This makes the language easy to learn but limits its utility. Moreover, although the system takes care of the details of data retrieval, the user has to know the structure of the data base and use the names of records and fields correctly. Only rarely, as in the Management Data Query system [30], is there a facility that allows the user to ask questions about the structure of the data base. There is, of course, no knowledge about the contents of the data base and consequently the input language is rigid and unforgiving. Some systems allow the user to put together a number of statements to produce reports. These languages are procedural and although they are certainly easier to work with than COBOL or PL/1 they share the same characteristics and limitations except for the fact that they free the user from having to program around the details of the organization of the data files.

It is little wonder then, that these systems are not very successful. They recognize a genuine need but only go a little way towards filling it. The facilities offered are limited and address only a fraction of the user's needs.

1.3.2 Knowledge-Based Systems

W. A. Woods has developed a "transition network" grammar for natural languages and implemented a parser to analyze sentences in accordance with it. He used this parser in a system to answer questions about the chemical composition and other properties of the moon-rocks [72,73]. Woods' mainly syntactic parser is very powerful, although its efficacy is limited by its rudimentary semantic knowledge. His data bases are real, if simple, and he has been able to demonstrate the practicability of computer-based question-answering systems in real world situations. "The prototype (of the moon-rocks system) was run twice a day for three days . . . and during this time the lunar geologists . . . were invited to ask questions of the system. . . . During this demonstration 80% of the questions which were asked and that fell within the scope of the data base were parsed and interpreted correctly in exactly the form in which they were asked ..."

A question and command system implemented by Winograd [76] for the world of a set of children's blocks on a table-top demonstrates powerful capabilities of sentence analysis and comprehension. The simplicity of the world contributes in no small part to the impressiveness of the system, however. Further, most of the knowledge in the system is encoded as procedure and this makes it difficult to extend.

In developing a novel approach to computer-aided instruction Carbonnel and Collins developed a knowledge-based system that was capable of answering questions in addition to asking and evaluating them. A prototype system based around knowledge of the geography of South America is operational [12] with a limited subset of English. Carbonell's ideas have been developed further by Brown and Burton [7] and incorporated into a computer system that provides instruction in electronic circuit debugging. Burton has developed a very interesting, primarily semantic parser for this system and has had some encouraging preliminary results. He is planning a test with students from an electronics school shortly.

A number of other systems have been built (See Schank [59], Wilks [74], Kay[34]) that combine English language capability with intelligent behavior through the use of knowledge about the problem-domain. The success of these systems indicates that it may be possible to build knowledge-based English language systems to support managers effectively.

CHAPTER 2

SPECIFICATIONS FOR THE SYSTEM

2.1 GENERAL CHARACTERISTICS

Chapter 1 has motivated a system that allows a manager to work naturally and conveniently with a data base. In this chapter we attempts a more formal, a priori, definition of the system. It is a priori in the sense that it is based on our perception of managerial needs based on the early, informal experiment and not restricted by the available state of the art. In the succeeding chapters we make an analysis of managerial behavior and requirements and modify the system design in terms of it and on the basis of available technology. The final chapter presents this design.

To begin with, we feel that managers do not want to and should not need to learn a special language to use computers for day-to-day problem-solving. They should be able to converse naturally and comfortably with computers and the structure of the conversation should be dictated only by the nature of the problem. This implies that English be used as the input language. The advantage of English is that it will minimize learning time and provide tremendous power in terms of conversational capability. On the other hand, if the capabilities of the system are limited, and the user is restricted to only a few types of requests, English will seem cumbersome and repetitious. Thus, if the user is allowed to use English as an input language the system must be able to respond to requests that he can easily make in English. If the system offers only a few facilities and operators the user is better off with a simple, compact formal language.

The English language is very large and powerful and its very complexity makes it difficult to understand. There are many ways of phrasing a particular request and the system must be able to recognize them as isomorphic. Chapter 6 analyzes the English actually used by subjects in solving a management problem and finds that a very large number of their requests fall into a few basic sentence types. Thus, a reasonably complex parser supported by a knowledge-base should be able to provide adequate completeness and fluency within a particular problem environment.

Although the system does not require the user to learn a programming language it requires him to know its contents and capabilities. To be consistent with our philosophy, learning about the system should be made as easy and painless as possible. It is important, therefore, that the system be able to answer questions about itself and its capabilities.

For example:

"Can you calculate percentages?"

"What do you know about costs?"

"How is profit defined?"

"Do we have five products?"

These questions also help the manager to understand the situation he is dealing with and build appropriate conceptual and formal models for it.

A related issue in making the system natural to use is the protection of the naive user from system errors. It is well known that complex systems can never be completely debugged and thus occasional errors will continue to occur. These should however be trapped at some suitably high level and although the user may be informed of them he should not be required to take any corrective action. The system should merely say "Sorry, I cannot understand your request, please rephrase it" or "Sorry, I cannot perform the computations you wish, please ask for the them in an alternative way". In this way the user will learn to avoid certain types of requests and change his usage towards the requests he knows the system can answer but he will never be confronted by a cryptic message like : "ERROR 1273 ILLEGAL REFERENCE FROM 1623" and a dead system. Details of the errors and the requests that caused them should be logged into a special file that system programmers can look at from time to time and use to make suitable adjustments and improvements to the system.

The user cannot of course, be protected from catastrophic errors in the computer hardware or in the time-sharing monitor.

2.2 FACILITIES

One of the basic facilities that the manager requires is the ability to retrieve data from the data base using questions and imperatives. This, however, is not as simple as it seems for he often wishes to have the data cut in many different ways:

"What were sales to each customer in 1973?"

"Show me the sales from each plant in 1973."

Determining the parameters of data retrieval can involve fairly complex computations:

"Show me the sales for all plants that produced in excess of one million units or had budgets of over ten million dollars last year."

"What was the product mix at all plants whose profitability declined last year?"

Going beyond data retrieval, managers often want functions of data:

"What were the average sales to a customer in 1972?"

"What was the percentage increase in operating cost for each plant?"

Commonly used functions are sum, difference, increase, decrease, maximum, minimum, average, variance (in the accounting and in the statistical sense) distribution and percentage. Functions can, of course, be concatenated as in "maximum average" and "percentage increase".

While functions are useful for operating on available data, certain arithmetic functions of specific data come to acquire important positions in the user's model of the world. These are then graced with a name and known as models. Thus, "profit" is a model and may be defined as the difference between total revenue and total cost where total revenue and total cost may be contained in the data base or may, themselves, be models. Similarly, "contribution margin" may be defined as the difference between selling price and direct cost and "cost of goods sold" as the sum of overhead and production cost.

Such models are only the simplest of a class of models. Forecasting models that attempt to predict the future or the effect of some policy are often parameterized on some judgemental variable. Thus, these variables, as well as stored data, are necessary to evaluate the model. Still other types of models may be specialized by the user to his particular situation. BRANDAID [38], for example, provides a general model for forecasting the sales of a particular brand which may be specialized by the manager to take into account the charateristics of the product, the market and the promotional effort.

Forrester [20], Little [39] and Gorry [21] make a strong case for the need for model building facilities to allow a manager to come to grips with his environment and explore alternative action strategies. Our early discussions with managers confirm this. It seems clear that the ability to build, modify and use models is of prime importance in a system that attempts to support decision-making.

Closely related to models are "what-if" questions:

"What would profits be if sales increased to $60 million?"

"Suppose sales stayed the same and the price of product 4 was raised to increase its margin to $2.0, how would this affect profits?"

Clearly, in these cases, the user assumes that a model exists and desires its value given the parameters specified in the sentence. The model, however, needs to be quite sophisticated in certain cases. For example, if sales increase we can hardly expect all other costs to stay the same. Thus, the model should make "sensible" assumptions about the behavior of costs. These should be indicated to the user as part of the answer.

Some of the questions about the problem situation are phrased in such a way as to require either "yes" or "no" as an answer. Such questions are often used to test the user's model of the situation. For example:

"Are there any plants that were under budget for 1973?"

Yes-no questions may also be asked about the system to test if particular data or facilities exist.

"Can you calculate percentages?"

"Do you have any information on customer satisfaction?"

Identity questions, that start with "which" or "who", play an important role in the detailed isolation of problems.

"Which plants were over budget for 1973?"

"Who is our largest customer?"

Identity questions can also be asked about system capabilities but this is rare:

"Which items of data do you have for plant 1?"

The system should also contain a report generator so that retrieved data can be displayed in a form that the manager finds useful and convenient. The system should also be able to change the significance of numbers if desired i.e. display them in millions, or thousands or without fractional parts.

2.3 SYSTEM BUILDING FUNCTIONS

The ideal system should develop and grow with the manager and his job. In keeping with our general philosophy it should also adjust to the manager's idiosyncracies rather than the manager having to live with its peculiarities and limitations. A knowledge-based system implies continuous modification. Thus, the system should be able to accept changes and alterations in a natural manner as part of its normal functioning. Clearly, the system will not be able to accept basic structural changes. This is somewhat beyond the current state of the art but changes that stay within the general design should be acceptable.

Typically, the manager will want to add words to the system or declare words as equivalent. He may also want to add new items to the data base and their definitions and related knowledge to the knowledge base. Finally, he may want to add new functions and to define or modify models.

It is not necessary, however, that all of these types of changes be permitted on a conversational basis. This is very difficult in some cases and may not be important enough to justify the overhead. Building and modifying models, however, does seem very important as a particular model may be central to a problem situation. It is recommended, therefore, that of all the system building functions only model building be allowed at the console level by a naive user. Other kinds of additions and modifications can be carried out periodically by system maintenance people in response to a "wish list" maintained by the user. The system must, however, provide adequate facilities for system building functions.

2.4 SUMMARY

In summary, the capabilities of the ideal system can be classified as follows:

2.4.1 General Characteristics

1. The system will provide a conversational interface to a normal corporate data base containing transactional, and possibly other, data. It will supplement this with a knowledge base related to the contents of the data base and the capabilities of the system, the corporation and its environment.
2. The manager should be able to work with this data base and knowledge base in a convenient and powerful subset of the English language.
3. The system should be "bomb-proof" in that it should protect the user from the results of system errors and the need to respond to them or to take corrective action.

2.4.2 Facilities

The user should be able to:

1. Use questions and imperatives to retrieve data from the data base and properties of entities from the knowledge base and specify

the manner in which the data is to be displayed.

2. Ask for simple functions of stored data.
3. Build, modify and run different types of models. The input to these models could be stored data or supplied parameters.
4. Ask what-if questions based on underlying models.
5. Ask yes-no questions about the data stored in the data base and the properties and capabilities of the corporation and the system.
6. Ask identity questions related to entities belonging to the corporation and the system and their properties.

2.4.3 System Building Functions

Facilities should be provided to:

1. Add new words to the system.
2. Declare words as equivalent.
3. Add knowledge to the data base. An important subset of this would be knowledge related to data to be added to the system.

CHAPTER 3

THE PROTOTYPE SYSTEM

3.1 INTRODUCTION

To come to grips with the substantive problems involved in implementing a system along the lines described in Chapter 2, we decided to build a prototype based upon the activities of a hypothetical manufacturer of lead batteries called The Battery Company. Since the idea was to identify and explore the issues involved we designed the corporation to be simple but realistic. Details of The Battery Company and its organization are contained in the problem scenario included in Appendix I.

We then conducted some hand simulation experiments in which subjects were asked to solve a problem related to The Battery Company. This introduced them to a situation where profits were lower than last year despite the fact that sales had increased. In attempting to reach an understanding of this problem we found that the subjects asked for the facilities described in Chapter 2. The prototype system is designed to provide these facilities. In general, it attempts to support the solution of problems that are relevant and complex and possess a general structure. The system does not address itself to all management tasks but to the broad middle range of problems that are not routine but neither are they comletely unstructured.

Figure 3.1 is a schematic diagram of the prototype system. Functionally, the system can be divided into two parts, the parser and the processor. These two operating subsystems rely upon a knowledge base that contains a model of the world, a model of the scenario situation and knowledge of the contents of the data base. The data base contains operating data for The Battery Company.

The parser examines the input to the system and creates a parse for it. A parse is, in some sense, the meaning of the sentence, but more accurately, it is a canonical set of relations between semantically identified parts of the sentence encoded in a standard format. The processor uses the parse as input and attempts to generate an appropriate response to the input request. Understanding the sentence, therefore, takes place in a general sense within the parser and in a much more specific sense in the processor.

We shall say only a few words about the parser since it is the knowledge base and the processor that are central to this thesis. A number of good parsers have been written (Winograd [74], Woods [72], Burton [10]) and the more

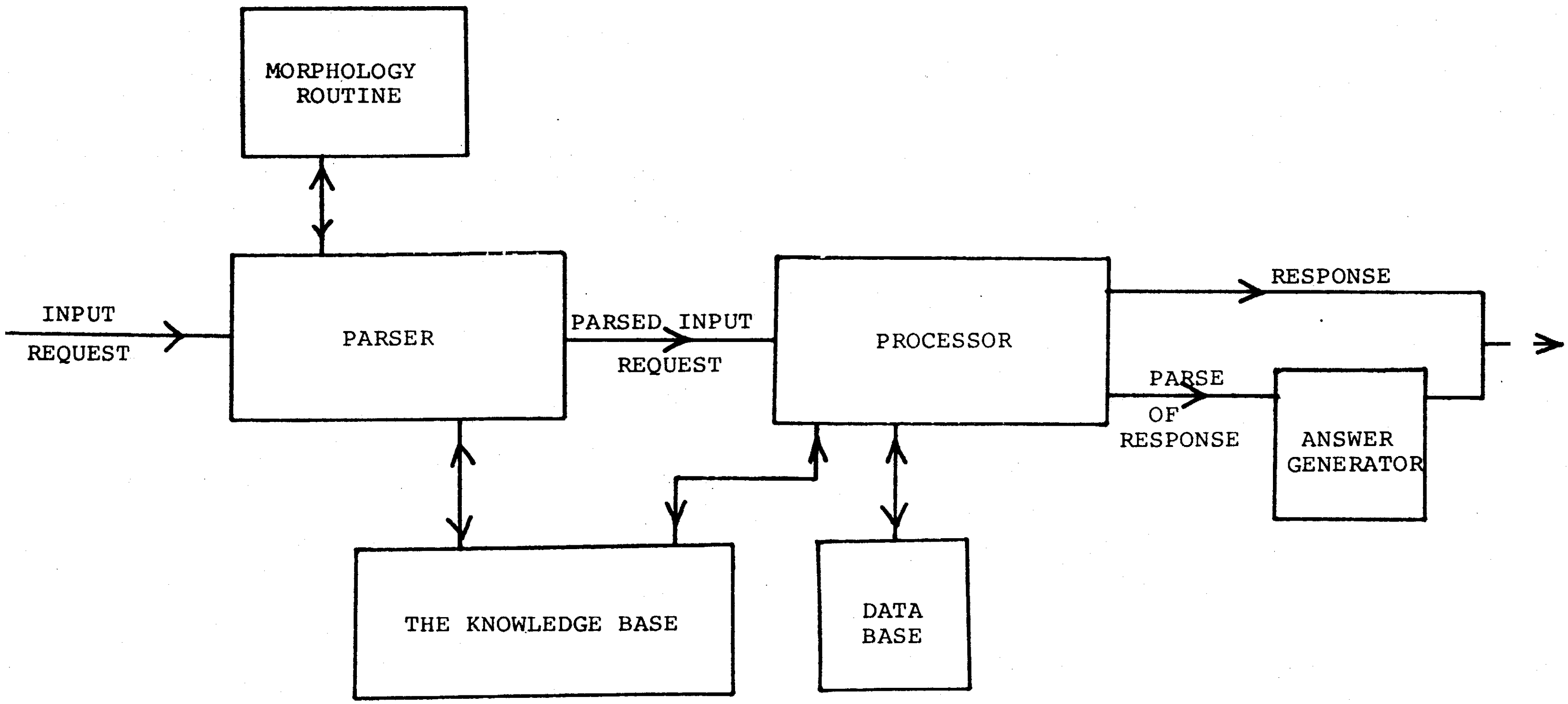


Figure 3.1 Schematic Diagram of the Prototype System

important problems today seem to be in the understanding of the sentence and in the creation of suitable responses.

The morphology routine acts as a preprocessor for the parser. It examines each word in the input request and checks if it is known to the knowledge base. Unknown words are analyzed to determine whether they belong to idioms or to general classes of words known to the system or are variants of known words. If a word cannot be recognized by the morphology routine a message is printed out indicating the offending word and the user is asked to retype his request. Chapter 5 contains a detailed analysis of the features required in a morphology routine.

Once the complete sentence is accepted, the "case-oriented" parser attempts to find the main verb and to arrange the noun phrases in the sentence as "cases" of the main verb. (See Fillmore [19] for the theory of case grammar and Celce-Murcia [13] for an early implementation of a case-oriented parser.) Initial prepositions that mark some of the noun phrases assist in this process, but the parser also uses knowledge about the verb which determines the cases it can take and the meanings of the nouns which determine the cases they can participate in. The parser implemented for the prototype system is an extremely simple parser that recognizes only ten basic types of sentences. It is described in Chapter 6.

If the sentence is ungrammatical or cannot be parsed, the parser prints out a message that asks the user to rephrase his request.

The following sections of this chapter describe details of the implementation of the prototype system and some of the technical problems that were encountered along with suggested solutions. They are aimed at the reader who is interested in building such a system rather than one who is concerned with its performance and capabilities.

3.2 THE KNOWLEDGE BASE

This section describes a language for encoding knowledge and the organization of knowledge for the prototype system. Later sub-sections discuss the impact of this organization on question-answering strategies and consider some alternative organizations.

The knowledge representation language (which may be the same as the programming language) must allow the representation of objects and their properties. It must also be able to keep property information of similar objects distinct. In addition, it is convenient if the language has facilities to query the data base and extract information from it. We use a version of a language called OWL [44] in which The Battery Company, known to the system as THE-BATTERY-COMPANY, can be represented as follows:

```
((IS (THE-BATTERY-COMPANY) (KIND CORPORATION))
 (MANUFACTURE (THE-BATTERY-COMPANY) (KIND PRODUCT))
 (SELL (THE-BATTERY-COMPANY) (KIND PRODUCT))
 (EMPLOY (THE-BATTERY-COMPANY) EMPLOYEE))
 (BUY (THE-BATTERY-COMPANY) MATERIAL))
```

This states that THE-BATTERY-COMPANY is a kind of corporation. It buys material, employs employees and manufactures and sells a kind of product. The properties of the product can be inserted into this definition or described separately as:

```
((IS (KIND PRODUCT) LEAD-BATTERY)
 (COUNT (TYPE (KIND PRODUCT) 5)))
```

This specifies that the product is five types of lead-batteries (lead-battery being used as a name). Similarly, the corporation could be described further as a sub-chapter 15 corporation in Massachusetts law but this may not be neccessary and (KIND CORPORATION) may be sufficient. OWL is, however, powerful enough to encode any kind of knowledge that we have felt the need for.

If the product definition is imbedded in the main definition, OWL will keep it distinct from other product definitions that may exist in the system.

OWL can also be used to represent events. The statement "We sent Sears a shipment yesterday." can be represented as:

```
((AGENT (SEND) WE)
```

(RECIPIENT (SEND) SEARS)

(OBJECT (SEND) (KIND SHIPMENT))

(DETERMINER (KIND SHIPMENT) INDEFINITE)

(TIME (SEND) YESTERDAY)

(TENSE (SEND) PAST))

The system contains knowledge about data, models, the corporation, the problem situation and the world encoded in OWL. The following sections discuss this knowledge and its use in responding to the various types of requests that may be received from the user.

3.3 WH-QUESTIONS

In general, wh-questions (questions starting with "what", "which" etc.) ask for the properties of objects or events. Object properties are questioned by "be" or "have" verbs (possibly modified by tense words) while event properties are generally questioned by the main verb of the event.

After the sentence is analyzed to determine the property questioned the property can be retrieved directly if the object or event exists in the knowledge base and if the property value is explicitly available. At this point we should recognize that the knowledge base may be organized by object or by event. If it is organized by object then each event will be represented as a property of one of its participants such as its agent. If the knowledge base is organized by event objects will be represented as properties of the existence, or "be", verb. A dual organization is also possible in which information is stored both ways. This allows a choice of retrieval paths and system knowledge can be used to select the path with the least expected search time.

Different question types require variants of the above general strategy but the basic tasks of finding the entity and finding or deducing the value of the questioned property remain more or less central in all cases.

If the information required has to be retrieved from the data base then the system has to use knowledge about the organization of the data base to create a program to perform the necessary retrieval. The problem of retrieval from variously structured files is not investigated in this thesis. Since real data bases often have complex structures this problem needs to be solved before such a system can be implemented. It seems to require, however, a developmental effort without presenting very complex issues for research.

Note that wh-questions requiring objective answers are restricted to the past and present tenses. It may be possible to answer "What was our profit last year?" from the data base but "What will be our profit next year?" requires a predictive model that embodies some level of subjective judgement. Thus, answering wh-questions in the future tense is completely different from answering them in the past or present tense and requires a search for suitable subjective models that may or may not exist.

3.4 AGGREGATE DATA

Some properties, such as costs and production figures are best kept as aggregates. The aggregation can be made in several ways and the request must specify exactly how the data are to be aggregated to produce the answer. "Sales", by itself, means little; it has to be qualified by the parameters, or keys, over which it is to be aggregated. These may be manufacturing facility, product, customer, salesman and time-period.

Each data item is associated with knowledge required to retrieve it from the data base. The prototype system assumes that all data is stored in arrays indexed by key values which characterize the individual piece of data and over which it may be aggregated. Each data item has, therefore, a list of key variables whose values must be specified before it can be retrieved. In addition, each data item has information as to where the key variable specifications will be contained in the sentence. As soon as the system recognizes that the request is for data retrieval it enters a special sub-system that uses this information to try to locate key information in the sentence. The keys for which the sentence does not provide information are filled by using defaults or with typical values. There is a powerful, general default that operates in English: IF A PROPERTY IS ATTRIBUTABLE TO SUB-ENTITIES THEN THE ABSENCE OF SUB-ENTITY SPECIFICATION IMPLIES SUMMATION OVER THE SET OF SUB-ENTITIES, I.E. IT IMPLIES A VALUE FOR THE ENTITY AS A WHOLE. In a sales specification, for example, if there is no mention of a product then the sum of sales for all products is required. This general default does not apply, however, if the product specification is omitted from requests for prices or unit costs. Clearly, such data cannot be aggregated over product and the appropriate response is to present the data for all products.

Typical values depend on the data in question. Time-periods, for example, can usually be assumed to be the last complete period if unspecified. Typical value information should also be contained in the knowledge about each data item. In the prototype system, however, it is organized centrally by classes of data items since this leads to a more compact organization.

If all the keys for the data item can be filled, the system generates a routine, using the key values, to retrieve and format the data.

In general, given a network or tree structured data base, the problem of defining key specifications and generating retrieval routines is somewhat more complex than for an array data base. There may be more than one way to get at a particular item of data and not only must the system know about the structure of the data base and the necessary and sufficient information to retrieve data from it but it also must know about relatively efficient ways of retrieving data. These problems are not investigated in this thesis. Some of the end user facilities that operate on complex data structures [4,5,26] incorporate (partial) solutions to this problem.

It can be argued that retrieval and aggregation of data are not the direct concerns of the question-answering system and it should restrict itself to extracting the necessary information from the sentence and passing it on to a data retrieval system that uses knowledge about the structure of the data files to perform the retrieval efficiently. This argument would segment the system into an "understander" and a "response generator". This seems to be a desirable decoupling.

Occasionally, a user will ask for aggregate data by a key it is not aggregated on. For example, he may ask for overhead costs by product from a data base that does not allocate overheads by product. Such a request usually indicates a misunderstanding on the part of the user of the world model used in the system. It is extremely important in such cases to point out the error to the user and, perhaps, provide additional information on how the data is stored. On no account should the erroneous specifications be ignored.

A management support system will, typically, know about a number of variously named costs such as "interest expense" and "product transportation cost". As each cost category must be stored under a unique name it becomes necessary to determine this name by operating upon the adjectives and classifiers of the word "cost" (or "expense") and what can be called its context case (contained in an immediately following preposition group starting with "of" or "for"). This is done in the prototype system by finding a type of data item and then checking whether a subset of it is named by one of the modifiers or the context. This is continued recursively till no further subsetting is possible. The problem of mapping the information contained in a noun group into a unique data name is one of the most difficult problems that have to be solved in a question-answering system. The prototype solution is somewhat naive. See also Chapter 7 and Appendix III.

Some sentences can be constructed so that the context serves to determine the unique name as well as provide a key for its retrieval. For example,

"Show me the cost for all products"

can be assumed to ask for the direct-manufacturing-cost as this is the only cost available that has "product" as a key. Overhead costs are not allocated by product for The Battery Company.

Since different aggregations may be required for different purposes an obvious strategy dictates that data be kept in a fairly disaggregated state and aggregated in response to the needs of the question. But maintaining data at one level of aggregation may not be sufficient. The aggregation of annual sales for five products from five plants to five customers stored by month requires 1500 probes of the data base! This can mean a long wait at the console for the user. Thus, for efficiency, higher levels of aggregation must be maintained as well. The system should be able to recognize the level of aggregation required to answer the question and in producing the answer should use the highest level of aggregation that is applicable and available. A tree structured data base can maintain disaggregated data at the leaf level and higher aggregations at the other nodes. This neccesitates more complex file maintenance procedures and, thus, these decisions must be predicated on an analysis of the extra cost required to maintain data at many different levels of aggregation as opposed to the cost of having to make aggregations whenever the question requires it.

3.5 MODELS

In management the use of the word model stems from its meaning as a replica. Typically, models are a set of relationships that establish the dependencies of target, or result, variables on independent, or decision, variables. The dependent variables are often figures of merit that measure the health of the enterprise or the success of some part of it and the model is used to assist in the decision-making process by predicting the effects of changes in the independent variables on the dependent variables.

Management models can take a variety of shapes and structures depending on the nature of the process they model and the nature of the decision-making they support. The prototype system considers only one class of models, those that can be represented by a set of mathematical operations on specified items of data and can, therefore, be encapsulated in a subroutine. This is the sense in which the word "model" is used in the rest of this thesis. Other types of models, in particular those that can be calibrated, specialized and modified by the manager, such as BRANDAID [38], are very important and very useful. They were excluded from the prototype system to limit its complexity.

One of the models incorporated into the prototype system is "profit"; defined as the difference between total revenue and total cost. Such models are referred to by name in the question. Input data is not specified unless it is exceptional and neither are the operations to be performed on it. This information has, therefore, to be available to the system as a property of the model. Key information relevant to the retrieval of input data must, however, be specified in the question or supplied by default.

A request for model evaluation causes a search for input names. After these have been located the data is retrieved using the key information and fed into the subroutine that calculates the value of the model. Although it is always possible to specify models as mathematical functions of data only, this is not always convenient. It is customary, therefore, to specify models that use the output of other models as input. For example, "profit" may be defined as the difference between "revenue" and "total cost" where "total cost" is itself the sum of "overhead cost" and "production cost". Thus, when the input retrieval routine encounters an input that is a model it calls the model evaluation routine to evaluate it. The structure of correctly specified models ensures that this recursion always terminates.

In addition to an input list and a subroutine, the knowledge about a model, like knowledge about data items, also contains information as to where key variable values for each key associated with each of its inputs can be found in the sentence. Each model must also have a definition that can be used to answer questions such as

"How is profit calculated?".

3.6 FUNCTIONS

Another facility that the system must provide is the ability to compute functions of data such as percentage, distribution or average. Unlike models, whose inputs are prespecified, functions can operate on different kinds of data and, therefore, the names of the inputs to the function have to be specified in the sentence along with key specifications for their retrieval.

A number of conventional devices are used to specify the data items on which the function is to operate. For example, if the percentage of a subset of an item is required, the question may name the data item and the subsetting characteristic:

"What percentage of plant capacity is utilized?"

Similarly, if a distribution is required, the data item may be named along with the key variable or subsetting name. The data will have to be aggregated over all the unspecified variables and retrieved for the allowable values of the specified characteristic.

In general, data names, key specifications and subsetting names will occur in different parts of the sentence and the determination of key specifications for each input must take into account the function to be executed. Consider,

"How much did sales increase over 1972?"

"How much did sales increase in 1972?"

To interpret and answer these two questions correctly the system must know that the two arguments to the function "increase" are the same data item with different key variable values. Typically, only one set of key variables is specified, the others being picked up by default. The defaults, however, depend on knowledge associated with the function.

Thus, analysis of questions that ask for functions of data has to depend heavily on knowledge about the function to determine the input data names and the key variable information. Once this has been done the data items can be retrieved and fed into a subroutine that evaluates the function.

Functions are usually specified by name in the question. An exception is "How many" which asks for the count of a set of objects or events. This is, however, an unusual function since it does not operate on data.

3.7 DEFINITIONS AND INFORMATION ABOUT SYSTEM CONTENTS

Consider the questions:

"What information do you have about product cost ?"

"What do you know about product cost ?"

After product cost is located directly and checked to be a subset of a possession of "you" (information) or located by a search from among the possessions of "you" what does the system reply?

The problem with "know" is similar, although the object known may be called knowledge and the search avoided. In fact, what is required is, in some sense, the definition of product cost. This is implicit in the question but must be made explicit for the system.

In fact, these examples demonstrate a more general convention in English: IF A "WHAT" QUESTION IS ABOUT A SPECIFIC OBJECT THEN ITS MOST IMPORTANT CHARACTERISTIC IS REQUIRED. For example, "What were sales in 1972?" asks for the value of sales: value being the principal characteristic of data. "What is this dog?" asks for the dominant characteristic of this particular dog: perhaps its breed. "WHAT" QUESTIONS ABOUT GENERAL CLASSES OF OBJECTS REQUIRE EITHER THEIR DOMINANT CHARACTERISTIC OR THEIR GENERAL CLASSIFICATION. Thus, "What is a dog?" can be answered by saying that it is an animal or that it is "man's best friend"! Carbonnel and Collins [12] represent these "most important characteristics" or, as they call them, "super concepts" for each noun explicitly in the data base.

3.8 YES-NO QUESTIONS

Yes-no questions, if restricted to the present and the past tense inquire about the truth of stated propositions. Propositions may concern the properties of stated objects and the identity of actors and other particulars about events. Yes-no questions in the future tense involve predictions and judgements about the future and were, therefore, excluded from the scope of the prototype system.

In a data base organized by nouns, a yes-no question inquiring about the truth of stated properties of nouns can be answered by selecting the appropriate nouns and matching their properties against the stated properties. A recursive property matching routine is very useful for this. It must be remembered, however, that the properties mentioned in the question must be checked against those in the data base and not vice versa as the data base may contain properties other than those questioned.

Event matching is somewhat more complex in such a data base. Since events are stored as properties of the agent or the possessor these nouns have to be located in the question and in the data base and their properties searched for the event. Once this is accomplished, or a set of possible events selected, their properties have to be matched against those in the question.

The alternate organization by event, or verb, requires a decision as to whether objects that participate in an event should be stored under the event and/or under the existence ("be") verb. Direct retrieval for identity questions and yes-no questions requires both facilities:

"Who robbed the bank?"	property of "rob"
"Did Jack rob the bank?"	property of "Jack"

If properties are only stored by event an indirect retrieval is required for the second question much like the indirect retrieval required for events in a noun-oriented data base.

There seems to be a duality between objects and events in that the world can be described either as properties of objects or as properties of events. Depending on the question and the contents of the data base, however, there may be more objects than events or vice versa. Thus, retrieval via one path or the other will result in less searching. If the length of data files is less important than the amount of searching required in the retrieval process, then a dual organization in which properties are stored under both events and objects and the retrieval path selected, with the help of the knowledge base, to minimize search will yield the best results. In fact, OWL is designed for the dual organization and automatically stores properties under both the object and the event.

Winograd discusses the difficulty in creating negative answers to yes-no questions.

> If we ask "Does the block support three pyramids?" and in fact it supports four, what is the correct answer? The system could ask for clarification between "at least three" and exactly three", then answer "yes" or "no". But it is more efficient and helpful to answer "FOUR OF THEM", leaving the speaker to interpret his own question.

The system should, then, attempt to provide as much information about the true state of affairs as possible rather than responding with a mere "No". The prototype system is not so sophisticated but this seems to be a desirable feature to incorporate into it.

3.9 IDENTITY QUESTIONS

Questions that start with "who" or "which" are quite different from other wh-questions and are, in fact, more like yes-no questions. What is required as an answer to these questions is the identity of the object that satisfies the properties stated in the question and the process of answering can be likened to answering a set of yes-no questions on a set of candidate objects that are capable of satisfying the conditions in general. Questions starting with "who" ask for the identity of animate objects while questions starting with "which" can ask for either animate or inamimate identities.

Answering routines for identity questions, therefore, start with the selection of a set of candidate objects. The generic name for this set is invariably specified as the main object noun in identity questions with "be" verbs. The candidate set is the set of all objects that are "a kind of" the generic name. These questions specify the properties of the required object rather than the event it participated in.

"Who" questions that ask the identity of the agent of an event do not give any direct clues to the candidate set of objects. In many data bases the set of animate objects is fairly small and so a search through all of them is not very time-consuming.

Once the candidate set is established, the selection process is much like performing a yes-no question matching on each event except that the result of each matching is the identity of the candidate or "no". The final response can be created from the set of identities that have been returned by the individual yes-no questions. If this set is the null set then the appropriate answer is "None of them" for "which" questions and "No one" for "who" questions.

3.10 IMPERATIVES

Besides asking questions, the user of the system can request services from the system by using commands. This makes for more natural dialog. Commands ask for action and as the actions possible by the system are limited, so are the types of commands that can serve as meaningful input. The services provided by the system are limited to data retrieval, model evaluation, computation of functions of data or model values and the provision of information about itself. Typical commands to the system, therefore, are as follow:

"Show me the sales to Sears for 1973."

"Display the names of customers with outstandings of over $5000."

These questions seem to ask for data retrieval.

"Compute the profit for '73."

"Calculate the return on investment last year."

These questions seem to ask for the value of functions of data or the execution of models. The distinction is specious, however. The structure of the data base will determine what can be retrieved and what has to be computed. The user will, in general, be unaware of the structure of the data base his choice of verbs should not be considered significant.

Other verbs have semantic significance and require special routines to process data and generate answers.

"Compare the distribution of sales to Gulf and Sears by unit."

"Contrast the sales for each quarter of this year and last year."

"Sort the customers by outstandings."

3.11 CONCLUSION

The prototype system served its purpose by alerting us to a number of important problems such as the naming problem, the aggregate data problem, the need for definitions and others mentioned in the preceding sections. Good solutions will need to be developed for these problems before a successful English language management-support system can be implemented.

The most important benefit of the prototype system was, however, that it made possible the "perfect" English language question-answering system described in the following chapter. This simulation system allowed subjects to request the system for any kind of data retrieval and manipulation they required in free English. The problem-solving protocols of these subjects were then analyzed to determine the real needs of managers upon which the design of a truly useful management-support system could be based.

The experiment and a summary of the results is described in Chapter 4. Detailed analysis of the requests and the problem-solving protocols is contained in Chapters 5, 6, 7 and 8

CHAPTER 4

THE EXPERIMENT

4.1 MOTIVATION

Earlier chapters have described the specifications that, from our early experiments, would be desirable for an English language system to assist management. It must be realized, however, that although the specifications can be agreed upon in general they have to be limited and refined before we can start to design a practical system. For example, although an English language capability may be felt to be desirable, complete English as an input language is neither feasible nor practical. Similarly, the knowledge contained in the system has to be limited as do the functions provided for operating on the data base.

It is extremely difficult to develop successful design specifications in an arbitrary manner. In fact many management-support systems seem to have come to grief because they tend to embody the designer's ideas as to what managers should need [26,27,28,29,49]. We feel very strongly that systems for managers should be designed based on some analysis of how managers actually behave. We decided, therefore, to conduct an experiment in which subjects would be asked to solve a realistic problem using a simulated "perfect" English language management-support system that, essentially was capable of answering any question and carrying out any command that the subject could state in English.

Problem-solving protocols on the perfect system were analyzed to determine the vocabulary and the syntax used by managers and the knowledge and the facilities that were required to respond to their requests. These were analyzed further to determine whether the creation of such a system is technically feasible and, if so, to develop the design specifications for a real-world system.

Analysis of the protocols also helps answer another, more basic, question --- whether English language systems are useful in managerial problem-solving and whether managers will be comfortable with them and will be able to use them with a minimum of instruction.

4.2 THE EXPERIMENTAL SETTING

The Perfect English Language System was constructed as a set of programs that allowed two consoles, logged into the Automatic Programming Group's PDP-10 time-sharing system, to communicate with each other. (See Note 1.) Experimental subjects were able to type into a hard copy console and "send" their requests to the system by typing in a special character. Their requests appeared on the experimenter's console and he created responses to them by invoking functions from a LISP system. The Perfect System was, of course, limited to the data contained in the data base and to the knowledge of the experimenter and the facilities available to him. In addition, the answers to some questions, to which the experimenter had to create answers, took considerably longer than answers to questions that could be easily invoked.

Since the subject received only the output of the LISP function, not the invocation, the system seemed to him to be responding directly to his requests. In fact, surprising as it may seem, few subjects realized that the experimenter was creating the responses until they were told so after the experiment. Until this secret was revealed, many subjects were extremely impressed by the range of capabilities displayed by the system. Thus, the Perfect System could be said to be a success as the subjects behaved as if it were an ideal English language question-answering system.

Subjects for the experiment were selected to have some acquaintance with management concepts and vocabulary. They represented, in fact, a wide range of experience from ten years of line management to engineers taking their first courses in management. Approximately half of them were experienced managers and half were students. Some of the students had, however, a few years of work experience with some management responsibility.

The experiment consisted of three parts: filling in a background questionnaire and completing a semantic differential test that attempted to elicit their attitudes towards computers and information systems, three trials of the Bruner concept formation test [9] and the analysis of a managerial problem using the Perfect English Language System. Details of the experiment are described in Appendix I.

The first two parts of the experiment were preliminary and were intended to elicit personality traits of the subjects that may be useful in explaining some of their behavior. The problem analysis was the heart of the experiment and consumed the bulk of the time required for it.

The experimental problem puts the subject in the shoes of the president of a lead battery manufacturing company who receives the operating results for the last year and is surprised to find that although sales increased by 20% profits decreased by 1%. The subject is asked to analyze the situation and, if possible, recommend a course of action. The problem is structured but not routine and it was designed to be typical of the sort of problem managers face regularly in the course of their duties.

4.3 ANSWERING THE QUESTIONS

The Perfect System was built around the prototype question-answering system for the lead battery manufacturer described in Chapter 3. This was the LISP system that was loaded in for the experiment. On receiving a request, the experimenter had three basic options. He could translate it into a sentence that could be accepted by the prototype system, he could type in the answer that would appear on the subject's console directly or he could invoke a general LISP function whose output would appear on the subjects console. In practice, the last option was rarely used. Some 60% of the requests could be answered by the prototype system and were translated into it. The other 40% of the requests were answered directly.

4.4 THE RESULTS

The requests made by the subjects to the Perfect English language system and their problem-solving protocols were analyzed to yield five major classes of results:

1. Subjects' behavioral reactions to the system and the setting.
2. Vocabulary requirements for an English language management-support system.
3. Parsing requirements for an English language management-support system.

4. Knowledge requirements for a domain-specific English language management-support system.
5. Analysis of conceptual structures and strategies used by subjects to solve the problem and the implications of this analysis for the design of English language management-support systems.

These results are summarized in the following five sub-sections. Details of the last four types of analysis are contained in the four subsequent chapters.

4.4.1 Behavioral Reactions to the System and the Setting

In every case, the subject read the problem scenario and the instructions to use the system and went readily to work. In a few cases there was some hesitation due to the unfamiliar technology and the mechanics of editing and sending a request had to be explained. This was done quite rapidly, however, and the subject was at work within a few minutes after reading the documentation.

Some subjects started out with very simple requests for single items of data. Gradually, as they gained confidence in the system, they asked more demanding questions requesting blocks of data, invoking models and performing complex computations on the data. They would then go on to ask "what if" questions, define models and ask for underlying causes. Thus, the subjects explored the capabilities of the system while solving the problem. They did this by gradually increasing the complexity of the questions and by asking direct questions about system capability:

"Can you format reports?"

"Do you perform mathematical computations?".

One of the initial, fuzzy notions behind the design of the system was that managers "should be able to talk to it like a human being". And indeed, after a few questions, the subjects began to treat it like one. Their English was informal, much closer to the spoken language than prose, they were cavalier about sentence forms and style and tended to ignore inessentials like punctuation. Having to type in the requests and the knowledge that they were interacting with a computer system did seem to have some effect on the input, however. As the next sub-section explains, the majority of their sentences were short and simple and for the most part they were coherent and unambiguous.

In two cases subjects brought with them their experience of psuedo English language systems they had worked with. This tended to make their conversation stilted and patterned. One of them defined models that gave the percentage increase over the previous year. When asked, after the experiment, why she didn't ask questions of the kind "How much did --- increase over last year?", she replied in wonderment "I didn't think you could do that"!

A few subjects expressed their impatience at having to precede all requests for data with "what is" by leaving it out. Almost invariably they were poor typists. Other subjects attempted to set specifications to be obeyed over the next set of questions. Yet another form of economizing on input was to define models and then merely specify parameters in subsequent questions. This seems to indicate that some people feel that English is a little cumbersome for routine data retrieval and that it may be desirable to build a command language on top of the English system for routine inquiries.

In summary, all the subjects took quite naturally to the system and were able to work with it without significant problems. After the experiment, most of them commented that the system "would be very useful if it could be implemented". A high-level manager for a retail food chain felt it would be very useful to train store managers and also to manage individual profit centers like a bakery. While reactions were overwhelmingly positive there were a few comments on the limitations of the system: it could not provide tabular reports, would not line up decimal points in lists of number and could not format numbers or change their significance (express them in thousands or millions). Some users wanted to ask for sets of related data such as the contents of the balance sheet, the profit and loss statement or the sources and uses of funds. The system was not designed to provide these.

4.4.2 The Vocabulary

The words used in the requests obtained from the subjects are analyzed in Chapter 5. This section summarizes the results. The 496 sentences used by the subjects were formed from 358 basic words. The probability of encountering new words in subsequent sentences decreases rapidly with the number of sentences. (See Figure 5.3) This seems to suggest that a vocabulary of 1000 to 1500 words may be sufficient for an English language system to support a particular business application. This opinion is based on a small sample of requests, however. A larger sample would allow a more confident prediction.

Chapter 5 also develops the requirements for a morphological analysis program that attempts to associate each word of the input sentence with the appropriate pieces of knowledge contained in the system. If a word cannot be found in the dictionary, the program checks to see if it is a member of a class of words it knows about. If so, it creates the required knowledge from general knowledge about the class and the special characteristics of the word. In this way it can recognize inflected forms of known words (ran and running from run), noun idioms (cost of goods sold), numerical nominalizations (products 1, 2 and 3), contractions (what's, I'm) and abbreviations (%, $, info, OH, mfg). In certain situations it should allow the use of unknown words. For example, if the user says "Let p-cost be the sum of production and overhead cost for each plant" the system must accept "p-cost" as the name of the new model. The program must also be able to make allowances for common misspellings and for run in words such as "whatis".

If a word is not contained in the dictionary and cannot be recognized as a member of a known class then the system does not have the knowledge to process it and the morphological analysis program prints out an appropriate message to the user.

4.4.3 The Parsing Requirements

A basic parser that analyzes sentences syntactically to match ten known sentence types and uses semantic knowledge to put together a canonical representation of the sentence can parse 78% of the sentences obtained from the users. A more complete description of the parser is given in section 6.2.

The sentences that could not be parsed by this basic parser included a large number of additional sentence types and syntactic conventions. Section 6.4 contains an analysis of the features that should be added to the basic parser to allow most of the unparsed sentences to be parsed.

The frequency of sentences classified by sentence type seems to follow the well known Pareto distribution [53]. This often appears in analyzing occurrence frequencies by class; be they sales by item or the amount of damage by fire. Typically, a few classes account for a large percentages of the occurrences. Thus, the majority of the sentences fall into a few types, but, if the tail is to be covered, a large number of sentence types must be added. There will, however, always be a few sentences that the system will not be able to parse. Looked at another way, there will come a point after which the increase in parser complexity will not be justified by the number of additional sentences that will be parsed. The system designer will have to decide where this point lies based on his particular situation. Chapter 6 provides some pointers.

Since some sentences will not parse, no matter how powerful the parser, the system should respond politely to them by asking the user to rephrase his question and providing as much information as possible as to why the request did not get a normal response.

4.4.4 The Knowledge Base

Chapter 7 and Appendix III analyze the knowledge required for a domain-specific English language management-support system. We find that a large variety of different kinds of knowledge is required. The system needs to have knowledge about data, about models, and about functions of data and model values. For each of these it requires a number of different kinds of information. The system also needs to know the properties of entities and deduction rules that can be used to relate questioned properties to stored properties. In addition to knowledge about the problem situation and the environment the system also needs to know how to respond to different types of requests including those that are ambiguous, incorrect or can not be analyzed by the system.

Every manager brings with him a conceptual model of the problem world. This includes assumptions about how information about the corporation is collected and stored. His model may, however, be at variance with the model incorporated in the system. This can sometimes create serious difficulties. It is important, therefore, to analyze the requests that cannot be answered by the system and respond to those that arise from differences in world models by providing information about the model incorporated in the system. Users who recognize these discrepancies in world models often wish to define quantities that are important to their world model and bring the world model in the system closer to their own.

Appendix III contains the knowledge required to respond to the requests obtained from the subjects. It is described in English and although it is very varied it covers only about 30 pages of text.

4.4.5 Analysis of Problem Solving Behavior

The "frame" paradigm of coming to grips with problem situations that is described in Chapter 8 was supported by the data obtained from the protocols.

The paradigm states that managers attempt to understand a gross problem by checking lists of sub-problems that may contribute to it. This is a hierarchical process that stops with the isolation of a set of sub-problems that can either be alleviated directly by decisions or for which more information or expertise is required.

In cases where the top level of potential sub-problems does not yield an existing problem the manager follows one or more of three strategies: he goes back over his frame and rechecks each sub-problem, perhaps using different data and different functions to test if it exists, he attempts to generate additional potential sub-problems, or he reverts to basic concepts and uses these to attack the problem.

Validation of the frame paradigm indicates that managers use a few basic processes to reach an understanding of problem situations. The system design we have presented in this thesis provides the capabilities for supporting these basic processes and is, therefore, suitable for a wide range of management problems. Moreover, if frame structures are found to be stable over a wide range of managers or if they can be identified for a set of managers then the design of the system can be based upon them.

CHAPTER 5

THE VOCABULARY

5.1 INTRODUCTION

Each word that is included in the system vocabulary is, in some sense, a concept that the system must "know". Although the knowledge required about different kinds of words varies considerably, a vocabulary analysis provides an initial estimate of the amount of knowledge required in the system. In some cases the system contains knowledge about classes of words such as inflected forms of a word (run, ran, running) and numbers, rather than for each individual word. This reduces the amount of knowledge required but creates the necessity of recognizing words as members of known sets and specializing the general knowledge about the set to the individual word. This is the responsibility of the morphological, or morphemic, analysis program that looks at the input sentence as a string of words and ties each of them into the relevant pieces of knowledge. The size of the vocabulary, therefore, depends on the power of this program. The following sub-section describes the features desirable in a morphemic analysis program. Subsequent sub-sections discuss the words obtained from the sentences used by the subjects as analyzed by a moderately powerful morphological analysis program. This is used to try to answer the general question of how large a vocabulary is required by a domain-specific system if it is to provide adequate fluency and power.

5.2 THE MORPHOLOGICAL ANALYSIS PROGRAM

The morphological analysis program acts as a preprocessor to the parser. Its name derives from the fact that its original purpose was to analyze inflected forms of dictionary words, formed by adding common suffixes, and provide properties for them. A modern morphological analysis program analyzes a variety of different types of words and provides properties for them, if these are not explicitly available, before they reach the parser. The program looks up each word of the input sentence in a dictionary containing its syntactic properties and perhaps a pointer to relevant entries in the knowledge base. Words not found in the dictionary may be genuinely unknown (or misspelt) or special forms that the system recognizes in various ways. The morphemic analysis program should process unknown words by testing if they have certain commonly encountered endings. If so, they can be treated as known words and the basic or root word and certain rules based on the ending used to attribute properties to it.

Thus, "lesser" becomes the comparative form of the adjective "less" and "smoker" the one-who-does noun form of the verb "smoke". Suffixes can, of course, be concatenated and must, therefore, be stripped recursively to allow for forms like "smokers" and "outwardly"

Winograd [75] and Allen [2,3] describe morphemic analysis programs. Allen's is considerably more powerful.

In addition to basic suffix analysis a good morphemic analysis program must be able to analyze common prefixes. This must be done in conjunction with suffix analysis to allow forms such as "unkindness" and "non-budgeted".

In many instances, a particular string of words, over a period of usage, comes to designate a single concept. These noun-idioms must be recognized as single entities by the morphemic analysis program and correspondence established with appropriate knowledge elements. Familiar management examples are "product mix" and "cost of goods sold"

Abbreviations and equivalences are another type of problem that the morphemic analysis program must be able to handle. Common usage allows "$" and "%" but people like to use more specialized abbreviations in specific problem contexts; "OH" for "overhead, "info" for "information" and "mfg" for "manufacturing" are familiar management abbreviations. There are two approaches to processing abbreviations. The simpler is to translate only known abbreviations. The more complex is to supplement this with a set of general rules to recognize abbreviations of known words. The simpler approach may be quite adequate for an initial system however. The complex approach still needs to be proven. The system must, however, provide some facility for recognizing abbreviations and replacing them by the parent word.

A more complex translation issue arises in the case of equivalences which may be used to reduce the number of entries in the knowledge base. A system designer may decide, for instance, that "expense" and "cost" are equivalent and that the morphemic analysis program should replace "expense" by "cost" on sight. Such equivalences must be declared with care, however. For most purposes "expense" may be equivalent to "cost" but "expensing" an item (instead of capitalizing it) is very different from "costing" it! Also, if the system knows that "doing" certain verbs gives rise to certain nouns, it will have to know that spending gives rise to "cost" and not the more natural "expense". Words display various facets of meaning in different contexts and it is difficult to find words that have equivalent meanings over many context domains.

Another responsibility of the morphological analysis program is to deal with numbers. Numbers usually act as quantifiers but they can also function as nouns in special ways. The morhological analysis program must therefore be able to distinguish the use of numbers in forms like "$ 3.2", "August 15, 1943", and "the 400 series" and produce appropriate representations in each case. The system must possess special purpose knowledge to be able to process each instance. The program should also be able to recognize that "three thousand and five" means the same thing as "3005" and should convert the words into the numbers. Similarly, it should be able to accept fractions, such as 2/3, and convert them into their decimal equivalents. It should also be able to accept ordinal numbers such as "fourth" and "22nd".

Numbers following nouns generally specialize them to entities identified by the numbers, such as product 3 and plant 4. It seems convenient to let the morphemic analysis program make the specialization, whether by general rule or by special knowledge. The numbers can, however, be conjoined in common forms such as "products 1, 2 and 3" and "plants 1 through 5". If the morphemic analysis routine has to make the specialization it must also be able to analyze the conjoined forms. This seems to move it into the domain of the parser, however, and may be undesirable. Thus, we recommend that the morphemic analysis program treat numbers following nouns as special kinds of nouns. The parser can analyze these nouns by its general mechanisms and make the necessary specializations.

In summary, the initial analysis of a user request consists of looking up each word in the dictionary and if it is not contained there, subjecting it to a series of tests, such as those described above, to determine whether it is a variant of a known word or if it belongs to a class of words, like cardinal numbers, that the system knows about. If these tests also fail then the word is not known to the system and it does not have the knowledge to process it successfully. The program prints out "------- is an unknown word" and the user is asked to rephrase his request. In a system that begins processing the request after it is completed, the entire request must be retyped. A system that processes incoming requests character by character can, however, complain as soon as an unknown word arrives, keeping the earlier message in a buffer. This allows the user to substitute another word and continue the request if he so desires.

It seems desirable to allow the user to define new words conversationally as part of his interaction. The problem is, however, that, except in special cases, each word in a knowledge-based system has a significant amount of knowledge attached to it. Without this knowledge it cannot be processed correctly, if at all. Since it is too much to expect the user to be able to supply this knowledge (in the proper format) it seems best not to allow words to be defined on the fly.

There are exceptions to this, however. Some users want to define models and in doing so provide names for the defined quantities. These names will be unknown to the system, but their definition and the fact that it is a name for a derived quantity is all that needs to be known about them. These user defined names should be accepted by the system. As the system may find it difficult to determine, without involved processing, whether such an unknown word is a name, some convention may be adopted to indicate them. They may, for instance, be enclosed within quotation marks.

It is also desirable to incorporate some simple form of spelling correction and recognition of run in words such as "whatis" into the morphemic analysis program. Such a facility is described by Teitelman [63] and incorporated into the SOPHIE system [10]. This provides some allowance for poor spelling and for typing errors and contributes to the general philosophy of making the system forgiving and easier to use.

Finally, in a system that processes the input character by character the user's request is typed directly into the morphemic analysis program. It must provide, therefore, facilities for erasing the last letter, word or line and for displaying the edited request. In a system that processes the input by line, the user types into an input utility and it provides the editing facilities.

5.2 THE RESULTS

The 496 sentences obtained from users were found to be formed from 358 basic words. These are presented in Figures 5.1 and 5.2, sorted by frequency of occurrence and alphabetically. This analysis was carried out using a moderately powerful morphological analysis program called MORPH. (See Note 2.) This program is capable of performing all the functions described in the previous section except for prefix analysis. It recognizes only known abbreviations and its abbreviation mechanism can be used to declare equivalences if desired. The comments preceding Figure 5.1 list the suffixes analyzed by MORPH and the extra words that needed to be included because of its limitations.

The entry "cardinal number" in the two lists stands, generically, for all instances in which a number appeared in the sentences.

Comparison of the list of words ordered by frequency with a similar list compiled from a million words from various sources at Brown University [36] shows interesting similarities and differences. The first word in our list is "cardinal number" which occurs most often as "1972" or "1973"; clearly a feature of the nature of the sentences and their purpose. The Brown list compiles numbers individually. The highest ranking, close to two hundredth place, belongs to "2". The next word on our list is "?" which is not recognized in the Brown list. The following word is "the" which is also the first word on the Brown list. The next word on our list is "for" while on the Brown list it is "of". It is interesting that both are prepositions. The usage "for 1973" is clearly responsible for "for" being so high on our list. The next word on our list is "be". This includes all variants of "be" while the Brown list has "is" in seventh place, "be" in sixteenth place and other variants in lower rankings. The sixth word on our list is "cost". This clearly displays the nature and purpose of the sentences from which the list was compiled. The first noun in the Brown list does not occur till past the seventieth position and is "man".

Thus, the lists have similar features, somewhat distorted by the idiosyncracies of compilation. Our list comes from a considerably smaller sample with a very strong content bias. This accounts for its special characteristics.

Figure 5.3 is a plot of the number of new words encountered in every twenty additional sentences. The sentences were analyzed in chronological order of occurrence. It seems clear from the figure that the probability of encountering new words becomes very low after about 300 sentences. A vocabulary of 1000 to 1500 words would, therefore, seem to be sufficient to support an English language question-answering system for a specific management application. This is considerably lower than the off-the-cuff estimates that have been mentioned by various people but depends, of course, on the size of the domain to be supported. Such a small vocabulary requirement is not really surprising if we consider that Basic English [52] which was being promoted as a universal language had only 800 words.

Further analysis shows that some users tend to use special words. (Others prefer special types of sentences.) Thus, a few users tend to bring particular words with them and these appear in their initial few sentences. This contributes to the bumpy behavior of the plot presented in Figure 5.3. Plots of new words encountered in every 50 or more sentences, that sum across more than one user are considerably smoother.

After a number of users, however, the number of new words introduced by a user becomes smaller and smaller. Also, as the vocabulary grows the inability to understand a word becomes less serious as the likelihood of having adequate synonymns increases.

An attempt to analyze the 358 words into general words that need to be known to any English understanding system and words specific to the problem domain yields approximately 90 words that are business related. This statement should be qualified by the fact that a large number of words such as "unit" and "increase" are difficult to classify. Further analysis into words that are general to an English language system that supports business problems and words that are particular to the given data base yields only about 50 words that are specific to the data base. Here again, a large number of words such as "incur" and "gross" are difficult to classify. These results should not be regarded as precise, therefore.

A number of people have remarked on the need for a "user profile" that would adapt the system to the style and requirements of the individual user. In so far as this consists of special names for data and functions it can be handled quite simply by means of an individual synonymn list. Special functions and facilities would, however, require a major additional effort.

COMMENTS ON FIGURES 5.1 AND 5.2:

1. Twenty four user defined words have been removed.
2. The numbers following each word indicate the number of occurrences in 496 sentences.
3. The number of occurrences do not take into account multiple meanings of words.
4. The morphological analysis program is capable of analyzing the following suffixes:

CONTRACTIONS	"'LL", "'VE", "N'T", "'D"
PLURALS	"S", "ES", "IES"
POSSESSIVES	"'S", "'"
VERB ENDINGS	"EN", "ED", "ING", "MENT", "TION", "SION"
ADJECTIVE ENDINGS	"ER", "EST", "LY", "NESS"
NOUN ENDINGS	"ER", "LY"

4. Total vocabulary is 362 words. the following words are included due to limitations in the morphological analysis program:

COMPARATIVE, RELATIVE	DOES NOT ANALYZE "IVE" ENDINGS
ADDITIONAL, MATHEMATICAL	DOES NOT ANALYZE "AL" ENDINGS
PROFITABLE	DOES NOT ANALYZE "ABLE" ENDINGS
PROFITABILITY	DOES NOT ANALYZE "ITY" ENDINGS"
PERCENTAGE	DOES NOT ANALYZE "AGE" ENDINGS
INSTALLATION, SATISFACTION	EXCEPTIONAL CASES OF "TION" ENDINGS
UNCHANGE, NON-MANUFACTURING	DOES NOT ANALYZE PREFIXES

FIGURE 5.1
LISTING OF WORDS ENCOUNTERED IN THE PROTOCOLS BY FREQUENCY:

(CARDINAL-NUMBER 464.)	(? 309.)	(THE 297.)
(FOR 248.)	(BE 226.)	(COST 193.)
(. 185.)	(AND 180.)	(WHAT 177.)
(IN 151.)	(OF 141.)	(PRODUCT 138.)
(PLANT 123.)	(EACH 114.)	(OVERHEAD 88.)
(SALES 88.)	(YEAR 77.)	(BY 69.)
(GIVE 60.)	(HAVE 52.)	(DO 49.)
(UNIT 49.)	(PRICE 47.)	(YOU 47.)
(LIST 45.)	(PROFIT 43.)	(TOTAL 41.)
(ME 38.)	(TO 38.)	(BUDGET 33.)
(PRODUCTION 32.)	(A 31.)	(LAST 31.)
(MARGIN 31.)	(PERCENT 31.)	(ACTUAL 30.)
(INCREASE 30.)	(PERCENTAGE 29.)	(PER 27.)
(ALL 25.)	(PRINT 25.)	(CHANGE 24.)
(DISPLAY 24.)	(FROM 24.)	(HOW 21.)
(DIRECT 20.)	(MANUFACTURING 19.)	(AT 18.)
(AVERAGE 18.)	(FIGURE 18.)	(SELL 18.)

(BETWEEN 16.)
(OVER 16.)
(ANY 14.)
(PLEASE 13.)
(TRANSPORTATION 12.)
(DATA 11.)
(ITEM 11.)
(TYPE 11.)
(WHICH 10.)
(LET 9.)
(INCLUDE 8.)
(THIS 8.)
(IF 7.)
(TABLE 7.)
(YES 7.)
(LARGE 6.)
(SAME 6.)
(VOLUME 6.)
(COMPANY 5.)
(GO 5.)
(SALARY 5.)
(ABOUT 4.)
(COMPARE 4.)
(DISTRIBUTION 4.)
(INCUR 4.)
(MANAGEMENT 4.)
(QUESTION 4.)
(TERM 4.)
(AMOUNT 3.)
(CALL 3.)
(FOLLOW 3.)
(IT 3.)
(MAJOR 3.)
(NO 3.)
(PREVIOUS 3.)
(RECORD 3.)
(SUM 3.)
(TIME 3.)
(! 2.)
(ALSO 2.)
(BREAK 2.)
(END 2.)
(GET 2.)
(HANDLE 2.)
(ITEMIZE 2.)
(MANY 2.)
(MOST 2.)
(OK 2.)
(OUTSTANDING 2.)
(PROPORTIONAL 2.)
(CONTRIBUTION 16.)
(I 15.)
(PAST 14.)
(DIFFERENCE 12.)
(CAN 11.)
(DOLLAR 11.)
(MUCH 11.)
(THAN 10.)
(AS 9.)
(BATTERY 8.)
(NUMBER 8.)
(CANCEL 7.)
(INFORMATION 7.)
(WHY 7.)
(ACCOUNT 6.)
(MORE 6.)
(SOLVE 6.)
(WITH 6.)
(DIVIDE 5.)
(KNOW 5.)
(UP 5.)
(ANSWER 4.)
(DECREASE 4.)
(GROSS 4.)
(INVENTORY 4.)
(MODEL 4.)
(QUOTATION 4.)
(THEY 4.)
(ASSOCIATED 3.)
(COMPONENT 3.)
(FREIGHT 3.)
(LIKE 3.)
(MATERIAL 3.)
(OPERATIONS 3.)
(PROBLEM 3.)
(REGARDING 3.)
(THAT 3.)
(VARIOUS 3.)
(ABSOLUTE 2.)
(ALTER 2.)
(CALCULATE 2.)
(EXCEED 2.)
(GREAT 2.)
(HIGH 2.)
(LESS 2.)
(MILLION 2.)
(NET 2.)
(OR 2.)
(OVERRUN 2.)
(REFLECTED 2.)
(ON 16.)
(OPERATING 15.)
(EXPENSE 13.)
(REVENUE 12.)
(CUSTOMER 11.)
(INTEREST 11.)
(RATIO 11.)
(THERE 10.)
(DEFINE 9.)
(BREAKDOWN 8.)
(PRODUCE 8.)
(COMPUTE 8.)
(PROFITABILITY 7.)
(WILL 7.)
(DISCOUNT 6.)
(REQUEST 6.)
(VARIANCE 6.)
(YOUR 6.)
(DURING 5.)
(ONE 5.)
(VERSUS 5.)
(COMPARATIVE 4.)
(DEPRECIATION 4.)
(GROWTH 4.)
(LABOR 4.)
(NOT 4.)
(RATE 4.)
(VALUE 4.)
(BOTH 3.)
(EXPECT 3.)
(GOOD 3.)
(LOAN 3.)
(MY 3.)
(OTHER 3.)
(QUANTITY 3.)
(RELATIVE 3.)
(THEIR 3.)
(WE 3.)
(ADDITIONAL 2.)
(BASIS 2.)
(DOWN 2.)
(FIXED 2.)
(GUIDELINE 2.)
(INSTALLATION 2.)
(LOW 2.)
(MINUS 2.)
(NORMAL 2.)
(OUR 2.)
(PERFORM 2.)
(RELATE 2.)

(REMEMBER 2.)	(RESULT 2.)	(RETAIN 2.)
(SHALL 2.)	(SHIPPING 2.)	(SPECIFIC 2.)
(SPECIFY 2.)	(SPEND 2.)	(STANDARD 2.)
(SUBTRACT 2.)	(SUCH 2.)	(SUPPOSE 2.)
(THEM 2.)	(THEN 2.)	(THESE 2.)
(THINK 2.)	(THOSE 2.)	(THROUGH 2.)
(VARIABLE 2.)	(WHERE 2.)	(WHOSE 2.)
(WIDE 2.)	(ACCORDING 1.)	(AGAIN 1.)
(ALLOCATE 1.)	(ALLOW 1.)	(ASSUME 1.)
(ATTRIBUTABLE 1.)	(AVAILABLE 1.)	(BACK 1.)
(BEFORE 1.)	(BELIEVE 1.)	(BORROW 1.)
(BRANCH 1.)	(BUT 1.)	(CARRIER 1.)
(CATEGORY 1.)	(CENTER 1.)	(CHARGE 1.)
(COMPETE 1.)	(CONCERN 1.)	(CONGRATULATIONS 1.)
(CONSTITUENT 1.)	(CONSTRAINT 1.)	(CONTAIN 1.)
(COUNT 1.)	(DEMAND 1.)	(DETAIL 1.)
(DEVIATION 1.)	(DIFFERENT 1.)	(DISREGARD 1.)
(DISTRICT 1.)	(DIVIDEND 1.)	(DUE 1.)
(EARLY 1.)	(ENTERTAIN 1.)	(EQUATION 1.)
(EQUIPMENT 1.)	(EVEN 1.)	(EVERY 1.)
(EXPENDITURE 1.)	(EXPRESS 1.)	(FACE 1.)
(FACTOR 1.)	(FALL 1.)	(FAR 1.)
(FEE 1.)	(FILL 1.)	(FIND 1.)
(FINISHED 1.)	(FORCE 1.)	(FORECAST 1.)
(FORMAT 1.)	(FORMULA 1.)	(FUNCTION 1.)
(FURTHER 1.)	(FUTURE 1.)	(HELLO 1.)
(HELP 1.)	(IE 1.)	(INCOME 1.)
(INDEPENDENT 1.)	(INDIRECT 1.)	(INFLATE 1.)
(INPUT 1.)	(INTENT 1.)	(INTERVIEW 1.)
(JOB 1.)	(JUST 1.)	(LEVEL 1.)
(LOCATION 1.)	(LONG 1.)	(MAKE 1.)
(MATHEMATICAL 1.)	(MAXIMIZE 1.)	(MEAN 1.)
(MEASURE 1.)	(METHOD 1.)	(NEXT 1.)
(NON-MANUFACTURING 1.)	(OFF 1.)	(OPERATE 1.)
(OUT 1.)	(OUTSIDE 1.)	(OVERALL 1.)
(PAID 1.)	(PART 1.)	(PASS 1.)
(PAY 1.)	(PERIOD 1.)	(PICTURE 1.)
(PIECE 1.)	(PLAN 1.)	(PROFITABLE 1.)
(PROPORTION 1.)	(PURCHASE 1.)	(QUOTE 1.)
(RAISE 1.)	(RECEIVED 1.)	(RECENT 1.)
(REGION 1.)	(REMAIN 1.)	(REPEAT 1.)
(REPEAT 1.)	(REPORTS 1.)	(RESPOND 1.)
(RID 1.)	(SATISFACTION 1.)	(SEPARATE 1.)
(SERVICE 1.)	(SHARE 1.)	(SHOW 1.)
(SIGNIFICANT 1.)	(SINGLE 1.)	(STRUCTURE 1.)
(STUDY 1.)	(SUBJECT 1.)	(SUBSTITUTE 1.)
(SUGGEST 1.)	(SUPPLY 1.)	(TAX 1.)
(TELL 1.)	(THANK 1.)	(THOUGH 1.)
(THOUSAND 1.)	(TOO 1.)	(TOUGH 1.)
(TRUCKER 1.)	(TURNOVER 1.)	(UNDERSTAND 1.)

(UNTIL 1.)	(US 1.)	(VARY 1.)
(WANT 1.)	(WAY 1.)	(WHO 1.)
(WITHIN 1.)		

FIGURE 5.2
ALPHABETICAL LISTING OF WORDS ENCOUNTERED IN THE PROTOCOLS:

(! 2.)	(. 185.)	(? 309.)
(A 31.)	(ABOUT 4.)	(ABSOLUTE 2.)
(ACCORDING 1.)	(ACCOUNT 6.)	(ACTUAL 30.)
(ADDITIONAL 2.)	(AGAIN 1.)	(ALL 25.)
(ALLOCATE 1.)	(ALLOW 1.)	(ALSO 2.)
(ALTER 2.)	(AMOUNT 3.)	(AND 180.)
(ANSWER 4.)	(ANY 14.)	(AS 9.)
(ASSOCIATED 3.)	(ASSUME 1.)	(AT 18.)
(ATTRIBUTABLE 1.)	(AVAILABLE 1.)	(AVERAGE 18.)
(BACK 1.)	(BASIS 2.)	(BATTERY 8.)
(BE 226.)	(BEFORE 1.)	(BELIEVE 1.)
(BETWEEN 16.)	(BORROW 1.)	(BOTH 3.)
(BRANCH 1.)	(BREAK 2.)	(BREAKDOWN 8.)
(BUDGET 33.)	(BUT 1.)	(BY 69.)
(CALCULATE 2.)	(CALL 3.)	(CAN 11.)
(CANCEL 7.)	(CARDINAL-NUMBER 464.)	(CARRIER 1.)
(CATEGORY 1.)	(CENTER 1.)	(CHANGE 24.)
(CHARGE 1.)	(COMPANY 5.)	(COMPARATIVE 4.)
(COMPARE 4.)	(COMPETE 1.)	(COMPONENT 3.)
(COMPUTE 8.)	(CONCERN 1.)	(CONGRATULATIONS 1.)
(CONSTITUENT 1.)	(CONSTRAINT 1.)	(CONTAIN 1.)
(CONTRIBUTION 16.)	(COST 193.)	(COUNT 1.)
(CUSTOMER 11.)	(DATA 11.)	(DECREASE 4.)
(DEFINE 9.)	(DEMAND 1.)	(DEPRECIATION 4.)
(DETAIL 1.)	(DEVIATION 1.)	(DIFFERENCE 12.)
(DIFFERENT 1.)	(DIRECT 20.)	(DISCOUNT 6.)
(DISPLAY 24.)	(DISREGARD 1.)	(DISTRIBUTION 4.)
(DISTRICT 1.)	(DIVIDE 5.)	(DIVIDEND 1.)
(DO 49.)	(DOLLAR 11.)	(DOWN 2.)
(DUE 1.)	(DURING 5.)	(EACH 114.)
(EARLY 1.)	(END 2.)	(ENTERTAIN 1.)
(EQUATION 1.)	(EQUIPMENT 1.)	(EVEN 1.)
(EVERY 1.)	(EXCEED 2.)	(EXPECT 3.)
(EXPENDITURE 1.)	(EXPENSE 13.)	(EXPRESS 1.)
(FACE 1.)	(FACTOR 1.)	(FALL 1.)
(FAR 1.)	(FEE 1.)	(FIGURE 18.)
(FILL 1.)	(FIND 1.)	(FINISHED 1.)
(FIXED 2.)	(FOLLOW 3.)	(FOR 248.)
(FORCE 1.)	(FORECAST 1.)	(FORMAT 1.)
(FORMULA 1.)	(FREIGHT 3.)	(FROM 24.)
(FUNCTION 1.)	(FURTHER 1.)	(FUTURE 1.)
(GET 2.)	(GIVE 60.)	(GO 5.)
(GOOD 3.)	(GREAT 2.)	(GROSS 4.)
(GROWTH 4.)	(GUIDELINE 2.)	(HANDLE 2.)
(HAVE 52.)	(HELLO 1.)	(HELP 1.)

(HIGH 2.)	(HOW 21.)	(I 15.)
(IE 1.)	(IF 7.)	(IN 151.)
(INCLUDE 8.)	(INCOME 1.)	(INCREASE 30.)
(INCUR 4.)	(INDEPENDENT 1.)	(INDIRECT 1.)
(INFLATE 1.)	(INFORMATION 7.)	(INPUT 1.)
(INSTALLATION 2.)	(INTENT 1.)	(INTEREST 11.)
(INTERVIEW 1.)	(INVENTORY 4.)	(IT 3.)
(ITEM 11.)	(ITEMIZE 2.)	(JOB 1.)
(JUST 1.)	(KNOW 5.)	(LABOR 4.)
(LARGE 6.)	(LAST 31.)	(LESS 2.)
(LET 9.)	(LEVEL 1.)	(LIKE 3.)
(LIST 45.)	(LOAN 3.)	(LOCATION 1.)
(LONG 1.)	(LOW 2.)	(MAJOR 3.)
(MAKE 1.)	(MANAGEMENT 4.)	(MANUFACTURING 19.)
(MANY 2.)	(MARGIN 31.)	(MATERIAL 3.)
(MATHEMATICAL 1.)	(MAXIMIZE 1.)	(ME 38.)
(MEAN 1.)	(MEASURE 1.)	(METHOD 1.)
(MILLION 2.)	(MINUS 2.)	(MODEL 4.)
(MORE 6.)	(MOST 2.)	(MUCH 11.)
(MY 3.)	(NET 2.)	(NEXT 1.)
(NO 3.)	(NON-MANUFACTURING 1.)	(NORMAL 2.)
(NOT 4.)	(NUMBER 8.)	(OF 141.)
(OFF 1.)	(OK 2.)	(ON 16.)
(ONE 5.)	(OPERATE 1.)	(OPERATING 15.)
(OPERATIONS 3.)	(OR 2.)	(OTHER 3.)
(OUR 2.)	(OUT 1.)	(OUTSIDE 1.)
(OUTSTANDING 2.)	(OVER 16.)	(OVERALL 1.)
(PAID 1.)	(PART 1.)	(PASS 1.)
(PAST 14.)	(PAY 1.)	(PER 27.)
(PERCENT 31.)	(PERCENTAGE 29.)	(PERFORM 2.)
(PERIOD 1.)	(PICTURE 1.)	(PIECE 1.)
(PLAN 1.)	(PLANT 123.)	(PLEASE 13.)
(PREVIOUS 3.)	(PRICE 47.)	(PRINT 25.)
(PROBLEM 3.)	(PRODUCE 8.)	(PRODUCT 138.)
(PRODUCTION 32.)	(PROFIT 43.)	(PROFITABILITY 7.)
(PROFITABLE 1.)	(PROPORTION 1.)	(PROPORTIONAL 2.)
(PURCHASE 1.)	(QUANTITY 3.)	(QUESTION 4.)
(QUOTATION 4.)	(QUOTE 1.)	(RAISE 1.)
(RATE 4.)	(RATIO 11.)	(RECEIVED 1.)
(RECENT 1.)	(RECORD 3.)	(REFLECTED 2.)
(REGARDING 3.)	(REGION 1.)	(RELATE 2.)
(RELATIVE 3.)	(REMAIN 1.)	(REMEMBER 2.)
(REPEAT 1.)	(REPEAT 1.)	(REPORTS 1.)
(REQUEST 6.)	(RESPOND 1.)	(RESULT 2.)
(RETAIN 2.)	(REVENUE 12.)	(RID 1.)
(SALARY 5.)	(SALES 88.)	(SAME 6.)
(SATISFACTION 1.)	(SELL 18.)	(SEPARATE 1.)
(SERVICE 1.)	(SHALL 2.)	(SHARE 1.)
(SHIPPING 2.)	(SHOW 1.)	(SIGNIFICANT 1.)
(SINGLE 1.)	(SOLVE 6.)	(SPECIFIC 2.)

(SPECIFY 2.) (SPEND 2.) (STANDARD 2.)
(STRUCTURE 1.) (STUDY 1.) (SUBJECT 1.)
(SUBSTITUTE 1.) (SUBTRACT 2.) (SUCH 2.)
(SUGGEST 1.) (SUM 3.) (SUPPLY 1.)
(SUPPOSE 2.) (TABLE 7.) (TAX 1.)
(TELL 1.) (TERM 4.) (THAN 10.)
(THANK 1.) (THAT 3.) (THE 297.)
(THEIR 3.) (THEM 2.) (THEN 2.)
(THERE 10.) (THESE 2.) (THEY 4.)
(THINK 2.) (THIS 8.) (THOSE 2.)
(THOUGH 1.) (THOUSAND 1.) (THROUGH 2.)
(TIME 3.) (TO 38.) (TOO 1.)
(TOTAL 41.) (TOUGH 1.) (TRANSPORTATION 12.)
(TRUCKER 1.) (TURNOVER 1.) (TYPE 11.)
(UNDERSTAND 1.) (UNIT 49.) (UNTIL 1.)
(UP 5.) (US 1.) (VALUE 4.)
(VARIABLE 2.) (VARIANCE 6.) (VARIOUS 3.)
(VARY 1.) (VERSUS 5.) (VOLUME 6.)
(WANT 1.) (WE 3.) (WHAT 177.)
(WHERE 2.) (WHICH 10.) (WHO 1.)
(WHOSE 2.) (WHY 7.) (WIDE 2.)
(WILL 7.) (WITH 6.) (WITHIN 1.)
(YEAR 77.) (YES 7.) (YOU 47.)
(YOUR 6.)

```
SENTENCES   I  NEW WORDS
20          I    5   10   15   20   25   30   35   40   45   50   55   60*
I           I
40          I                                        *
I           I
60          I                           *
I           I
80          I                    *
I           I
100         I                   *
I           I
120         I                     *
I           I
140         I               *
I           I
160         I     *
I           I
180         I                                *
I           I
200         I *
I           I
220         I                              *
I           I
240         I             *
I           I
260         I *
I           I
280         I      *
I           I
300         I      *
I           I
320         I   *
I           I
340         I              *
I           I
360         I   *
I           I
380         I
I           I
400         I                         *
I           I
420         I   *
I           I
440         I               *
I           I
460         I    *
I           I
480         I           *
I           I
```

Figure 5.3

Occurrence Of New Words

In Every Additional

Twenty Sentences

CHAPTER 6

PARSING THE SENTENCES

6.1 THE PARSER AND THE PROCESSOR

After each word has been recognized by the morphological analysis program and provided with properties and pointers to entries in the knowledge-base, the sentence is analyzed further by the parser and its component parts encoded into a canonical representation. Parsing used to be regarded as the syntactic analysis of the sentence into subject, verb, object, etc., or any other suitable canonical form, based on the syntactic typology of the constituent words. It became clear, however, that this was inadequate in many ways and that the semantic properties or "meanings" of words had to be utilized if the parsing process was to be increased in power and scope and useful parses were to be produced for practical purposes. (See Katz [33], Minsky [46], Winograd [76].)

The "meaning" of a word is, however, difficult to define since there are many facets of meaning, each useful in its own contexts and for its own purposes. If the parsing process uses some kinds of meaning, then the process of responding to the user's request or, more accurately, to the parse of the user's request uses other layers of meaning. The parser may, for example, use the fact that "1973" refers to a time duration. Further analysis may require the information that "1973" is a particular year for which the system has data. Thus, "understanding" a request is carried out partly in the parser and partly in the processer and consists of using different types of knowledge to map individual sentences into more general models of concepts and situations.

Parsing performs a part of this mapping, typically using "general purpose" knowledge, while the processor further specializes the mapping using "special purpose" knowledge The division, however, seems to be arbitrary, and it is not difficult to think of systems with a powerful processor and simple parser behaving similarly to those with a simple parser and a complex processor.

For the purpose of this analysis we will consider a somewhat simple parser that analyzes each sentence in terms of the main verb and the noun groups that participate in various ways in the action. A noun group, or noun phrase, as diagrammed in Figure 6.3, consists of a main, terminal noun along with preceding determiners, ordinals, numbers, adjectives and classifiers. The verb "sell", for example, takes an "agent" who initiates the action, a "recipient" who receives the goods, an "object" that is sold and an "exchange" that is paid for it. In addition to these cases that are special to "sell", verbs can take general cases indicating time, location, manner, frequency, etc. Not all of these cases occur explicitly in every sentence that relates to selling but every sentence that relates to selling can be analyzed in terms of them. For example:

"John sold Sue a book yesterday."

can be analyzed by setting "John", "Sue", "a book" and "yesterday" as the agent, recipient, object and time cases of the verb "sell". Other information such as the tense, number and person of the verb and the classification of the sentence completes the parse.

A number of researchers have advocated case systems for English. (See Fillmore [19], Chafe [14], Martin [42,43].) At this stage, however, none of them seem to possess particularly significant advantages over the others and a universal system has not been agreed upon. Martin's system, which uses a larger number of cases than the others, seems to have some advantage in terms of precision and computability. We shall follow his terminology.

Knowledge of word meanings is used by the parser primarily to decide on the cases to be assigned to the various noun groups. It will, for example, create a "manner" case from the prepositional phrase "as a percentage". The name of this case indicates that it has some bearing on the way the result has to be expressed. The parser does not know, however, what "percentage" really means, that it has two arguments, or how it is to be calculated. These pieces of information are filled in by the processer as it finds and acts on the cases assigned by the parser.

6.2 THE PARSER

By all indications, the most powerful parser operational today is the one developed by Woods. This has been improved over a period of years and he has used it to implement a system that answers questions about the chemical composition and other properties of the moon-rocks [72]. In testing the latter system Woods found that "80% of the questions that were asked and that fell within the scope of the data base were parsed and interpreted correctly in exactly the form in which they were asked".

Earlier parsers [54,77] attempted to use Chomsky's ideas about transformational grammars [16] and tried to "unwind" the transformations to reproduce the deep structure of the sentence. It soon became apparent that this was very difficult as transformational grammars are biased toward the process of generating sentences rather than interpreting them. Woods [70] discusses the "combinatorial explosion" inherent in the inverse transformational process. Recognizing the limitations of the transformational model, three parsers by Thorne, Bradley and Dewar [64], Bobrow and Fraser [6] and Woods [70] appeared in 1968 and 1969 based on augmented transition network grammars. These systems model the parser as a transition network much like the finite state recognizer used for regular languages in automata theory. The finite-state model is extended, however, by allowing networks to make recursive calls to other networks and to themselves. The condition on moving from one state to another is extended from examining merely the next word to a call to a network, such as noun phrase. This can act as a recognizer and, if it is successful, use up more than one word of the input. The process then continues from the following input word and in the new state. In addition to this feature, the structure of the network is controlled by a set of registers whose contents can be modified during the recognition process. Thus, words at the start of a sentence can influence the recognition process for the rest of the sentence.

Augmented transition networks have the power of Turing machines and can handle any type of grammar that could be parsed by machine. Their advantages and limitations lie in the manner they are able to represent actual language operations. Woods' parser encodes syntactic structures in the transition network and takes decisions based on the syntactic properties of words in the sentences. It pursues only the most likely parse at any stage, keeping track of all possible parse paths by means of likelihood indices. If the index for the path being pursued falls below a certain value it is abandoned and the system backtracks to the most likely path that was not pursued. The output of the parser is in the form of a set of variable values.

At the time that Woods developed his parser the augmented transition network concept was new and powerful. It fits the syntactic structure of English fairly well and Woods seems to have pushed the syntactic approach as far as it seems to go. To transcend the power of his parser, semantics, or the meaning of words, has to be brought into the parsing process. Woods' entire approach is syntactic in nature, however, and it is very difficult to incorporate semantics into his control structure. Besides, he has a good product that he maintains for a number of users and it is not clear if a major change in approach is worthwhile from his point of view.

A number of approaches have been tried to incorporate semantics into the parsing process. (See Winograd [76], Wilks

[74], Simmons [61], Martin [43] and Burton [10].) These range from using semantics in the recognition process and in the structure of the final parse to using general models of sentence purpose that decide the manner in which the sentence is to be interpreted. Burton's parser, for example, on encountering a request for a "measurement" invokes a "measurement" model that looks for certain types of information in the sentence.

Parsers are of little use by themselves. They usually act as front ends to language translation or question-answering systems. The process of understanding, therefore, takes place partly in the parser and partly in the processor that operates on the parse. Parsers that incorporate semantic reasoning absorb more or less of the processing that used to be carried out in the latter part of the system. In other words, the division between the parser and the processor is moved towards the processor. It is made at different points of the understanding process by different systems, however. Moreover, the representation and processing of semantic knowledge is different in each system and general guidelines have yet to emerge.

Our parser is relatively simple. It analyzes sentences on the basis of their syntactic types and uses semantics for the final case assignment. The parser contains a set of parse trees that set up appropriate transition networks and these are used to analyze the input sentence on the basis of the syntactic properties of each word. This analysis sets up the information contained in the sentence in named registers. Knowledge of word meanings is used to transform the contents of these registers into a case representation of the sentence.

The following sections describe the structure of our parser. It contains only ten sentence types and the more sophisticated analysis in terms of world models or sentence purpose models is relegated to the processor. The limited number of sentence types accepted by the parser eliminates the problem of multiple parses. This problem will appear, however, as additional sentence types are added and a decision will have to be taken whether to pursue all possible parses or use semantic reasoning to pursue only the most probable parse.

6.2.1 The Control Structure

The parser looks at the syntactic type of the next word in the sentence and attempts to start all the basic sentence units or "groups" that could begin with that word. In addition, it attempts to add the word to all the existing groups that are open at that point.

Starting a group puts the parser into a particular state where it expects subsequent words of certain syntactic types. If they are not of these types it resets its state and attempts to close the group with any words that may have been added to it. Exceptions to this method of starting groups by the syntactic type of the first word are the simple sentence (SS), the major clause (MC) and the secondary clause of the type "the man the dog bit". These are started by special mechanisms.

At each point the parser attempts to start all possible noun groups that it thinks may be useful. If a word has more than one syntactic type it tries all possibilities with each type. For example, "increase" is both a noun and a verb and the parser attempts to start a noun group and a verb group with it. If a noun group (NG) or a verb group (VG) are open when it is encountered it is added to them.

Let us consider an example. If the input sentence is "The man in the moon ate some cheese." the parser starts SS and MC immediately. On seeing "The" it checks its syntactic type, finds that it is a determiner and starts a noun group which is the only group that can start with a determiner. At this stage the stack of open groups is SS MC NG; proceeding down the stack from entities to constituents from left to right. The next word encountered is "man". This can fit in to the open NG but, being a noun, it can start a NG also. The system adds "man" to the open NG and considers whether it should start a subsidiary NG starting with "man" for the NG it has just started with "the". It finds that this is not useful and does not start it.

The next word encountered in the sentence is "in". This is looked up and found to be a preposition. The system finds that it cannot add "in" to the existing NG but it can start a preposition group (PG) subsidiary to the NG. It does this, and the stack grows to SS MC NG PG.

The next word is "the". The parser cannot add "the" to the PG directly but it can start a NG subsidiary to the PG. The stack increases to SS MC NG PG NG. "Moon" is added to lowest open NG and word "ate" is found to be next. The system realizes that it is a verb and can start a verb group but cannot be added to the NG, nor can it start a group subsidiary to the last NG. At this point it looks up the stack and for each open group checks to see if a verb group can be started. It finds that this is impossible for the PG and the two NGs but is possible for MC. Since the NG following MC is complete, it is closed and a VG started. The stack now is SS MC VG.

The next word encountered is "some". By reasoning similar to the above it closes the VG and starts a NG. This absorbs the final word "cheese". We now reach ".". This indicates that the sentence has ended. The parser looks to see if any groups can be closed and closes NG, MC and SS.

At this point we have the NG, VG and NG that make up the MC. The constituents of the various NGs have been strung together and, in fact, a considerable amount of knowledge has been used in doing this. Knowledge checks are applied before each word is accepted into the noun group and after it is accepted to make up the appropriate representation.

The final stage is to assign the noun groups as cases of the main verb. The first NG is found to be the "agent" since it is in the subject position, is capable of action and occurs before an active verb. The final noun group is found to be the "object" since it occurs immediately after the verb and is a noun that can be acted upon. The preposition group "in the moon" is found to be a location since "the moon" is a place and "in" can serve as a locative preposition. The case information can be represented as:

```
((AGENT EAT (KIND MAN))
(DETERMINER (KIND MAN) THE)
(LOCATION (KIND MAN) (KIND MOON))
(DETERMINER (KIND MOON) THE)
(OBJECT EAT (KIND CHEESE))
(DETERMINER (KIND CHEESE) SOME))
```

In each 3-tuple the first position indicates the name of the property or the relationship, the second position the entity whose property is being described and the third position the entity that stands in given relation to the second position entity. The "kind" before a noun indicates a generic class which is specialized by determiners, adjectives, etc.

The properties of the verb and the characteristics of the sentence have to be added to the case information to complete the parse. The verb is past tense and third person and may be singular or plural:

((TENSE EAT PAST)

(PERSON EAT THIRD)

(NUMBER EAT SINGULAR-PLURAL))

The sentence is declarative:

((TYPE EAT DECLARATIVE)).

6.2.2 Another Sample Parse

To accomplish the process described above the parser needs to know the syntactic type of each word. This enables it to decide whether the word can fit into a group and to make the appropriate case assignments. The parser also contains "parse trees" for each group that can appear during the parsing process. It is these parse trees that decide whether a new group should be started or not. A small portion of the question part of the major clause tree is shown in Figure 6.1.

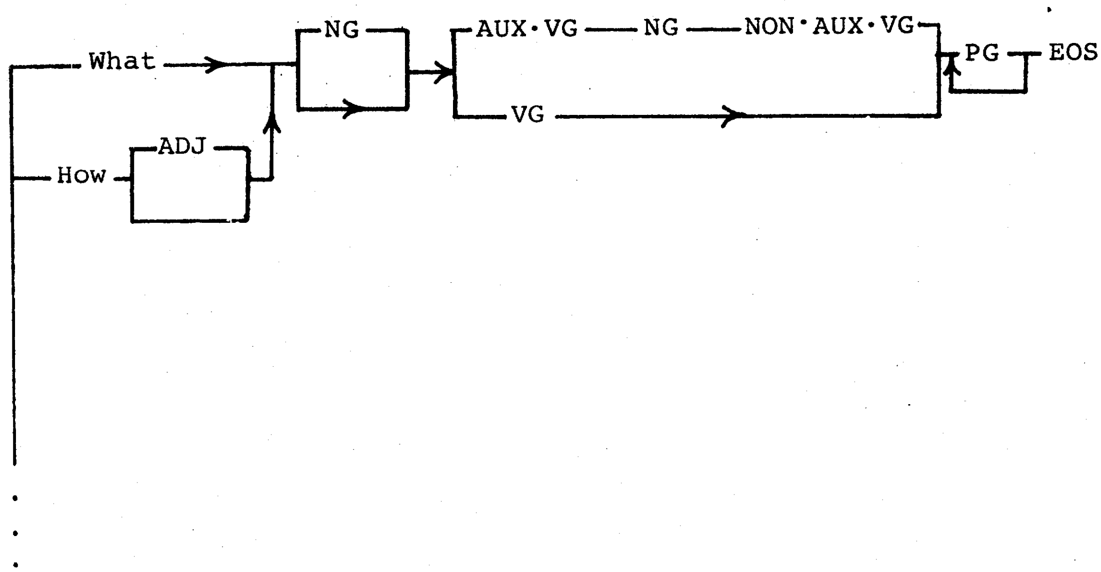
This shows that a major clause may be started by a "what" or a "how". The "what" may or may not be followed by a noun group (with trailing preposition groups) followed by a verb group which may or may not have a subjective noun group imbedded between the auxiliary verbs and the non-auxiliary verb. The verb group can be followed by any number of preposition groups and the tree ends with an end of sentence marker: ".", "?" or "!". A major clause starting with "how" may or may not be followed by an adjective. After this, it continues as the parse tree for "what" sentences.

Now, if we consider that a simple sentence can consist of a single major clause, we can follow the parsing of a question in somewhat more detail. Suppose the question is

"What percentage of sales to Sears were of product 1?"

the SS and MC groups are started immediately and "What" is absorbed into the MC. "Percentage" is a noun and starts a NG. The "of" starts a PG within the NG and "sales" a NG within the PG. At this stage the stack is SS MC NG PG NG. The following "to" closes the PG and NG and starts another PG setting the stack to SS MG NG PG. "Sears" starts another NG . The next word is "were". The first NG in PG and all its secondary groups are closed and an auxiliary verb group (AUX-VG) and a VG are started as "were" is capable of starting both. This is an example of a fork in which the parser pursues two paths

MAJOR CLAUSE TREE



ABBREVIATIONS:

NG	Noun Group
VG	Verb Group
AUX·VG	Auxiliary Verb Group
NON·AUX·VG	Non-Auxiliary Verb Group
PG	Preposition Group
ADJ	Adjective
EOS	End of Sentence

Figure 6.1

A Simplified Portion of the
Question Part of the
Major Clause Tree

simultaneously. The following word is "of". This clearly cannot start an NG and therefore the AUX-VG path is left hanging, the VG completed and a PG started. This is continued by "product-1" and closed by the "?".

It is easy to see how even this simple tree can parse a large number of sentences such as:

"What were sales to our largest customer in 1973?"

"How many people were killed in Vietnam last summer?"

6.2.3 Special Mechanisms

In addition to the basic control structure and the functions that build the representation of various sentence parts from the component words, there are a set of general mechanisms that take care of some facets of the language. These general mechanisms operate independently of the parse tree and the current situation in it.

6.2.3.1 Initial Preposition Phrases

Very often, initial prepositional phrases are used to set the time or place of the situation described in the sentence or otherwise qualify its contents. These phrases pertain to the entire sentence, rather than some part of it and can be transferred to the end of the sentence. For example:

"At plant 2 list the operating cost incurred in 1972 and 1973."

"For each of the factors just listed give the total value incurred at plant 4 in 1972 and 1973?"

6.2.3.2 Conjunctions

Conjunctions can be used in a variety of complex constructions in English. A simple mechanism that is capable of analyzing a large number of uses of conjunctions looks for a group (or a word) after the conjunction similar to the one that occurred before it. Thus, it is capable of analyzing conjoined sentences; "Show me the sales in 1972 and show me profit for the last five years.", conjoined noun groups; "overhead expenses and operating costs" and "actual and budgeted margins", conjoined preposition groups; "in 1972 and 1973" and even conjoined verb groups; "Compute and print the estimated profit if sales increased by 20%.". As we shall see in subsequent analysis such a mechanism seems to be satisfactory for a basic system. (See also Winograd [76].)

The conjunction mechanism should also be able to recognize a few special cases such as "1971, 1972, 1973", where the "and" is omitted before the final item in the list.

6.2.3.3 The "Found Group" Mechanism

As the control structure of the parser attempts to start all possible groups, there may be cases in which a particular group gets analyzed twice from two different directions. To prevent this, all groups that are analyzed are remembered. Thus, if a group that has been analyzed is restarted it is not recomputed, but its characteristics are retrieved from the previous computation. This is a means of improving the efficiency of the parser.

6.2.3.4 Ellipses

In addition to having facilities for parsing complete sentences, the parser also has special mechanisms for recognizing and parsing single word sentences, such as "Yes." and "No.", noun groups optionally followed by preposition groups and preposition groups all by themselves. These fragmentary sentences, or ellipses as they are called, arise in connected discourse, in response to questions, or as an abbreviation mechanism that uses information from previous sentences or from context. The following is a typical sequence:

What was product cost?

Do you want production cost or unit production cost?

Production cost for 1973?

$4,983,000

For 1972?

$3,782,000

6.2.3.5 Noise Words

Noise words, such as "please", are ignored by the parser.

6.2.4 The Knowledge Needed For the Parser

In addition to the syntactic type of each word the parser has to have access to a case-frame for each verb known to the system. This indicates the cases associated with the verb, their characteristics, the position in the sentence they can occur in and the properties of the nouns that can fill them. The verb "sell", for example, has the cases "agent", "recipient", "onject" and "exchange" associated with it. Time, location, manner and frequency cases can occur with all verbs and come in a variety of forms. With "sell", the the agent, typically, appears in the noun group before the verb or in the subject position. The recipient and object appear after the noun in the first and second object positions. Sometimes the object may move to the first object position. In such sentences, the recipient, if specified, appears in a preposition group starting with "to". The exchange information occurs in a preposition group typically starting with "for". Time and location information occurs in preposition groups or in a final noun group. The preposition groups may start with a number of different prepositions such as "at", "before", and "during" depending on the nature of the information.

Case frames, however, are very similar for classes of verbs and the system stores frames for each class and associates each verb with a particular class. This considerably reduces the knowledge that has to be stored in the system.

Each case also has requirements on the nature of the noun that can fill it. For "sell" the "agent" and "recipient" must have the capacity for independent action. "Object" and "exchange" can be pretty much anything but are typically an inanimate object and money, respectively. Both should possess some value, however. Correspondingly, each noun has properties that are checked by the case tests. Nouns may be animate or inanimate, active or passive, free or valuable, abstract or concrete, fixed or movable, etc.

Another kind of knowledge that the parser has access to is the relationship between nouns and adjectives and the properties of adjectives and nouns that allow them to be related to each other. Similarly, the parser has knowledge about nouns that can serve as classifiers to each other. This is important in analyzing sentences like:

"John bought Sue roses."

The parser has to realize that "Sue", being a proper noun, cannot serve as a classifier to "roses". Thus the verb is followed not by a single noun group "Sue roses" but by two nouns and that "Sue" must be the "recipient" and "roses" the "object" of the verb "buy".

6.3 ANALYSIS OF THE REQUESTS MADE BY THE SUBJECTS

Of the 496 sentences obtained from the subjects 387, or 78%, could be parsed correctly using a parser built along the lines described in section 6.2. Since a question-answering system does not expect to process declarative sentences we postulated a parser that does not have parse trees for declaratives. Thus, 36 declarative sentences that occurred in the protocols could not be parsed, as it were, by default.

The structure of the sentences was, for the most part, found to be fairly simple. Figure 6.2 shows the basic major clause parse trees that accounted for all the parsed sentences and the relative frequency of invocation for each tree. The trees are represented somewhat more simply than in Figure 6.1 by separating each path and by disallowing certain transitions. In addition, for all the question trees, it is assumed that a subjective noun group could occur between the auxiliary verbs and the main verb of the verb group. In addition to these trees, the parser must have a sentence tree. For our simple parser this consists of a single major clause. The parser must also have parse trees for noun groups, verb

<u>Wh- Questions</u>

1. What — BeVG — NG — nPG 35.6%

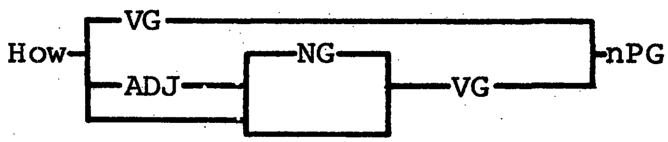
Example: What were sales to Sears from Plant-1 in 1972?

2. What — NG — VG — nPG 3.2%

Example: What information do you have about product cost?

3. Which — NG — VG — nPG 0.9%

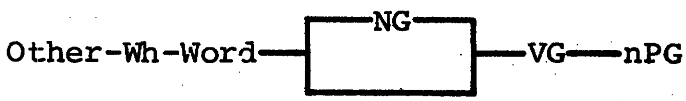
Example: Which plants were over budget in 1973?

4. 

3.3%

Examples: How is profit calculated?

How many plants do we have?

5. 

0.9%

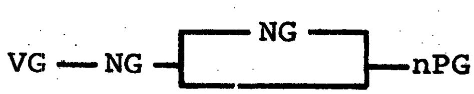
Example: Who was our best customer in 1972?

<u>Yes-No Questions</u>

6. VG — nPG 12.7%

Example: Can you calculate percentages?

<u>Imperatives</u>

7. 

34.0%

Examples: Show me the overhead cost for Plant 1.

Display sales from all plants in 1973.

Figure 6.2 The Ten Basic Sentence Types and Their Relative Frequencies of Occurrence in the Parsed Sentences

Ellipsis

8. NG —nPG 4.7%

Example: Product cost.

9. nPG 0.9%

Example: For each plant in 1973.

10. Single Words 2.9%

Example: Yes.

Notes:

1. Abbreviations: NG=Noun Group; VG=Verb Group; BeVG="Be" Verb Group; ADJ=Adjective; PG=Any number of Preposition Groups

2. In questions, a subjective Noun Group may occur between the auxiliary verbs and the main verb in the VG.

3. NGs may be followed by qualifying PGs.

4. "Other-Wh-Word" includes "who", "where", "when", etc.

Figure 6.2 The Ten Basic Sentence Types and Their Relative Frequencies of Occurrence in the Parsed Sentences groups and preposition groups. These are shown in Figure 6.3.

Of the sentences which were parsed, 35.6% asked for data or model values and started with "what" and were followed by a be-verb (is, are, were, etc.). These sentences followed the first tree shown in Figure 6.2.

The next most popular sentence tree was number 7. This allows commands that start with an action verb (print, compute, give, etc.) and ask the system to do something. Typically, to provide the value of a data item or a model. These sentences accounted for 34.0% of the parsed requests. The third largest group of sentences were yes-no questions that checked whether a particular statement about the simulated corporation or the system was true. These constituted 12.7% of the parsed sentences and were asked by the subjects to explore the capabilities of the system and to test their understanding of the situation.

Single noun groups and noun groups followed by preposition groups made up 4.7% of the parsed sentences. The remaining 13.0% of the parsed sentences were distributed among the six other sentence types shown in Figure 6.2.

Thus, we find that the few, simple parse trees diagrammed in Figure 6.2, supplemented by mechanisms to handle conjunctions and initial prepositional phrases and supported by a suitable knowledge base, were capable of parsing 78% of

NOUN GROUP

1. DET — ORD — NUM — ADJ — CLASF — NOUN

2. NG — PG

PREPOSITION GROUP

1. PREPOSITION — NG

ABBREVIATIONS:

DET	Determiner
ORD	Ordinal
NUM	Number
ADJ	Adjective
CLASF	Classifier
NG	Noun Group
PG	Preposition Group

Figure 6.3 The Noun Group and Preposition Group Trees

the sentences obtained from the subjects.

In the following section we will analyse the sentences that could not be parsed by the system and the additional facilities that would be required to handle them.

6.4 SENTENCES THAT COULD NOT BE PARSED

Unlike the sentences that could be parsed and fell into a few simple classes the sentences that could not be parsed, being exceptional, were of many different types. Basically, however, they can be grouped into sentences that are "bad English" or otherwise incorrect and sentences with complex syntactic features that were not included in the parser. It must be borne in mind, however, that the distinction between "bad English" and special syntactic forms is often arbitrary and a sentence type that is used frequently enough should be recognized no matter how idiosyncratic. The following sub-section describes improper sentences. Special syntactic conventions are described in later sub-sections.

6.4.1 Improper Sentences

Perhaps the worst example of improper sentences occurs when equations are used in the middle of a sentence.

"Display

((Sales in 1973 - Sales in 1972)/ Sales in 1972).

This could have been rephrased as "Display percentage increase in sales last year" or "Display the difference in sales between 1973 and 1972 divided by the sales in 1972". The rephrased sentences are fairly simple but equations seem to come naturally to users in certain situations. It is not difficult for the system to accept equations but since a normal typewriter terminal is not well adapted for typing equations a set of special conventions will need to be adopted. This will necessitate an instruction manual or a tutorial subsystem.

Equations also occur in defining models:

"Define Discount (x) =

(List Price (x) - Selling Price (x))/(List price (x))

Although this can also be rephrased into a simple sentence it must be conceded that describing models in single sentences is often difficult. At the same time, it is important to allow the user some rudimentary model definition capability. Most reasonable models, however, seem to take more than one sentence to describe. This gets into the problem of understanding connected discourse and is a little too complex, given the state of the art, to incorporate into a question-answering system. The alternative seems to be to allow models to be defined as single sentences:

"Define %increase as the percentage increase over 1972?"

and/or to allow the user to invoke and enter a speciai model definition subsystem in which the system structures the dialog by asking questions. This may be cumbersome but seems to be the best alternative in the near future.

The other major class of improper sentences may be called "bad English". There is, however, a continuum between "bad English", "informal usage" and sentence types not recognized by the system and the decision to accept a certain sentence type must be based on how often it is used. Let us consider some examples.

"What's difference between list price and average quotation price?"

If we can deduce that "What's" is a contraction for "what is" and do not insist on a determiner before "difference" then this sentence should be accepted. It is a little deviant but close enough to be parsed. Consider however:

"What would have 1973 profits have been compared to 1972 if the product mix had not changed in those two years?"

This is a long sentence and the user inserted an extra "have". If it is not accepted he will realize his error and retype it. Similarly:

"List prices of single unit prices for both 72 and 73."

The subject has mixed up "List single unit prices" with "List prices of single units". He will realize his error, however, if the system rejects the sentence. Consider however:

"What was the overhead cost, 1972 vs 1973?

This is idiosyncratic usage. The comma is made to do the work of a preposition. It should be rejected but there are four examples of it in the 496 sentences and perhaps it makes sense to accept it as an exceptional case. The same sort of problem occurs in "for 1971, 1972, 1973". Clearly, this is incorrect usage as an "and" should appear before "1973". If this happens often enough, however, it may be worth accepting as a special case.

Another example of non-standard usage is the parenthetical use of afterthought qualifiers and adjectives.

"Display the direct cost variance (absolute $ and %) for all plants"

"Please display overhead figures (actual and budget) for all plants?"

Or even worse:

"What was sales by product (5 products) for 1972 and 1973?"

"Display the difference between the list price and actual cost (direct + overhead) divided by list price."

These constructions are not difficult to accept as the parenthetical information modifies the entire noun group preceding it. Whether it is worth incorporating them in the parser as yet another special case depends on how often they occur and how important the users feel them to be.

Another interesting class of improper sentences occurs when the initial "What is" is omitted from a question:

"Profit margins for each product."

"Total profit."

The user may be reacting to the redundancy in English that makes the "what is" necessary or he may think he is issuing commands to a computer system. The "what is" does tend to get dropped only from routine requests for data and it may be desirable to create a default that responds to noun groups that are data items by providing their values. The SOPHIE system [7] adopts this default and responds to noun groups by providing their values, if this is possible.

In the same way, the initial auxiliary verb may be omitted from a yes-no question.

"Any equipment purchased for long term depreciation?"

This is more unusual, however, and it may not be worth creating a special mechanism to take care of it.

As we consider the many examples of exceptional usage by the subjects it becomes clear that a real life system will evolve continuously to accept new styles of usage in addition to new data, additional facilities and more knowledge. This is a basic feature of knowledge-based systems. All parts of the system should, therefore, be designed to make such modifications convenient as a natural feature of its use.

6.4.2 Sentence Types Not Included in the Parser

The basic parser described earlier included only the parse trees for ten types of sentences. We showed that some 78% of the sentences obtained from subjects were covered by these sentence types. Of the sentences that could not be parsed, some were of a form for which the parser did not have a parse tree or used syntactic conventions that the parser did not recognize. These sentence types are discussed in this section. Each type of sentence occurred only a few times, but, taken together, these additional sentence types cover a substantial number of the input sentences. A successful parser would, therefore, have to accept at least the more popular of these forms.

6.4.2.1 Preposition Before Wh-word

The initial preposition serves to indicate the nature of the answer required.

"For what year was that figure?"

"By how much did operating cost increase in 1973?"

"By what percent did overhead expenses
in 1973 increase over those in 1972?"

The last may be somewhat informal, though. The preposition can be used to indicate the case relationship asked for.

"By whom was the decision taken?"

"For what are these loans used?"

6.4.2.2 Prepositions and Adverbial Forms at the End of the Sentence

The initial preposition may sometimes travel to the end of the sentence.

"What are these loans for?"

"What was the increase in operating cost due to?"

"What was the increase in operating cost caused by?"

Complex adverbial forms may appear at the end of the sentence to indicate the nature of the answer.

"Give me the budget for each plant and
the overruns if any."

"Do you have any model at all?"

"Have they been this way for the past years too?"

6.4.2.3 Complex Verb Groups

The basic parser assumes a verb group consisting of a modal (will, may, should), a "have" verb, a "be" verb and an action or semantic verb. All of these are optional, subject to the constraint that at least one verb must be present.

Some sentences, however, contain verb groups that deviate from this pattern:

"Where does operating cost get included?"

"Why was there such a great increase in operating cost?"

"How many plants are there?"

The "there" following the "be" verb seems to be a special convention indicating existence.

6.4.2.4 "Which" Constructions

Normally, a "which" starting an identity question is followed by a noun or a verb.

"Which plant had the maximum production in 1972?"

"Whicn is our largest plant?"

In some cases, however, the noun may be replaced by more complex structures.

"Which of our four plants had the largest sales in 1973?"

"Which product of the five had the largest variance?"

"In 1972 which product or products had the largest variances?"

The word "what" is often used instead of "which" in identity questions.

"What components of overhead cost went up last year?"

This may not be good English but it seems to occur frequently and should, therefore, be accepted.

6.4.2.5 Comparative Clauses

Comparative clauses consist of a comparative adjective followed by "than" followed by a noun group. The final noun group may provide a value or name an entity with whom the comparison is to be made.

"Which plants were over budget by more than 2%?"

"Please express numbers over 100 but less than 100000 in units of thousands."

"Is plant 5 larger than plant 3?"

Comparative clauses are essential for indicating conditional retrieval. They must, therefore, be allowed since conditional retrieval is very important in the problem domains for which a system of this kind would be designed.

6.4.2.6 "What-if" Constructions

What-if constructions are sometimes used to evaluate the effect of hypothetical policies or test the effect of a certain factor:

"What would profits have been if sales would have increased to 55 million dollars?"

A what-if branch tree will have to be added to the simple sentence tree if such forms are to be accepted. Alternate forms start with "suppose":

"Suppose the sales in 1973 had remained unchanged, would profit have altered if the selling price of product 1 have been increased to allow a margin of $5.5?"

6.4.2.7 Colon Clauses

A colon (:) followed by a string of noun groups, with optional preposition groups, is often used to ask for a set of information.

"Give me two tables: sales by plant and production by product."

"Give me the following information: sales, production and total costs for 1973?

The same device can also be followed to give instructions:

"Compute profit according to the following formula: ---."

6.4.3 <u>Special</u> <u>Mechanisms</u>

These require special recognition routines in the parser rather than additional parse trees.

6.4.3.1 Relative Clauses

A frequently used mechanism for qualifying noun groups is the use of a relative clause.

"Show me the sales for all products that were produced at plant 3?"

The "that were produced at plant 3" is a relative clause that qualifies the noun group "all products". The "that" and the "were" may disappear leaving sentences like:

"Show me the sales for all products sold by plant 3?"

This construction occurs frequently and gives rise to examples like:

"Show me the various costs attributable to each product."

"What are the costs associated with each product?"

"List all data you know about."

In exceptional cases, the preposition may also disappear:

"What is the difference between the prices charged our customers and list prices?"

Relative clauses may appear in many different guises:

"Has the product mix changed for any plant whose profitability decreased?"

"Have the sales decreased for any product which is produced in the Midwest?"

The "which is" will often disappear in constructions like the last sentence.

The relative clause mechanism is very powerful and is used extensively to specialize noun groups. In fact, almost 20% of the sentences that could not be parsed by the basic parser contained relative clauses.

6.4.3.2 Participles

Since single word prepositions are inadequate for expressing all the subtle relationships that may exist between entities, they are sometimes augmented by adding a verb, usually in the past tense, or a verb and a particle before them.

"What would have 1973 profits been compared to 1972
if the product mix had not changed?"

"List the sales of products produced by plant-1
broken down by product?"

Here "compared to" and "broken down by" serve as prepositions. Similarly, "analyzed by", "distributed by", "fragmented by", etc., can also serve as complex prepositions and indicate the form in which the answer is required.

An especially interesting case arises when participles are used to specify mathematical operations on data.

"Display profit for each plant divided by
plant sales."

6.4.3.3 Verbs as Adjectives

Entities associated with certain kinds of actions or produced as a result of them may be specified by the operative verb. The verb may appear as an adjective in the noun group naming the entity.

"Do you have a forecasting model for demand?"

"What was the inventory of finished goods at the end of 1972?"

"What were shipping costs in 1973?"

"Do you have any information on repeat customers?"

6.4.3.4 A More Powerful Conjunction Mechanism

Of the 496 sentences obtained from users, 141 were found to contain conjunctions. By an overwhelming majority "and" was the most popular conjunction. Of the 141 sentences, 68 contained the usage "1972 and 1973" with minor variations. The remaining 73 sentences used conjunctions in a variety of forms, most of them simple.

In the simplest case, complete noun groups were conjoined. This was the largest percentage in the 73 sentences. The next most popular usage was to use conjunctions between components, typically adjectives and classifiers, of noun groups. Other simple constructions included conjoined preposition groups and conjoined sentences. All of these forms can be accepted and correctly analyzed by a conjunction mechanism that looks for a group after the conjunction similar to the one that appeared before it.

In the case where noun groups followed by preposition groups are conjoined, such a syntactic mechanism will give two interpretations. Consider "sales in 1972 and production for 1973". The parser will not be able to tell whether a noun group ends with "1972" or if "1972 and production" is the noun group for the preposition group starting with "in".

This ambiguity cannot be resolved at a syntactic level. Semantic knowledge must be invoked either by the conjunction mechanism or during the understanding process.

In discussing the simple conjunction mechanism contained in the basic parser we discussed the desirability of accepting aberrant conjoined noun lists of the form "1971, 1972, 1973", adding a default "and" before the last element. Similarly, it may be desirable to accept badly constructed lists like "plant 1, 2 and 3 and 4".

A more complex problem arises in the interpretation of the sentence:

"What is the difference between plant 1 and plant 2 and plant 3 and plant 4?"

This is impossible to analyze unless we outlaw lists such as "plant 1, 2 and 3 and 4" with more than one conjunction. If these lists are excluded, however, we can use the knowledge that "between" must be followed by a plural or a conjoined list and the only legal possibility is "(plant 1 and plant 2) and (plant 3 and plant 4)".

Sentences can also be joined together. They often give rise to difficult problems of interpretation as the second sentence may assume some of the information provided in the first. Consider:

"What is the increase in production cost and the ratio of this increase to the increase in sales?"

The problem of correctly attributing "this" to "increase in production cost" is relatively simple, however. Similarly, "they" has to be associated with "handling costs" in:

"What are the handling costs associated with each product and did they change over the last two years?"

Another complex example is:

"What percentage of overhead cost is interest cost and what percentage is operating cost?"

In this example the prepositional phrase "of overhead cost" is omitted from the second sentence. In

"Would profit have increased if prices had been increased by $2.0 and by how much?"

the second sentence takes the entire first sentence as context and omits the noun group, verb group and if clause. (See also Winograd [76].)

6.5 CONCLUSIONS

The basic conclusion that emerges from the analysis of the sentences obtained from the subjects is that they mainly used a few sentence types that can be analyzed by a fairly simple parser. In fact, a parser of the type described in this chapter, using the meanings of words to decide upon the analysis to be performed and the structure of the final parse and equipped to handle some twenty basic sentence types and other syntactic conventions would provide enough power to serve as a front end to an English language system to support management.

In analyzing the requests made to the moon-rocks system Woods [73] found extensive use of relative clauses. Our requests do contain relative clauses but they are few and far between. The moon rocks, however did not have convenient names and had to be referred to by their properties. Also, users often wanted to investigate subsets that displayed a conjunction of properties. This kind of request requires relative clauses. In our data base different kinds of data have names that indicate their properties. These are convenient to use and, thus, relative clauses appeared less often. The structure of the grammar required is, thus, predicated, to some extent, on the nature and structure of the data base and the operations to be performed on it. Our results would hold, therefore, for corporate data bases used for management decision-making and less so for other kinds of problems and data base structures.

6.6 THE UNPARSED SENTENCES

These are the sentences that could not be parsed by the basic parser. A few duplicate sentences have been deleted.

1. I BELIEVE YOUR OVERHEAD VARIANCE ACCOUNTS FOR YOUR LOWER THAN EXPECTED PROFITS?
2. I SUPPOSE I SHOULD.
3. WHAT IS THE DIFFERENCE BETWEEN PLANT 1 AND PLANT 3 AND PLANT 2 AND PLANT 4?
4. WHERE DOES TRANSPORTATION COST GET INCLUDED?
5. I WOULD LIKE TO END THE INTERVIEW.
6. WHAT WAS COST OF PRODUCING EACH PRODUCT FOR BOTH 1972 AND 1973?
7. PRODUCTION COST FIRST FOR ONE UNIT.
8. WHAT'S RATE OF UNIT COST FOR EACH YEAR AND THE RATIO OF THIS PRODUCTION INCREASE TO PRODUCT PRICE?
9. WHAT IS THE PERCENT OF INCREASE OF EACH PRODUCT FOR EACH YEAR STUDIED?
10. WHAT'S DIFFERENCE BETWEEN LIST PRICE AND AVERAGE QUOTATION PRICE?
11. PRODUCT 4 AND 5 SHOW GREATEST DIFFERENCE BETWEEN LIST AND QUOTATION PRICES, WHY?
12. IT SHOULD BE INCLUDED.

13. DO YOU HAVE FURTHER BREAKDOWNS OF OVERHEAD ATTRIBUTABLE TO EACH PRODUCT WITHIN PLANTS?

14. LIST ALL DATA ITEMS YOU KNOW ABOUT.

15. FOR 1973 LIST THE SALES OF PRODUCTS PRODUCED BY PLANT ONE, BROKEN DOWN BY PRODUCT.

16. FOR 1973 AND PLANT 1 LIST DIRECT MANUFACTURING EXPENSES BY PRODUCT AND ALSO TOTAL OVERHEAD.

17. IN THE FUTURE, PLEASE EXPRESS NUMBERS OF OVER 100000 IN TERMS OF UNITS OF MILLIONS, AND NUMBERS OVER 100 BUT LESS THAN 100000 IN UNITS OF THOUSANDS.

18. WHAT ARE THE COMPONENTS OF THE VARIOUS COSTS YOU KNOW ABOUT?

19. WHY WAS THERE SUCH A GREAT INCREASE IN OPERATING COST IN PLANT 0?

20. PRINT EVERY PIECE OF INFORMATION YOU HAVE CONCERNING PLANT 0 IN 1972 AND 1973.

21. DISREGARDING PLANT 0 TOTALLY, WHAT IS THE DIFFERENCE IN TOTAL PROFIT BETWEEN 1972 AND 1973?

22. WHAT COMPONENTS OF THE OVERHEAD COSTS GO UP MORE THAN 2%
?

23. WHAT WAS THE INCREASE IN INTEREST COST DUE TO?

24. WHAT WOULD HAVE 1973 PROFITS HAVE BEEN COMPARED TO 1972 IF THE PRODUCT MIX HAD NOT CHANGED IN THOSE TWO YEARS?

25. WHAT PERCENT OF OVERHEAD COST IS INTEREST COST

AND WHAT PERCENTAGE IS OPERATING COSTS?

26. I KNOW WHAT THE PROBLEM IS.

27. WHAT WERE THE OUTSTANDING LOANS, 1971, 1972, 1973?

28. ANY EQUIPMENT PURCHASED FOR LONG TERM DEPRECIATION?

29. DO YOU HAVE INFORMATION ABOUT WHAT THESE LOANS ARE FOR?

30. OK I THINK I KNOW WHAT THE PROBLEM IS.

31. HOW MUCH OF EACH PRODUCT WAS PRODUCED IN 1972 AND IN 1973?

32. GIVE ME THE BUDGET FOR EACH PLANT AND THE OVERRUN IF ANY?

33. PLEASE COMPUTE THE FOLLOWING: PERCENT CHANGE IN UNIT SALES, PERCENT CHANGE IN UNIT PRODUCTION COST FROM 1972 TO 1973?

34. DO YOU HAVE A FORECASTING MODEL FOR DEMAND?

35. DO YOU HAVE ANY MODEL AT ALL?

36. LIST THE FUNCTIONS YOU CAN PERFORM.

37. ARE THERE ANY VARIANCES BETWEEN ACTUAL PRICES CHARGED OUR CUSTOMERS AND THE GUIDELINE PRICES?

38. DO YOU HAVE A MODEL TO MAXIMIZE CONTRIBUTION TO THE COMPANY SUBJECT TO PRODUCTION AND OTHER CONSTRAINTS?

39. WHY WERE THE QUOTATION PRICES LOWER THAN LIST PRICES IN 1973?

40. HAVE THEY BEEN THIS WAY FOR THE PAST YEARS TOO?

41. COMPUTE PROFIT FOR 1972 AND 1973 ACCORDING TO THE FOLLOWING FORMULA: ACTUAL UNIT SALES BY PRODUCT

TIMES LIST PRICE MINUS PRODUCTION COST FOR THE PRODUCT SUMMED OVER ALL PRODUCTS LESS OVERHEAD COST FOR THE YEAR.

42. I THINK I UNDERSTAND THE PROBLEM. THANK YOU.

43. WHAT ARE THE HANDLING COSTS ASSOCIATED WITH EACH PRODUCT AND DID THEY CHANGE OVER THE LAST TWO YEARS?

44. HANDLING COSTS ARE COSTS ASSOCIATED WITH PRODUCTS THAT ARE NOT REFLECTED IN DIRECT MFG COSTS.

45. THE INTENT OF MY QUESTION IS TO FIND OUT IF YOU KNOW IF YOUR ACCOUNTING METHODS CAN RELATE THE CHANGES IN SALES TO CHANGES IN YOUR EXPENSE STRUCTURES. DOES THIS HELP?

46. PLEASE GIVE ME CHANGES IN EACH TYPE OF COST ASSOCIATED WITH EACH PRODUCT?

47. IN AS MUCH AS ALLOCATING COSTS IS A TOUGH JOB I WOULD LIKE TO HAVE THE TOTAL COSTS RELATED TO EACH PRODUCT. I MEAN I WOULD LIKE THE COST OF EACH PRODUCT BROKEN DOWN ON A DIRECT AND INDIRECT BASIS?

48. HAVE ANY PLANTS BEEN SUPPLYING BATTERIES TO OTHER THAN NORMAL CUSTOMERS IE OUTSIDE OF THEIR NORMAL SALES DISTRICT?

49. PLEASE DISPLAY OVERHEAD FIGURES (ACTUAL AND BUDGET) FOR ALL PLANTS FOR THE PAST FOUR YEARS?

50. WHICH PLANTS WERE OVER BUDGET ON OVERHEAD BY MORE THAN 5% ?

51. WHICH PLANTS WERE OVER BUDGET ON FIXED COSTS BY MORE THAN 5 % ?

52. DISPLAY PROFIT FOR EACH PLANT DIVIDED BY PLANT SALES.

53. I SUGGEST WE GET RID OF PLANT ZERO!

54. HAS PRODUCT MIX CHANGED IN ANY PLANT WHOSE PROFITABILITY HAS FALLEN OFF?

55. HAS PRODUCT MIX CHANGED BY MORE THAN 1 % IN ANY PLANT WHOSE PROFITABILITY HAS DECREASED?

56. DISPLAY THE DIRECT COST VARIANCE (ABSOLUTE $ AND %) FOR ALL PLANTS.

57. HAS THERE BEEN A DECREASE IN CONTRIBUTION MARGINS FOR ANY PRODUCT?

58. DISPLAY THE OVERHEAD DIVIDED BY SALES (%) FOR EACH PLANT.

59. WHY ARE THE OH FIGURES FOR PLANTS 2 AND 4 HIGHER THAN FOR 1 AND 3?

60. BY WHAT PERCENT DID THE OVERHEAD EXPENSES IN 1973 INCREASE OVER THOSE IN 1972?

61. GIVE ME DETAILS OF HOW THE ADDITIONAL SALES REVENUE IN 1973 WAS SPENT.

62. WHAT ARE THE RELATIVE PERCENTAGES OF EACH PRODUCT SOLD BY EACH PLANT?

63. WHICH OF THE FOUR PLANTS HAD THE LARGEST VALUE FOR TOTAL SALES IN 1973?

64. AT PLANT 2, WHICH PRODUCT ACCOUNTED FOR THE LARGEST

PERCENTAGE OF TOTAL SALES IN DOLLARS?

65. DOES PRODUCT 2 ALSO ACCOUNT FOR THE LARGEST PERCENTAGE AT PLANT 4?

66. WHAT IS THE NUMBER OF UNITS OF PRODUCT 2 PRODUCED AT PLANT 2 IN 1973 TIMES THE UNIT COST OF PRODUCT 2?

67. DEFINE THE TERMS "UNIT COST" AND "UNIT PRICE".

68. WHAT WAS THE NUMBER OF UNITS OF PRODUCT 2 PRODUCED AT PLANT 2 IN 1973 TIMES THE UNIT PRICE OF PRODUCT 2?

69. FOR EACH OF THE FACTORS JUST LISTED GIVE THE TOTAL VALUE INCURRED AT PLANT 2 IN 1972 AND 1973.

70. AT PLANT 2 LIST THE OPERATING COST INCURRED IN 1972 AND 1973.

71. FOR DEPRECIATION MANAGEMENT SALARY AND INTEREST COST LIST THE AMOUNTS INCURRED IN 1972 AND 1973.

72. IN 1973 WHAT PERCENTAGE OF THE DIRECT MANUFACTURING COST WAS ACCOUNTED FOR BY OPERATING COST.

73. DEFINE P-COST TO BE THE SUM OF OVERHEAD COST AND MANUFACTURING COST. WHAT PERCENTAGE OF THE P-COST IS ACCOUNTED FOR BY OVERHEAD COST?

74. FOR WHAT YEAR WAS THAT FIGURE?

75. GIVE ME TWO TABLES, THE CONTRIBUTION MARGIN FOR ALL PRODUCTS IN EACH PLANT FOR THE YEARS 1972 AND 1973.

76. GIVE ME THE FOLLOWING PROPORTIONS:

THE SALES OF PRODUCTS ONE, TWO AND FIVE DIVIDED BY THE

TOTAL SALES FOR 1972 AND 1973.

77. WAS THE ACTUAL OVERHEAD EXPENSE IN PLANT 4 HIGHER THAN THE BUDGETED AMOUNT IN 1973?

78. BY HOW MUCH?

79. SUPPOSE THE SALES IN 1973 HAD REMAINED UNCHANGED, WOULD THE PROFIT PICTURE HAVE ALTERED IF THE SELLING PRICE OF PRODUCT 1 HAD BEEN INCREASED TO ALLOW A PROFIT MARGIN OF $5.5, AND BY HOW MUCH?

NEXT, WOULD THE SALES HAVE ALTERED SIGNIFICANTLY IF THERE HAD BEEN THIS PRICE INCREASE.

80. EVEN THOUGH THE PLANTS ARE NOT OPERATED AS PROFIT CENTERS, COULD YOU TELL ME THE CONTRIBUTION TO PROFITS FROM EACH PLANT FOR THE YEARS 1972 AND 1973?

81. THE RATIO OF PRODUCTS COSTING $6.25 AND $5.00 FROM EACH PLANT DURING 72 AND 73?

82. I WANT THE SUM.

83. REMEMBER THIS REQUEST (CALL IT REQUEST A).

84. PLEASE RESPOND TO REQUEST A FOR YEARS 1972 AND 1973.

85. PLEASE RETAIN THE RESULTS OF SPECIFICATIONS UNTIL I CHANGE THEM.

86. DISPLAY ((SALES IN 1972 - SALES IN 1973)/SALES IN 1972).

87. REMEMBER TO RETAIN SPECIFICATIONS OF PREVIOUS REQUESTS.

88. AGAIN BY PRODUCT PLEASE.

89. NO THEY AREN'T.

90. LET ALLOC BE

((OVERHEAD/PRODUCTION COST)*TOTAL PRODUCTION COST)

91. DO YOU HAVE A LIST OF PRODUCTION COST ITEMIZED

PER TYPE OF DIRECT COSTS?

92. DEFINE EQUATION

DISCOUNT(X) =

(LIST PRICE(X)-SELLING PRICE(X))/(LIST PRICE(X)).

93. SOLVE DISCOUNT (PRODUCT 2),

THEN PRINT DISCOUNT (PRODUCT 2).

95. DEFINE %SALES(X) =

(TOTAL SALES PRODUCT(X))/(TOTAL COMPANY SALES).

96. PRINT THE NUMBER OF UNITS OF EACH PRODUCT PRODUCED

BY PLANT.

97. DEFINE %CHOVERHEAD(T) =

(OVERHEAD(T)-OVERHEAD(T-1))/(OVERHEAD(T-1)).

98. DEFINE %CH (ITEM T) = (ITEM(T)-ITEM(T-1))/(ITEM(T-1)).

99. WHAT WAS THE PERCENTAGE INCREASE IN OVERHEAD COST,

1973 VS 1972?

100. ARE ALL INCREASES FROM FREIGHT CARRIERS PASSED ON TO

THE CUSTOMER?

101. WHAT WERE THE SALES BY PRODUCT (5 PRODUCTS) FOR

1972 AND 1973.

102. DO YOU HAVE A MODEL FOR MEASURING CUSTOMER SERVICE?

103. DO YOU HAVE A COUNT OF THE NUMBER OF SALES REQUESTS

AND THE NUMBER OF REQUESTS FILLED?

104. WHICH PRODUCT OF THE FIVE HAD THE LARGEST PERCENTAGE VARIANCE?

105. IN 1972 WHICH PRODUCT OR PRODUCTS HAD LARGEST VARIANCES?

CHAPTER 7

THE KNOWLEDGE BASE

A system of the type proposed in this thesis provides one principal advantage over conventional decision-support systems: it understands the user's requests phrased in a natural manner and frees him from having to specify solution procedures in complete detail. This is possible because the system embodies knowledge about the problem situation and shares a set of common knowledge with the user. The power of the system and its utility to the manager will depend on the quantity and quality of the knowledge it contains about the problem domain. Specifying and categorizing the knowledge required is, therefore, central to the design of such a system. The amount and complexity of the knowledge required also has a bearing on the feasibility of systems of this type.

This chapter analyzes the knowledge required for a system designed to support management in a particular area. It does this by analyzing the requests made by the subjects in attempting to solve the realistic management problem described in Appendix I and developing the different categories of knowledge required to understand and respond to them. To provide an estimate of the total amount of knowledge required in a domain-specific system, Appendix III contains an attempt to specify the corpus of knowledge required to respond to the 496 requests obtained from the subjects.

The requests obtained from the subjects, which can be considered to be typical of those that will be made to a management-support system of this kind, can be divided into two major classes: requests for information about the problem situation and requests for information about the contents and capabilities of the system. The following pages present examples of different types of requests within these two broad classes. The succeeding sections develop the knowledge required to respond to each of the different kinds of requests. Additional detail about response analysis strategies and the knowledge required for them will be found in Appendix III.

REQUESTS ABOUT THE PROBLEM SITUATION

Aggregate Data

What were sales in 1972?

What was production by plant by product?

What is operating cost for each plant?

Properties of Entities and Identity Questions

How many plants do we have?

Which products are made by plant 4?

Which is our largest plant?

Models and What-If Questions

What was profit in 1972?

What was contribution margin for each plant?

What would profits have been if there was no deviation between selling price and list price?

Would sales have decreased if the price of product 5 was raised to give a margin of $2?

For Functions Of Data

What are the average sales to each customer?

What is the ratio of overhead cost to sales for the last 2 years?

What is the percentage increase in sales of each product in 1973?

Yes-No Questions

1. About the corporation

Do we have five plants?

Does each plant manufacture every product?

Do we have any repeat customers?

Was any equipment purchased for long term depreciation?

2. Asking if a sub-problem exists (See Chapter 8.)

Did the product mix change for any plant whose profitability had decreased from last year?

Were profit margins maintained in 1973?

Did overheads increase more than 5% in any plant?

Did overhead costs increase significantly last year?

Causality

Why was there such a great increase in operating cost in plant 0?

Products 4 and 5 show the greatest deviation of selling price from list price, why?

Do you have any information what these loans are for?

Model Definitions

Define p-cost to be the sum of overhead and production cost.

Define %chcost =

((Cost in 1973 - Cost in 1972)/(Cost in 1972))

REQUESTS ABOUT THE SYSTEM

Regarding Capability

1. Computational Capability

   Can you calculate percentages?

   Can you format reports?

   List all the functions you can perform.

2. Content Capability

   Can you produce a profit figure for each product at a specific plant?

   Can you give me data on product mix from each plant?

Regarding Contents

How far back does your information go?

Do you have variable budgets?

Do you have a forecasting model for demand?

Do you have any information on customer satisfaction?

List all the data items you know about.

Regarding Composition of Data Items

Give me a breakdown of items in your overhead.

Do overhead costs vary with volume?

Where does transportation cost get included?

Are production cost and manufacturing cost the same?

What makes up operating costs?

<u>Definition of Data Items and Models</u>

Define the terms "unit cost" and "unit price".

How is profit calculated?

What is the definition of profit for a product?

7.2 UNDERSTANDING AND ANSWERING REQUESTS FOR DATA

The distinction between data and properties of entities is, to some extent, artificial as the production or profit of a plant is as much its property as its location. It arises due to the nature of the system as a front end to a data base. The contents of the data base, which may have been created for other purposes, are called data. Additional information is stored in the knowledge base in different structures and with different names. Data is stored in arrays of upto four dimensions. The dimensions are called its keys. It tends to be stored independently, under its own name, being a property of the parent corporation, whereas the properties of entities tend to be attached to the entity.

Models, which are mathematical functions of data, are referred to in exactly the same way as data in the subjects' requests. This is appropriate since what is stored as data and what is computed as a model depends on how the data base is organized.

Let us consider requests for data first. The data item is referred to by name and its key values are usually specified as noun groups contained within prepositional phrases in the sentence. The general method for analyzing and responding to requests for data is, therefore, to analyze the noun groups in the request and associate them with the name of the data item and the key specifications for its retrieval. Once the slots for the data name and the key specifications are filled from the sentence, or by default, as explained in Chapter 3, a program for retrieving and formatting the data can be prepared.

Filling the slots can be fairly complex. The data item stored as sales, for example, may be referred to as "sales revenue", "records of sales", "total sales", "gross sales", "sales figures" and "gross sales figures". Similarly, profit may be referred to as "gross profit", "overall profit", "pre-tax profit", "profit figure", etc. "Figure" and "value" seem to be used in a generic manner with most data items. Adjectives may be used to further specialize the data name. For example, "budgeted cost", "expected revenue", "product cost", "overhead expenses", "unit sales" etc. Prepositional groups may also be used to complete the description of the data required. Thus, "cost of production" and "cost of goods sold". Profit may also be requested as "profit on operations" and "profit for the company".

Thus, the problem of analyzing the main noun group and associating it with names known to the system is a formidable one. This will be referred to as the naming problem. In general, a complex structure of equivalences and noun idioms and an analysis routine that processes the noun group and certain types of preposition groups is required to decide exactly which data item is being asked for. This routine must ensure, for example, that requests for the "price of lead" and "lead price" are answered in the same way.

The problem of determining key variable specifications from information contained in prepositional phrases is similar, though generally somewhat simpler. Key values may be specified as: "for product 3", "by plant", "for all customers", "for each battery type", "for product 1 through product 5", "for each plant separately", etc. The preposition is not very useful in indicating the key variable specified. Plant specifications, for example, may be preceded by "at", "in" and "by" and the more general "per", "of" and "from". The nature of the noun group has to be analyzed before it can be associated with a particular key with confidence.

Time period specifications can, however, take on a large variety of forms. Figure 7.1 lists nineteen forms that were used by the subjects in their requests. Others, such as "for all available years" come readily to mind. The interpretation problem this creates is somewhat alleviated by the fact that time and space nouns are unique in character and rarely ambiguous. This property is used in the syntactic convention that allows time or space nouns to terminate a sentence.

"What were our sales of product 1 last year?"

"Could we have increased our sales overseas?"

In rare cases, key specifications may occur as adjectives of the main noun group. For example, "monthly sales". They may also occur in relative clauses such as "all products produced by plant 3".

1. By year
2. For '71 (71, 1971)
3. For the year 1972 (for the years 1972 and 1973)
4. For each year
5. For each year studied
6. In (for) each of the last (past, previous) 2 (two) years
7. For the last 2 years, 1972 and 1973
8. In the previous year
9. In the most recent 2 years
10. In (for) 1972 and 1973
11. For both 1972 and 1973
12. In 1972 and (&) in 1973
13. For 1972 vs 1973
14. For 1969 through 1973
15. Ratio of 1973 sales to 1972 sales
16. From 1972 to 1973
17. In 1973 over 1972
18. 1971 1972 1973 1974 (71 72 73 74)
    With or without commas between items and with or without "and" or "&" before the last item.
19. At the end of 1972

FIGURE 7.1

TYPES OF PREPOSITION GROUPS SPECIFYING TIME PERIODS

7.3 COMPLEX PROBLEMS IN REQUESTS FOR DATA

Not only can data, such as sales, be referred to by a large variety of noun constructs but they can also be invoked as the result of the appropriate verb.

"How much did we sell to Sears in 1972?"

Similarly:

"How much did we spend on entertainment last year?"

Thus, the system must know that doing certain verbs gives rise to certain nouns.

Another kind of complexity arises when subsidiary processing is necessary to determine the key specifications. This kind of information is often contained in relative clauses.

"Show me the product mix for all plants whose profitability decreased last year."

"What is the ratio of sales of items whose margins are $2.0 to those whose margins are $1.5?"

"Show me all the costs associated with product 1."

In some cases the manner of presentation of the data appears as the main noun group in the request.

"Give me a table of sales for each product."

"Show me a plot of sales versus overhead cost for each of the last five years."

The naming system has to do some extra work on such constructions to obtain the data name from the prepositional group. Other indications of manner also occur:

"What was the product mix in terms of sales dollars in 1972?"

These forms are used mainly to specify mathematical functions of data and are discussed in section 7.6.

An interesting and important problem arises, however, when the user specifies a key value that is not applicable to the data requested. For example, in solving the experimental problem for the battery manufacturing company, a subject may ask for overhead costs by product. The database, however, does not contain overheads allocated by product. This problem is serious because it has a bearing on the subject's understanding of the data base and contributes to the "model of the world problem" discussed in the next chapter. The subject must, of course, be told that overheads are not allocated by product. It may also be desirable to give him some information that explains the structure of the database and the rationale behind it. On no account must the aberrant specification be ignored. In fact, as a general rule, no part of the input request should be ignored.

If the naming mechanism cannot establish a correspondence between the noun group whose value is requested and a known data item, then the system may not have that data or it may be present in other forms, perhaps as a property of some entity. This should be checked before the user is sent a message saying "Sorry I do not have any information about ---".

A few subjects asked for the balance sheet and the income statement for the corporation. Similarly subjects asked for "cost attributable to each product". This kind of question seems to indicate the need for sets of data grouped together in terms of a common denominator or for a particular problem. This will be discussed in more detail in Chapters 8 and 9.

7.4 PROPERTIES OF ENTITIES AND IDENTITY QUESTIONS

As we mentioned earlier, properties of entities are different from aggregate data in that they are attached to the entities rather than stored independently. To retrieve properties the entity must be isolated first. It is often contained in the agent case, the possessor case of a "have" verb, or sometimes in a prepositional group starting with "of". Once the entity is isolated and the property determined the system can look it up in the knowledge associated with the entity. This may yield a value or it may yield a pointer to a data item which can be retrieved as described above.

Thus, the price of lead may be contained in the knowledge associated with lead. The entry may be lead-price which gives the name of the data item whose value is required. This technique can be used to set up knowledge to answer questions like:

"How much did Sears buy from us?"

In the knowledge for Sears, under the name "buy" we can insert the entry "sales". Thus, we know that value of purchases by Sears will be found under the names "sales".

Some requests seem, on the surface, to ask for aggregate data but, in fact, ask for properties or some function thereof.

"What is the location of plant 3?"

"How may plants are there?"

Identity questions are considerably more difficult to answer. These start with "who" or "which" and require as an answer the identity or identities of the object(s) that satisfy the properties stated in the question. The process of answering these questions can be likened to answering a set of yes-no questions on a set of candidate objects that are capable of having the stated properties. Questions starting with "who" ask for the identity of animate objects while those starting with "which" can ask for either animate or inanimate identities.

Answering routines for identity questions, therefore, start with the selection of a set of candidate objects. The generic name for this set is invariably specified as the main object noun in identity questions with "be" verbs. The candidate set is the set of all objects that are "a kind of" the generic name.

Questions starting with "who" that ask for the identity of the agent of an event usually do not provide direct clues to the candidate set of objects. The set of animate objects is fairly small in many corporate data bases, however, and so a search through all of them may not be very time-consuming. There is, however, the requirement that the objects in the candidate set must be able to perform the given event. This can often be used to narrow down the candidate set from the set of all animate objects.

Once the candidate set is established, the selection process is much like performing a yes-no question test on each event. Understanding and operationalizing the selection criteria may, however, require special pieces of knowledge and complex deduction rules. For example, the question:

"Who is our largest customer?"

does not ask for the largest corporation who is our customer but, rather, for the customer who bought the most from us.

Once the candidate set is known and the selection criteria established, each member of the candidate set can be tested by the criteria. The result of each test is the identity of the candidate or "no". The final response can be created from the results of these tests. If no member of the candidate set passes the test then the appropriate answer is "None of them" for "which" questions and "No one" for "who" questions.

7.5 MODELS

In the management literature, the word "model" has a wide range of meanings. At the simplest level, models can be functions of stored data. For example, profit can be defined as the difference betwen total costs and total revenues. Similarly, contribution margin may be defined as the difference between unit price and unit cost. As mentioned earlier, this amounts to an alternative way of organizing the data base. Thus, managers refer to such models exactly as if they were data. The name of the model becomes associated with the function and the input data required. This information must, therefore, be available to the system as properties of the model. Key information relevant to the retrieval of input data must, however, be specified in the question.

Chapter 3 has described the manner in which requests for model values are analyzed and executed. To accomplish this the system must associate each model with a list of inputs and a subroutine that acts on the inputs and evaluates the model.

In addition to an input list and a subroutine, the knowledge about a model, like knowledge about data items, also contains information as to where key variable values for each key associated with each of its inputs can be found in the sentence. The model value can, however, be produced for only those keys that are common to all its inputs. For example, profit cannot be calculated by product since one of its inputs, overhead cost, does not have it as a key. It can, however, be calculated by plant since each of its inputs is stored by plant.

Information about the key specifications for which model values can or cannot be produced should be known explicitly to the system so that it can answer questions like:

"What is the definition of profit for a product?"

"Can you compute a profit figure for a specific product at a specific plant?"

The mechanism described in Chapter 3 is able to evaluate models that are functions of existing data. There is however a fundamental difference between:

"What was profit in 1973?"

"What will profit be in 1974?"

The latter is really a completely different type of model. It may use existing data as input, but, typically, it is also parameterized on assumptions about the future. Consider:

"What would profit be if sales were $55 million?"

The system can assume that costs would stay the same as the last year, but this is clearly unrealistic, especially for direct cost. Thus, the system may assume that if costs are not specified, overheads stay the same and direct production costs stay the same percentage to sales. This may be realistic but different from what the user had in mind so it must be pointed out and the user given a chance to ask the question again with different assumptions. In fact a good response to the above question may be of the following form:

```
Assumptions:
        Overhead cost:       As in 1973
                             $5.27 million
        Production cost:     Same percentage to sales
                             as in 1973
                             $43.78 million
Profit:                      $5.95 million
```

The user may then ask:

"What would sales be if overhead cost increased by 10% over 1973 and sales were $55 million?"

and so on. The understanding and processing of assumptions can be very complex, however, and may need to be specialized for each forecasting model.

If an unfamiliar user were to be permitted to use forecasting models of reasonable complexity in conversational mode he would have to go through a trial and error learning stage during which he learnt the parameters of the model and learned to specify them correctly. This can be avoided by moving him into a special sub-system, possibly tailored to the model, in which his interaction is structured and he is asked for model assumptions in a formal, systematic manner.

Another large and powerful class of models are those that can be individually specified and fitted to the user's needs. (See Urban [66,67].) These are also best served by leading the user into a special model sub-system in which he can parameterize, test and run the model. (See also Krumland [35].)

7.6 FUNCTIONS OF DATA

Next to data and model values, functions of these values figure most frequently in the requests made by the subjects. Functions differ from models in that the inputs are specified in the request rather than encapsulated in knowledge associated with the model name. Thus, profit could also be obtained by asking:

> "What is the difference between total revenue
> and total cost?"

Typical functions requested by the subjects were "difference", "increase", "ratio", "percentage" and "variance". In the simplest form they are asked for by name and the inputs specified in preposition groups:

"What is the percentage of overhead costs to sales in 1972?"

"What is the distribution of sales by product?"

There are, however more complex forms:

"What is the variance between actual and budgeted cost?"

"What was the sales increase in 1973?"

"What percentage of capacity is idle?"

"What percentage of capacity is being utilized?"

A considerable amount of specialized knowledge is needed to interpret such requests correctly and decide on the data items to be retrieved and fed into the function subroutine. Consider "distribution". The system must know that this refers to the way in which an aggregate data item is distributed along one of its keys. Thus, the request should be expected to provide a data name and a key. The data name is usually preceded by "of" and the key by "with" or "by". Once this information is extracted from the sentence it can be sent to the general control program that is responsible for computing the result. This checks if the data is indeed aggregated by the key, providing a suitable response if it is not, retrieves the data and computes the distribution.

In the same way, "percentage" must have special knowledge associated with it. The system must know that it requires two inputs one of which may be a subset of the other. This permits analysis of following forms:

"What is the percentage of operating costs in overheads?"

"What percentage of plant capacity is utilized?"

Similarly, "increase" must have knowledge associated with it to indicate that it is often applied to the same data item at two periods of time. This allows sentences like:

"What was the increase in overhead cost in 1973?"

"What was the increase in overhead cost over 1972?"

to be interpreted correctly and to lead to the same response.

Another method of specifying functions is to use a prepositional group to specify the manner in which the answer should be calculated.

"Show me overhead costs as a percentage of sales for the last two years."

"Give me the production distributed by product."

"Show me overhead costs divided by sales for the last two years."

The preposition "as" and the past participles used in such sentences indicate that the answer requires special processing. Again, knowledge about the function must be used to fill its input slots before the answer can be computed.

Thus, each function has to have knowledge associated with it of the type of inputs it expects and where to find them in the parse of the input sentence as well as a subroutine to calculate the results.

Functions can, of course, be concatenated:

"What was the percentage increase in sales by product in 1973?"

"What is the average variance of production cost for each plant?"

The general control program responsible for invoking the routines that calculate the required functions, providing inputs to them and formatting the results also processes concatenated functions. If the inputs to a function, such as percentage, are two equally large sets of numbers the program invokes the percentage function once for each pair of corresponding numbers from each set. Thus, it takes care of cases where the function has to be invoked more than once for inputs other than single numbers.

Finally, mathematical functions such as multiply and divide can be used as verbs and participles in sentences.

"Divide overhead cost by sales."

"Show overhead costs divided by sales for each plant."

The system must be able to analyze such sentences and set up routines to perform the necessary calculations.

7.7 YES-NO QUESTIONS

Basically, yes-no questions ask if a situation or relationship is true. They are used by the subjects to learn about the corporation and the system and sometimes to ask if specific sub-problems exist. The percentage of yes-no questions in this sample is somewhat higher than would be expected from routine users of such a system because they would have less need to learn about the corporation and the system. In general, they will have less need to build models of the situation. (In terms of the frame theory described in the next chapter they will not build "frames" so much as exercise them.)

Since yes-no questions are used to build situation models it is very important that they be analyzed and answered, and answered correctly.

Yes-no questions that inquire about the corporation need, basically, a good model of the corporation encoded in the knowledge-base to interpret and answer them. This model should contain knowledge about each entity in the corporation, such as plants, products, customers, the system, the corporation and the user. The knowledge base should contain the properties of each entity and the relationship of each entity to other entities. In addition to this basic knowledge the system should be able to operationalize the meanings of the various concepts that may be used to test the properties of the system. These may have to be translated into deduction rules that operate on known properties of entities. A simple example is:

"Do we have five plants?"

The system should be able to respond to this question by associating the "we" with the corporation, looking at the list of entities pointed to by the "have" pointer of the corporation and counting those that can be described by the term "plant". Similarly, it must know what "repeat customers" means, and if this is contained in a question it should invoke a deduction rule to check the customer lists for the appropriate periods and find the common subset.

The other class of yes-no questions about the corporation ask directly if a certain sub-problem that could contribute to a problem situation exists. (Analysis of a problem into potential sub-problems is explored in more detail in Chapter 8.) Typically, these concern changes or deviations from the previous year or from budget, plan, forecast, or expectation. Questions about these sub-problems can be divided into two distinct categories according to whether the criteria that makes a change significant are specified or not.

"Did overheads increase by more than 5% for any plant?"

This is a variation of the more popular identity question form:

"For which plants did overheads increase by more than 5%?"

If answered in the affirmative, this kind of question is usually followed by a question like "For which ones". Thus, the identity list created in answering the question should be stored until at least the next question.

In certain questions the discriminating criteria for establishing whether a sub-problem exists are stated in fuzzy terms:

"Were profit margins maintained last year?"

"Did the product mix change in any plant?"

"Have lead prices fluctuated in the last two years?"

In these cases the words "maintained", "change" and "fluctuated" must be given a more specific meaning before the system can fashion an appropriate response. These general words cannot, however, be given meaning by the system. The only response that the system can provide, therefore, is to present the data (if the relevant data can be deduced from the question and if it is available) and let the user draw his own conclusions. It seems better to respond in this way rather than to ask him to clarify the question.

7.8 CAUSALITY

In discussing the behavioral reactions of people to the system, we mentioned that some of them started by assuming only minimal capabilities and by asking progressively more difficult questions to explore its scope and power. As they gained confidence in the system they started treating it like a human being. To some extent, this was the intent of the experiment, but it led to subjects asking about causes and motivations. In general, such questions cannot be answered:

"Why was there such a great increase in operating cost in plant 0?"

"Products 4 and 5 show the greatest deviation of selling price from list price, why?"

"Do you have information what these loans are for?"

In some cases, however, the reasons for certain actions are clear and well defined and could be entered into the database as causal properties of certain entities. Some of the equipment could have a property, for example, that said "Reason for purchase: Depreciation benefits". This kind of augmented data base would be much more powerful for problem-solving.

A few of the questions that ask for the reasons behind some action can be responded to by supplying data. This is the case in the last example which can be answered by providing the sources and uses of funds.

7.9 MODEL DEFINITIONS

One of the more powerful uses to which the system can be put is to define new models or new functions of existing data. This is done in order to test hypotheses regarding potential problems, test proposed policy alternatives or create quantities that the user is familiar with and that fit better into his model of the situation..

Specifications for models defined as functions of existing data are fairly simple. For example, "contribution margin", abbreviated as "margin", has the following pieces of knowledge associated with it in the prototype system:

```
(LEARN (CONTEXT MARGIN PRODUCT)
       (DURING MARGIN TIME-INTERVAL)
       (ARG-LIST MARGIN (LIST-PRICE STANDARD-COST))
       (TYPE MARGIN MODEL))

(DEF MARGIN
       (SAY CONTRIBUTION MARGIN IS THE DIFFERENCE BETWEEN
                                 STANDARD COST AND LIST PRICE))

(DEFUN MARGIN (A B) (*DIF A B))
```

The first expression states that margin is a model, its inputs are list-price and standard-cost and lists the cases in the parse where the key specifications for retrieving the inputs can be found. The second expression is a definition of contribution margin and the third the function that computes its value. In addition to this information there are some default rules for retrieving data in the absence of complete key information. These are contained in the retrieval programs of the prototype system but there is no reason why they could not be encoded efficiently and compactly along with the above specifications. Thus, while the specifications of each model are quite simple, they involve some knowledge that the user does not have.

Creation of model specifications from sentential input is very difficult. Often, more than one sentence is used to define the model and the construction of the sentences is usually very complex. In addition, a large amount of knowledge about the corporation and its environment is necessary if more powerful models are to be defined. In general, the problem of learning from an unstructured dialog with the user needs to be solved. (See Schank [59], Abelson [1], Charniak [15].) This is very difficult and it seems that building models conversationally, as part of a problem-solving dialog, is somewhat beyond the current state of the art.

Since models are extremely important to the problem-solving process, model definition facilities must be provided in the system. The best solution, given the state of the art, seems to be to lead the user into a sub-system whenever he attempts to define a model. Inside the sub-system he can participate in a structured dialog to define the model. The sub-system may know about various model types and the interaction may be tailored to the type of model the user wishes to work with.

7.10 REQUESTS ABOUT THE CAPABILITY OF THE SYSTEM

At the simplest level, questions about the capability of the system can be answered by using the list of entities under the "does" property of the system with suitable synonym and interpretation capabilities.

"Can you perform mathematical computations?"

"Can you format reports?"

"List all the functions you can perform."

The more difficult questions of this genre involve the capabilities of a model or the details of the data available.

"Can you produce a profit figure for each product at a specific plant?"

"Can you give me data on product mix from each plant?"

These questions have to be answered from the knowledge associated with each model and with each type of data. (See also section III.4.)

7.11 REGARDING CONTENTS OF THE DATABASE

This is the most frequent class of questions about the system. Model names are, of course, used interchangeably with data names.

"Do you have variable budgets?"

"Do you have any information on customer satisfaction?"

In some cases the question is phrased as a yes-no question but provides complete specifications for retrieving the data.

"Can you show me the overhead cost for each plant?"

The subject seems to have little doubt that the data exists and he is asking for its value. In these cases the the data should be provided. If the information in the sentence is inadequate the definition of the data and its principal characteristics should be printed out.

7.12 REGARDING COMPOSITION OF DATA

These questions are asked to determine the composition of various types of data so as to construct suitable models of the world. They are relatively simple to answer, however. The only knowledge needed to respond to questions such as:

"What makes up overhead costs?"

"Where do operating costs get included?"

is knowledge of the components of each type of cost and the ability to go either up or down the component tree.

Questions like:

"Are transportation costs included in overheads?"

should either be answered with "Yes" or with "No, they are included in cost of goods sold". In general, a negative response to a yes-no question should attempt to provide as much information as possible about the state of the world.

7.13 DEFINITIONS

Requests for definitions of various items contained in system are very similar to the above class of questions. The system should respond to them with pre-written pieces of text. If the request is not for the definition per se but requires definitional information then there seems to be a case for selecting from a set of definitions depending on the question and the context. For example,

"How are overhead costs defined?"

asks for the definition of overhead costs which includes their nature and how they are stored in the system. On the other hand,

"Do you have overhead costs by product?"

merely asks whether overheads are allocated by product and perhaps the rationale behind the decision.

The system should also contain information on the nature of various costs and how they vary. This allows it to answer questions like:

"Do overhead costs vary with volume?"

The knowledge associated with a model should also include the nature of the output it produces and the inputs it requires. This enables the system to answer questions like:

"Do you have a forecasting model for demand?"

"Do you have a model for measuring customer satisfaction?"

"Do you have a model for maximizing contribution subject to production and other constraints?"

7.14 PRONOUN REFERENCE, ANAPHORIC REFERENCE AND ELLIPSES

Any system that purports to allow convenient conversational interaction in English must be able to deal with pronoun and anaphoric reference and ellipses. These are used by subjects for brevity and conciseness. Consider for example, the following sets of questions:

1. a. "What was the profit for 1971?"

   b. "What was it for '72?"

2. a. "What were product 4 sales to Sears in '71?"

   b. "What were they for product 2?"

   c. "For product 5?"

3. a. "Did overheads at plant 1 exceed budget?"

   b. "What about plant 2?"

4. a. "Did anyone buy more than $100,000 worth last year?"

   b. "Who did?", or merely, "Who?"

In each of these sets of questions, a later question omits some of the information provided in the earlier question. In 1b. the pronoun "it" has to be correctly assigned to profit. In 2b. and 2c. the unit number has to replaced in the framework of 2a. to obtain the full question. Alternatively, the missing cases have to be supplied from 2a. In 3b. only the plant specification is changed. The rest of the information has to be supplied from 3a. Question 4b. is clearly an identity question except that the function is unspecified. Thus, 4a. must be used to fill out the cases that specify the function.

In the same way the second sentence in a pair of conjoined sentences may refer to information supplied in the first one.

"Did overheads exceed budget in plant 1
and by how much?"

Most of the reference devices used in the subjects' sentences can be handled by storing the parse of the last complete sentence that was received as well as its main noun group. The noun group can be used for pronoun reference as in 1b. and the parse can be used to create complete requests from partial, incremental information as in 2b, 2c, and 3b. We mentioned earlier that the identity list of identity questions should be maintained for at least one sentence to be able to respond to questions similar to 3b.

This basic strategy can be extended to include complex cases like:

"Give me the same figures for 1973."

"Substitute "direct manufacturing cost" for "cost of production" in the previous input."

Some subjects attempt to set specifications that should be followed for the next few questions. This can also be mechanised in the above manner except that the stored specifications must be used for every request until reset.

References and responses to questions asked by the system can be deciphered by keeping a parse of the questions and their main noun groups. Processing these sentences can be simplified by the system phrasing its questions in such a way that only a limited number of responses are possible.

Anaphoric reference and ellipses are common and powerful devices. Thirty seven (7.4 %) of the subject's requests made use of them. There are, however, a large variety of anaphoric devices that are used in English. The mechanism described above is capable of analyzing the more popular ones and covers a substantial percentage of the cases. The effectiveness of the system should not be seriously diminished by its inability to understand the more obscure ones.

7.15 THE WORLD MODEL PROBLEM

Every data base and every question-answering system embodies a particular model of the world. Further, it expects questions about concepts and properties that are sensible in terms of this model. A severe problem can arise, therefore, if the user has a model of the world which is at variance with that of the system in significant ways. For example, our data base contains direct manufacturing costs and overhead costs for activities that cannot be directly attributed to manufacturing. The overhead costs are not allocated to products, since we feel this to be artificial. Break-even points, therefore, have no meaning in the system. A user who is accustomed to thinking in terms of break-even quantities will ask for them and may not be able to proceed if the system merely says it cannot provide break-even data.

Ideally, the system should be able to realize that there is a discrepancy between world models and since it cannot change its model of the world it should explain it to the user to try to influence his.

The prototype system adopts an extremely simple-minded approach to this problem. It maintains a list of concepts, such as break-even, that it knows belong to variant models of the world. Every time a question asks for one of these concepts it responds by printing out an explanation of its world model and why the question is inappropriate.

In some cases subjects defined models to bring the world model of the system closer to their own. For example, a subject who is accustomed to working with total cost rather than with production cost and overheads separately may:

> "Define p-cost to be the sum of production cost and overhead cost."

The difference in world models may go unnoticed sometimes and a user may misinterpret the data and come to the wrong conclusions. For example, a user may work with production cost assuming that it includes allocated overheads. This may lead him to conclude that cost increases were not a problem when, in fact, overhead cost increases were seriously depressing profits.

7.16 CONCLUSION

The knowledge required for an intelligent management-support system is large and varied. In the prototype system knowledge about data, models and entities is encoded in OWL and stored in descriptive form. Knowledge about functions of data, deduction rules and how to answer different types of questions is encoded as procedure.

Appendix III describes the corpus of knowledge required to respond to the requests that were made by the subjects in the course of the experiment. We find that even though the amount of knowledge required is rather large and of many types, it takes only about thirty pages of text to describe. Thus, it is not intractable to incorporate it into a knowledge-based system. A plan for doing so can be formed based upon the above analysis

CHAPTER 8

ANALYSIS OF PROBLEM-SOLVING BEHAVIOR

8.1 INTRODUCTION

This chapter analyzes the problem-solving behavior exhibited by subjects in solving the experimental problem described in Appendix 1. The analysis is made in terms of a "frame" oriented paradigm of coming to grips with problem situations. The paradigm is developed in this chapter and we attempt to analyze the subjects' problem-solving protocols in terms of the processes postulated in the paradigm. We then argue that a successful management-support system must be able to support these general processes and one that does will be suitable for a wide range of management problems.

A number of models have been proposed for problem-solving behavior (See Simon [62] and Gore [22].) but none of them have proved very successful in explaining the specifics of solving a particular problem. The "frame" model attempts to do this in terms of special structures and processes used by human beings to comprehend situations. Frames are mental constructs introduced by Minsky [47] but with roots that go back to Gestalt psychology. (See, for example, Wertheimer [69].) We believe that frame operations can be used to understand the mechanics of coming to grips with certain kinds of management situations. The basic process consists of mapping an existing situation into general models that encapsulate the essential features of such situations. Details of the mapping characterize the situation and lead to the decisions, or more realistically, the action plan.

One of the most important functions of a manager is to explore a situation where a problem exists and isolate the detailed causes of the problem. We call this problem diagnosis: the isolation of specific problems from gross symptoms. Problem diagnosis can be considered to be one component of the decision-making process but the paradigm shows it to be inextricably linked with another component: the search for solutions.

Consider, for example, the manager who discovers that profits for the last quarter were lower than expected. Starting from such a problem, which may be recognized by any of the means described by Pounds [55], he will usually want to dig deeper and isolate problems at a more detailed level.

Problem diagnosis involves a particular kind of thinking. It is concerned with problems that are generally familiar but are neither routine nor monumental. In other words, they do not have ready-made solutions but there exist general problem-solving processes that are applicable to them. Typically, the manager's level of aspiration is not very important and he does not try to innovate and create new models to solve them. These characteristics are common to a wide range of problem situations. Later sections of this chapter will describe the paradigm in detail and present supporting evidence from the problem-solving protocols of the subjects.

8.2 NEED FOR PROBLEM DIAGNOSIS

The problem diagnosis process can be characterized as the analysis of a problem known to be soluble and for which the solution criteria are predefined. Clearly, the problem diagnosis process is hierarchical. If low profits are analyzed to result from high overheads then high overheads can be analyzed further for even more detailed causes. (There is considerable evidence of problem decomposition and hierarchical processes in problem-finding/problem-solving. See, for instance, Newell and Simon [51].) Thus, starting from gross symptoms the process creates a tree of problems. Each node of the tree is a problem and gives rise to secondary nodes that represent the sub-problems that contribute to it. Since the objective is to decide upon courses of action to alleviate the gross symptoms, the process stops with the isolation of problems that can be influenced directly by decision variables. Thus, the search for detailed problems is influenced by available solutions. Simon's [62] "intelligence" and "design" phases are, therefore, seen to be combined into a single process and the search for problems is intermixed with the search for solutions. The culmination of this process leaves the manager with not only a better understanding of the situation but also with a plan of action geared towards its resolution. Thus, problem diagnosis and problem-solving are seen to be closely related and, in fact, to form a single syndrome.

A problem branch may also terminate due to the inability of the manager to find more detailed problems or to obtain the required information. In general, the exact set of detailed problems at which a tree will terminate will depend not only on the situation but on the manager's perception of it, the mental models and the data he possesses to analyze it and the decision variables available to him. These factors will be explored in later sections of the paper.

8.3 THE STRUCTURE OF MENTAL MODELS

Minsky [47] has postulated a theory that seems to be useful in explaining how people analyze situations and build mental repesentations of them.

> "When one encounters a new situation (or makes a substantial change in one's view of a present problem) one selects from memory a structure called a _Frame_". This is a remembered framework to be adapted to fit reality by changing details as necessary.
>
> A _frame_ is a data-structure for representing a stereotyped situation like being in a certain kind of living room, or going to a child's birthday party. Attached to each frame are several kinds of information. Some of this information is about how to use the frame. Some is about what one can expect to happen next. Some is about what to do if these expectations are not confirmed.

Each frame has a number of terminals for attaching specific information. These can be thought of as slots that can be filled by specific instances or by data. Each terminal has conditions which must be met by its assignments. Thus, the process of filling a frame consists of assigning data values or instances to each of its terminals. The act of filling terminals may, however, invoke frames for the terminal situations and require further assignment of terminal values. Let us consider an example. Minsky quotes a fragment of a children's story:

> There once was a Wolf who saw a Lamb drinking at the river and wanted an excuse to eat it. For that purpose, even though he himself was upstream, he accused the Lamb of stirring up the water and keeping him from drinking. (etc.)

The processes by which we form a mental repesentation of this story seem to be as follows. Reading the first sentence invokes the situation "Wolf wants to eat Lamb". In our mind, there are two frames associated with this situation. One is the real-life frame in which the Wolf catches, kills and eats the Lamb directly. The other is the story-book frame in which some ruse or stratagem is required to trick or trap the Lamb. The next sentence invalidates the real-life frame and confirms the story-book frame. Now, we try and fill the terminal of this frame which asks for the ruse or strategem employed. To do this we invoke a ruse/strategem frame, perhaps a very simple one since we realize that this is a children's story, and try and fill its terminals with the specifics of the ruse.

Ultimately, we analyze the story into a set of frames and terminal values (some of which are filled by default) and create a mental model. This model is our "understanding" of the story and we can use it to answer questions about the story. In fact, the frames and terminals of this model fit into a more general structure of frames and terminals in our mind --- our model of the world --- and we are able to use it to answer questions about the story that go beyond the facts contained in it. Such a question may be "Was the Wolf salivating?" and it will receive an affirmative answer because a default terminal value of the "desire-to-eat" frame is salivation.

An interesting confirmation of frame-oriented thinking comes from an experiment described by Mosher and Hornsby [50] in which children were asked to play a variation of "twenty questions" and determine the cause of an accident -- "A man is driving down the road in his car, the car goes off the road and hits a tree." Asked to describe his "system" for getting the answers an eleven-year-old responded:

> Well, to eliminate big things quickly -- like was there anything wrong with the road -- was there anything wrong with the weather -- was there anything wrong with the car -- was there anything wrong with the person -- if there's something wrong with the person, you start from the bottom and go to the top.
>
> I group like all the things with weather, breaking (sic), then I group them smaller and smaller till I get to the point.

It seems clear that the description of the accident causes him to invoke an analysis frame that contains four potential causes -- the road, the weather, the car and the person. He intends to explore each of these until he finds the applicable one. Then, he plans to investigate it further.

The top level of his analysis frame may, therefore, be diagrammed as below. It is reasonable to postulate that he also has frames for each of the terminal causes which allow him to carry the analysis further.

```
                        ACCIDENT
                  (Going off the road)
                           I
--------------------------------------------------
  I               I               I               I
 ROAD          WEATHER           CAR           PERSON
PROBLEM        PROBLEM         PROBLEM         PROBLEM
```

Thus, the potential causes in the frame are used to form the branches of the problem tree. The next level of the problem tree is built up by invoking and filling frames for the applicable problems.

8.4 PROCESSES IN SITUATION ANALYSIS AND COMPREHENSION

The above examples suggest the following processes in situation comprehension:

1. Invocation of frames from named concepts.
2. Selection between competing frames, and frame validation.
3. Assignment of terminals.

A more complete paradigm of situation comprehension may be described in terms of these processes as follows:

A. Invoke frames.

B. Select between competing frames and validate the selected frame.

C. Attempt to fill each terminal of selected frame. This may require the invocation and filling of frames i.e. steps A. to E. This makes the process recursive.

D. Are all terminals filled?

If so, proceed to E.

If not, use characteristics of unfilled terminals to A.

E. Have all important facts been considered?

If so, proceed to F.

If not, use unaccounted facts to A.

F. Fit frames together to create an internal representation of the problem.

The above paradigm implies the ability to invoke new frames that differ from given frames according to specified terminal characteristics. This is used in a feedback mechanism that matches the given situation against a structure built from the manager's mental frames. In certain cases, appropriate frames may not exist in the manager's mind and, within certain limitations, new frames may be created. Often, these are created incrementally from old frames or hierarachically from more powerful concepts.

Basically, frames are used for situation comprehension; a complex form of what Bruner [9] calls "categorizing". Once this is accomplished and an internal representation created, it dictates the implications and the cause-effect relations to be used in drawing conclusions and in the search for action alternatives. (See also Note 3.)

A variety of frames and other mental constructs may arise from a concept such as "profit". There are frames that analyze the reasons for low and high profit and probably others for more specific situations. In addition, there are cause-effect relations that say "Profit increases if revenues increase." and "Profit decreases if costs increase." etc. Above these is the master concept that "profit is the difference between revenue and cost".

The master "profit" concept exists in the background and seems relatively difficult to work with directly. Its utility lies in the creation of new frames and cause-effect relations and as a place to hold pointers to more specific frames. Some frames, such as low-profit, have loose evaluative judgements, such as "bad", associated with them.

If managers do think in terms of frames then a management-support system should support frame processes that allow him to categorize today's problem in terms of known frames. The following section analyzes the processes of comprehending problem situations and finding solutions in more detail. A subsequent section analyzes the requirements that supporting these processes places on the system.

8.5 PROCESSES IN PROBLEM DIAGNOSIS

The general paradigm for situation comprehension described above can be refined to elucidate the detailed processes used in problem diagnosis. Situation comprehension frames can take on a variety of structures e.g. a description frame for a person may contain his job, his hobbies, his age and his net worth as terminals. In problem diagnosis the analysis frames consist of a list of sub-problem terminals each of which may or may not exist in the given situation. The process of problem diagnosis may be described as follows:

A. The manager considers whether he can take a decision that will solve the problem directly, i.e. if a decision frame exists for the problem (sub-problem).
If so, he will attempt to assign terminals to it. These may be inputs to a decision rule.
If this is successful the problem branch will terminate.

B. If a decision frame does not exist or cannot be filled adequately he will invoke one or more analysis frames for the problem (sub-problem).

C. Validating questions will be asked to eliminate some of the analysis frames. Typically, these are yes-no questions.

D. Each analysis frame will contain potential sub-problems as terminals. The manager will attempt to fill these terminals i.e. test each sub-problem to see if it exists. Each sub-problem that exists will be analyzed further starting at A. Competing analysis frames may also be eliminated at this stage.

E. In rare cases alternative branches of the problem tree will be created and alternative analysis frames pursued further and eliminated on the basis of information obtained at a lower level.

F. Decision frames selected for the sub-problems will constitute the "decision" or the action plan. Occasionally, the manager will ask a few questions at the end of the session to test the decision frames.

Thus, each problem is either attacked directly by a decision frame or analyzed into sub-problems using its analysis frame. In practice, there is another important method of dealing with problem branches --- they may be held in abeyance. This may be done to gather more information or to consult someone who has special knowledge i.e. can bring more refined frames to bear on the problem. Sub-problems may also be (temporarily) abandoned in favor of more promising branches. Thus, branches of the problem tree may end in a "wait" state. Finally, because of cognitive limitations, the manager may forget or ignore certain sub-problems and/or terminals.

Simon [62] models decision-making as a three stage process:

> The first phase of the decision-making process --- searching the environment for conditions calling for decision --- I shall call _intelligence_ activity (borrowing the military meaning of intelligence). The second phase --- inventing, developing and analysing possible courses of action --- I shall call _design_ activity. The third phase --- selecting a particular course of action from those available --- I shall call _choice_ activity.

Our model starts with the existence of a gross symptom that indicates an undesirable situation and necessitates a search for solutions. In one sense then, the intelligence activity precedes our model and terminates with the recognition of the gross symptom. In another sense, however, additional intelligence activity is required because the gross symptom can rarely be alleviated directly with decisions. Thus, further intelligence is needed to isolate the detailed sources of the problem that can be attacked directly by decisions. In this sense our model can be considered an elaboration of the intelligence phase. Design and choice activity succeeds this phase for each of the detailed problems that are isolated.

From another perspective, however, the processes in the model can be considered to be the search for a solution to the gross problem. The design and choice phases seem to take place in frame analysis and are dependent on the nature and the structure of the frames. A better characterization of the model seems to be the reaching of an understanding of a problem situation --- the factors that are important, those over which the manager has some control and others for which he needs more information. This seems to be closer to actual managerial behavior, for certain problem types, than the more general intelligence, design, choice syndrome. Further, the frame model is capable of explaining the details of the analysis in terms of the manager's conceptual structures.

The frame model seems, in fact, to be closer to the heuristic model advocated by Gore [22]. Gore's model has a much broader scope but the early phases of frame invocation and validation seem to correspond to his development of the "Orientation Set" and the later phases of hierarchical analysis of problems and the selection of decisions to the development of the "Evaluation Set".

8.6 EXPLORATION OF THE FRAME MODEL

Malhotra [41] describes two problem-solving protocols that substantiate the frame paradigm. This section presents further supporting evidence. Chapter 4 decribes an experiment in which the subject, in the role of the president of The Battery Company, is confronted with the fact that although sales increased last year profits decreased. He is asked to try and reach an understanding of the situation and recommend a course of action with the assistance of the perfect English language system.

We shall start by describing the behavior of two subjects whose protocols follow the basic paradigm. Later sections will consider variations in the paradigm to accomodate special circumstances. The protocols are slightly edited to eliminate non-problem-solving interaction with the system. Complete protocols are listed in Appendix II.

Subject 18

The subject was a production manager with ten years of line experience and two years of staff experience. He starts off by asking three basic, validating questions to get the feel of the system and his bearings about the company.

1. What was total revenue for the company?
2. What was the cost of goods sold?
3. What was the net income?

He now proceeds to test the first terminal of his frame; whether production costs have increased.

4. What is the unit cost for each product in each plant?

He is told that unit costs are the same in each plant and provided with the unit cost at each plant.

5. What was the actual unit cost change per product

in 1973 over 1972?

Instead of first determining whether there was an increase in unit costs and then breaking this down by plant and by product he investigates the unit cost situation completely. This done, he moves on to his next terminal.

6. How much did overhead cost increase in 1972 over 1973 in each plant?

Again he investigates overhead cost problems completely before moving on.

7. What was the volume increase per product in 1973 over 1972?

After this, he says he has a plan of action but asks two further questions before closing the session.

8. Who are my customers and what is their volume per customer?

9. What is the price of each product?

His action plan was "I would increase price on products 1 and 2. I would also reduce overheads in plant 2 and 4 and in headquarters." This follows from the fact that products 1 and 2 had the highest volumes and plants 2, 4 and headquarters had the highest increase in overhead cost.

Subject 18 seems to use the set of frames pictured in Figure 8.1 to solve the problem. He is an experienced manager and his frames are clean and well defined. He explores each

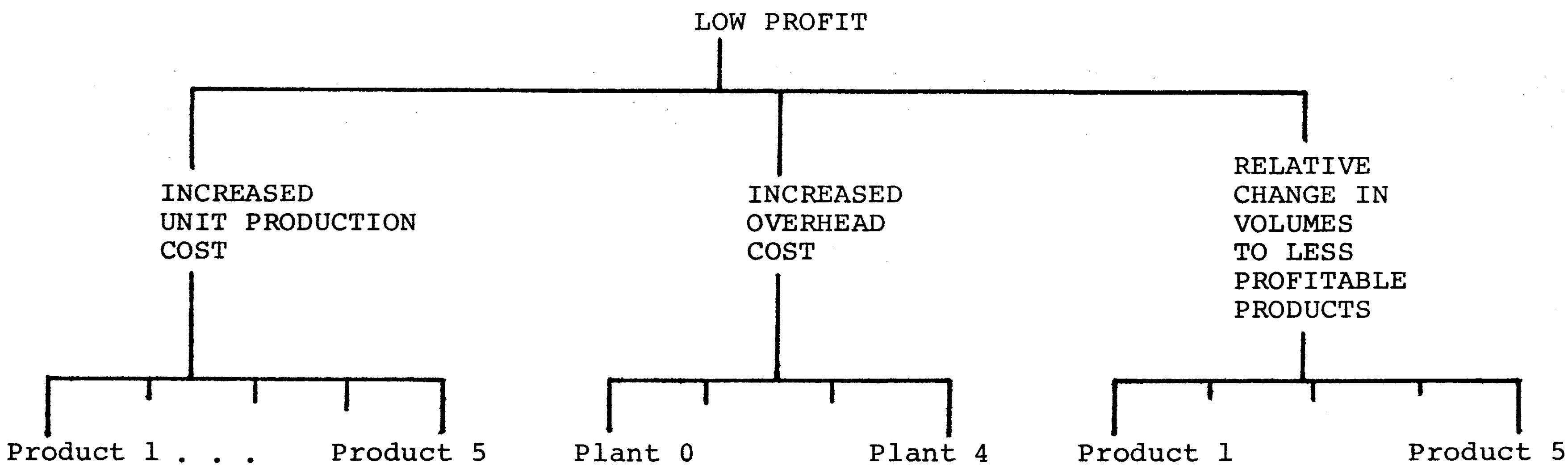


Figure 8.1 Frames for Subject 18

sub-problem terminal in depth before he proceeds to the next one. The question about customers seems to be a general validating question to see if there is anything of interest there. It is not explored further. The final question seems to be asked to test an action recommendation: to ascertain if the prices of products 1 and 2 could be raised i.e. they were not too high already.

The next protocol also fits in with the basic structure of the paradigm and seems to indicate that the subject is thinking in terms of frames.

Subject 23

The subject was an engineer with a degree in Operations Research and ten years of experience as a production engineer. He was taking his first, formal management courses at the time of the experiment.

He starts off by asking a few questions to get the feel of the system and its capabilities.

1. What types of data do you have?
2. Is revenue recorded by product?
3. What are revenues for each product?
4. What are sales by plant?

Now he moves into his first terminal which seems to be a shift of sales between products or plants.

5. What are sales by plant by product?

6. Subtract 1972 sales by plant by product from 1973 sales by plant by product.

He now starts on his second terminal; increase in production cost.

7. Did any product cost exceed budget in 1973?

8. By plant by product which cost exceeded budget?

9. Which product of the five had the largest percentage variance?

10. In 1972 which product or products had the largest variances?

The next terminal on his frame seems to be "decreased profit margins".

11. What were 1972 and 1973 profit margins by product?

Next, he investigates changes in unit manufacturing costs.

12. Can you give unit costs by plant by product?

13. What were actual unit costs for plant 2?

14. What were unit costs for 1972?

Finally, he looks into changes in product mix and price problems.

15. What was product mix by percent in '72?

16. What was product mix by percent in '73?

17. What were 1972 and 1973

prices for each product?

His analysis was:

> "The prices in 1973 have not kept pace with the rise in costs. The roughly five percent rise in prices has not equalled an approximately seven percent hike in costs. The pricing should cover the costs but since all products in all plants were over budget one must assume that something common to all like overhead is the culprit. Institute tighter internal cost controls."

The subject's frames are illustrated in Figure 8.2. We notice his frames are wider than for subject 18 and he investigates production cost and unit production cost separately. This is somewhat inefficient as he seems to be attempting to get at overhead through production cost. Questions 8, 9 and 10 are interesting examples of the use of "which" questions to pinpoint the exact location of the problem. Like subject 18, subject 23 also investigates each terminal in depth before moving on to the next. This may be because the system makes it very easy to ask for data by plant and by product. There are, however, subjects who go completely across the top level of their frames before investigating any of the operant sub-problems in detail. The next protocol does this for overhead cost.

One of the more unusual types of terminals that some subjects used can be called "Product Problem" and "Plant Problem". The subject seems to feel that the problem lies in

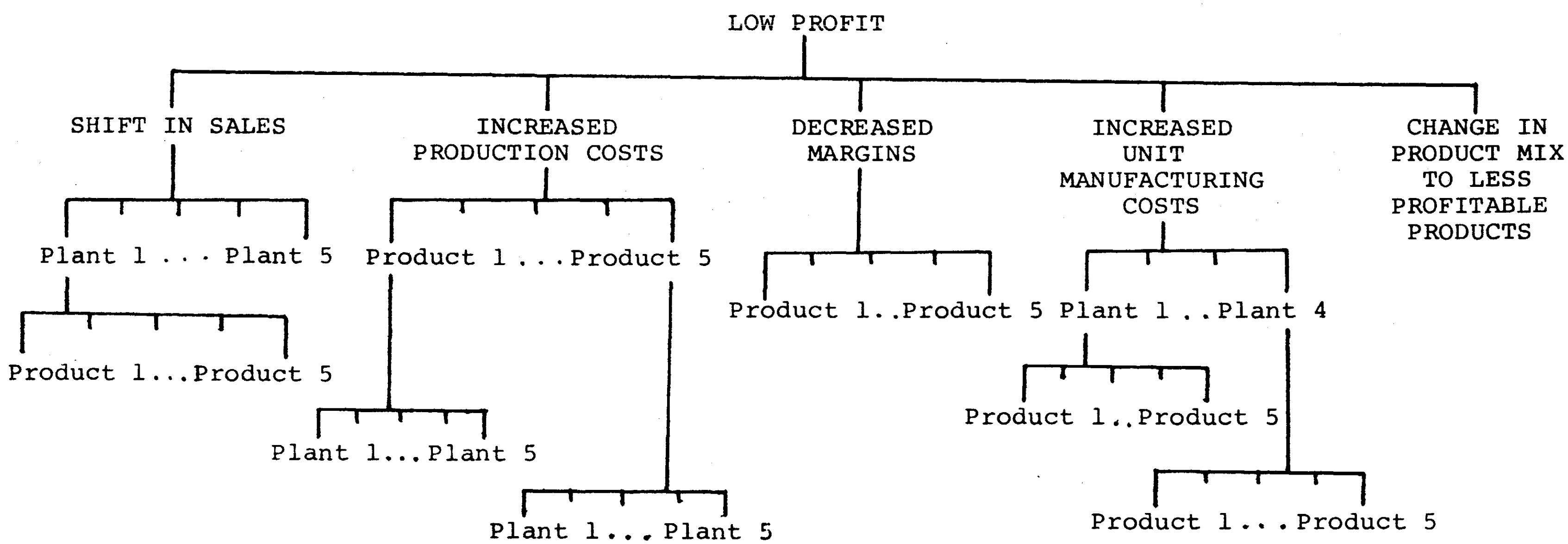


Figure 8.2 Frames for Subject 23

one or more of the products or the plants and proceeds to ask for data to make comparative analyses. This kind of terminal may arise from a view that treats the corporation as being divided in two ways: by plant and by product. The following protocol provides a good example of such terminals. Other protocols show the subject asking a number of questions about one plant or one product as if he was investigating the largest, most important or typical case.

Subject 8

The subject was an industrial enginneer with sixteen years of management experience.

1. What was overhead for 1972?
2. What was the difference in overhead from 1972 to 1973?
3. List the product mix for 1972 and 1973?

These questions test the "High Overheads" and "Change in Mix to Less Profitable Products" terminals of his frame. These seem to be in the nature of validating questions. He decides that overheads are a problem but does not investigate them in detail. He now moves into the "Product Problem" terminal.

4. Give me the profit margin on each product for 1972 and 1973.
5. What is the cost of each product for

1972 and 1973?

6. What were the sales for each product in 1972 and 1973?

Now he goes in to "Plant Problem" and as part of it investigates overhead by plant.

7. Give me the production cost budget for each plant and the overrun if any.
8. What was the overhead budget for each plant?
9. What overhead costs were incurred at each plant?
10. What is the percent overhead overrun at each plant?
11. Give me sales percent increase at each plant for 1973 over 1972?

His action plan was "It appears that plant number 0 and 2 and 4 have excessive non-direct expenses. Rather than increase prices I would pursue a program of cost reduction.". Note that the only two policies he mentions are in the Plant Problem (high overheads) and Product Problem categories (low prices). The frames used by subject 8 are pictured in Figure 8.3

8.7 EXCEPTIONAL CASES

Occasionally, a manager will go across the top-level of his frame and not find a terminal that signals an existing

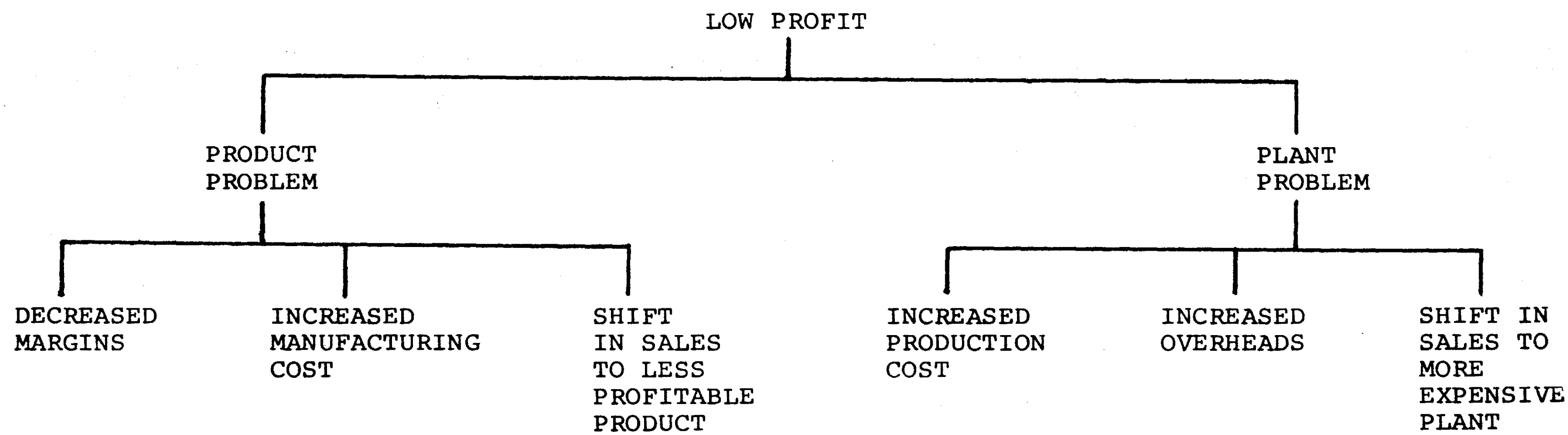


Figure 8.3 Frames for Subject 8

sub-problem. This usually happens when one or more terminals have been checked inappropriately, perhaps because they have the wrong sorts of frames associated with them. Let us consider an example. One of the potential sub-problems for the experimental problem is that overhead costs have increased inordinately. If the user checks this by asking for the percentage of overheads to sales in the last two years the answer is 11.9% and 12.3%. This does not seem to indicate a problem, although overheads have increased by over a million dollars and have seriously affected profits. The problem does not show up because overheads are small compared to sales and sales increased by 20%. In this way, a sub-problem terminal can be wrongly marked as negative. Some subjects were cognizant of this problem and asked for other data to place the change in overheads into perspective. Another reason for not finding a problem is that the subject may be missing an important terminal from his frame.

When the subject goes across the top level of his frame and fails to find a problem he tends to become perplexed and may take a variety of actions. We analyze these strategies in this section.

The first reaction on not finding a problem on the top level of the problem frame is, usually, to go back and recheck all the sub-problems. On this second round the subject may ask for different data and use different discriminant functions on the data to test whether or not the problem exists. For example, in testing whether increased overheads are responsible for the decreased profits, he may now ask for the "increase in overheads in 1972" or "the ratio of overheads to profits in 1972 and 1973". The altered discriminant functions may lead to different conclusions and the subject may find an operative sub-problem.

Another reaction to not finding a problem in the top-level terminals of the frame is to question its structure and try and develop additional terminals for it. This seems to be much more difficult than invoking frames and filling terminals. It manifests itself in a number of validating questions, asked seemingly at random, and it may lead to the creation of new terminals one or more of which may indicate an operative sub-problem.

Finally, the subject may question the problem frame so strongly as to rise above it to the general, more powerful concepts of the profit identity and the cash flow equation and use these to solve the problem.

These three strategies may be called: Retesting, Reformulation and Reconceptualization. Although each strategy is distinctly different in attitude and purpose, the first two often appear intermixed with each other as the subject thrashes about, trying to make sense of the perplexing situation in which there is a problem but he cannot find it. The third strategy seems to appear after the subject has overcome his initial bewilderment at not finding the problem. Frames can be considered to be conceptual hueristics (See Note 4.) that the manager uses to try and get a quick understanding of commonly encountered problem types. If they fail he takes a deep breath and starts again using more rigorous and powerful methods.

The following protocols present examples of these three fall-back strategies.

Subject 3

The subject was a school teacher with no management experience. At the time of the experiment he was a student at the Sloan School of Management. He starts by checking a number of terminals, generally asking one question for each terminal.

1. List sales for product 1 through product 5 for the last 2 years.
2. List prices of single units for both 72 and 73.
3. What was the cost of producing each product for both 1972 and 1973?
4. Did one plant assume more production of batteries

from the other plants?

5. In 1972?

6. What was the rate of increase of shipping cost between 1972 and 1973?

7. Are shipping costs reflected in production cost?

Note that the subject has not asked for overheads. Instead, he asks for shipping costs which he may have assumed to be part of overheads. If, on the other hand, he assumed they were part of production costs then he is rechecking a part of production costs that he has already tested with question 3. In fact, it seems that initially the subject had four terminals on the top level of his frame. He checked these with questions 1 to 4. The following questions recheck these terminals, often in a different way or using some subset of the data. Some of them try out new terminals, thought of on the spur of the moment. Others are validating questions asked to generate new terminals.

8. Do you have information on customer satisfaction?

9. What is the percentage of repeat customers in 1973 and 1972 and in 1971 and 1972?

10. What was the unit price in 1973?

11. What is the percentage of increase of each product for each year studied?

12. What was the percentage increase of profit for

the last five years?

13. What were the overhead costs for the last 5 years?

14. What is list price vs selling price for last 2 years?

15. Product 4 and 5 show the greatest difference between list and quotation prices, why?

16. Do you have a list of changes in sales force for each branch?

The subject seems to sit back and take stock at this point. Overheads seem to be a possibility and he returns to analyze them in greater detail.

17. Give me a breakdown of items in your overhead.

18. Include each of these by plants.

19. Compare overhead costs for the last five years.

20. Do you have further breakdowns of overheads attributable to each product within plants?

He seems to decide that overheads are not a problem. Later he said that the million dollar increase in overheads did not seem "much".

21. What % of each product is sold from each plant for each of the last 5 years?

22. What is the total volume for each plant (sales) for each of the 5 years?

These questions seem to ask for sales by plant and are rechecking the sub-problem tested by question 4. Now he goes back to checking overheads and their components.

23. Compare plant overhead with total overhead for the last 2 years, 1973 and 1972.
24. List increases in overhead for each plant for the last five years.
25. Compare overhead costs for plants for the last 5 years.
26. What are salary increases for each plant for the last two years?
27. List increases in interest costs for the last two years.
28. List inventory of product at end of 1971 and 1972.

The subject abandoned the problem without having reached a satisfactory understanding. This protocol illustrates Rechecking and Reformulation. The following protocol illustrates Reconceptualization.

Subject 11

The subject was the manager of an operations research department with five years of staff experience.

1. What was the % of overhead to sales in each of the

last five years?

2. Were there any changes in the product mix in terms of sales dollars?

3. What were the profit margins of the five batteries in the last two years?

4. What are the handling costs associated with each product and did they change over the last two years.

5. What are the actual selling prices of the five batteries?

He seems to have gone across the top level of his frame and failed to find a problem. In fact, question 5 seems to be a half-hearted stab at testing a new, not quite properly formulated, terminal. He now goes back to the basic cash flow identity.

6. How much was the additional revenue received from the 20% sales increase and where was it spent?

The system could not answer this. Hence:

7. The intent of my question is to find out if you know if your accounting methods can relate the changes in sales to changes in your expense structure.

8. Please give me changes in each type of cost

associated with each product.

9. I would like the cost of each product broken down on a direct and indirect basis.

10. What was the total production cost in each of the two years?

The subject concluded:

> "Your problem is in the area of controlling production costs. You experienced on 8 million dollar increase in sales and a 7 million dollar increase in production cost and some increase in the overhead."

If he had checked the percentage of production cost to sales he would have found that it changed hardly at all and the million dollar increase in overheads accounted almost completely for the decrease in profitability.

In general, the subjects' protocols seem to substantiate the frame paradigm. Analysis of the problem follows their frame structures although it is contaminated by questions asked to learn about the system and validate their final recommendations. Frame analysis seems to be the first level of attack on the problem, if it fails the subjects tend to retest, question or rise beyond their frames to more powerful concepts.

The composition of frames seems to be fairly similar among subjects except that some start with a few general problems and move later to specifics while others check a number of specific problems at the top level. This may be a matter of problem-solving style and the manner in which they organize concepts in their minds.

8.8 COMMENTS ON THE PROTOCOLS

1. Frame structures seem to be very different for different subjects. Some of them, like subject 23, have wide frames with specific terminals. Others, like subject 18 have fewer, more general terminals.
2. Certain terminal values seem to be filled by default and subjects do not seem to bother to ask questions to assign them. For example, they know that management salaries go up every year.
3. The protocols indicate that managers ask general questions (using words like "sufficient" and "maintained") to try and eliminate a sub-problem. If it cannot be eliminated they ask more specific questions to assign terminal values.
4. Most of the questions that attempt to eliminate a sub-problem terminal ask for a piece of data in relation to a norm, plan, standard or history. (See Pounds [55].) Often the relationship is a loose one (expressed by words like "sufficient" and "maintained") such as whether the two numbers are of about the same magnitude or whether one

is greater or lesser than the other. This corresponds to the fact that the terminals are named "Low profit", "High Inventory", etc. The few single numbers that seem to be significant, such as sales, seem to serve as validating questions in that they select frames such as "small-company".

5. The processes of validation, frame creation and terminal assignment do not take place in "logical" sequence. In fact, terminals are left hanging as higher level terminals are filled, frames created, etc. This suggests that all elements of a frame are not retrieved at the same time, but seem to arise in some order of importance. Further, the subjects seem to have special processes that bring in the next terminal and check if all terminals have been filled and if all the given facts are accounted for.

6. Since this was the first pass at the problem a number of branches tended to be left in the "wait" state due to insufficient information or because the subject felt them to be less important relative to other branches. We expect that this will be found to be typical of early problem diagnosis attempts.

7. Similarly, since this is the first time the subjects used the system some of the questions are asked more to test

the system than for their answers. The user is trying to understand the system as well as the problem.

8. In considering the problem-solving behavior exhibited in the protocols we must remember that real-life problem diagnosis is not a one-shot process. Typically, the manager will go through a preliminary session and then repeat the process again using more detailed frames some of which may have been created as a result of earlier analyses. This iterative nature of problem diagnosis is underplayed in the experiment.

8.9 CONCLUSIONS AND OTHER OBSERVATIONS

We find that the model of problem diagnosis and it's variants described above seems to explain protocols quite well. The subjects seem to follow their frames faithfully to test terminals and reach the conclusions they dictate. The frames do seem to provide the basis on which they analyze problems rather than being merely a convenient representation of their problem-solving process. Thus, if frame structures are found to be reasonably stable among managers or if they can be isolated for a group of users then they can serve as a basis for system design. Frame structures may, therefore, provide a powerful empirical methodology for the design of knowledge-based systems.

Other investigations of problem-solving also provide evidence of frame processes. Studies of medical diagnosis by Sussman [65] and Rubin [57] indicate that doctors invoke initial disease frames based on information about age, sex and symptom. They process subsequent information in light of this initial hypothesis and ask questions to test it. The facts may however disconfirm it and invoke a new frame. Sussman describes a situation in which an inexperienced physician starts to describe a case to an experienced physician by reciting a long string of the patient's characteristics. The experienced physician soon grows restless and asks what the symptoms are, presumably because he does not have a framework into which to fit the facts.

The frame structures involved in medical diagnosis are very large, however, and they may get severely truncated due to cognitive limitations. (See Miller [45], Schroder, et al [60].) Doctors tend, therefore, to use formal questionnaires and checklists to make sure nothing is overlooked. Another precaution they take against missing information is investigate a subject completely when they start asking about it rather than merely ask for information to test the frame.

8.9.1 Generality of the Model

We maintain that the process of problem diagnosis is similar from person to person since the protocols display common features of frame invocation, terminal filling, etc. Differences in problem-finding/problem-solving style and effectiveness can be attributed to differences in the number, content and structure of frames for the situation at hand. These differences will determine the importance of various factors in the analysis and the order in which they will be considered. Further, individual cognitive capacities dictate how many frames and terminals will be "overlooked".

An expert may have a thousand detailed frames for every aspect of a situation. Someone with less experience may have only fifteen or twenty. The orientation process for naive users may, therefore, involve a considerable amount of frame creation while an expert may require only validation and would, therefore, be much quicker. This may account for the expert's ability to "get right to the heart of the matter". Thus, the performance of the novice will always be worse; he may be unable to analyze the problem due to inadequate frames and if he tries to develop them he may strain his cognitive capacity and degrade his performance in other tasks.

Since one can be expert only in certain limited areas it follows that the expert's frames will be better in certain areas than in others. This may account in part for the observed phenomenon of selective perception. (See Dearborn and Simon [17].) Malhotra [41] also provides an example of looking at a problem preferentially from one's speciality.

8.9.2 Problem Diagnosis Style

On the simplest level, a manager may decide to fill all the terminals of a problem frame before investigating the next lower level or he may investigate each sub-problem as it arises. He may decide to leave some branches in the "wait" state while he investigates more promising branches. Some of these decisions will be taken according to his a priori evaluation of the importance of various sub-problems while others may be functions of personality variables. Bruner [9] found that the different concept attainment strategies employed by his subjects were consistent features of their personality. This may also be true of problem-finding/problem-solving strategies. (See Note 5.)

8.9.3 Facilities for Frame Analysis

If the frame model is valid for the kinds of problem-solving we have been considering then managers come to grips with problem situations by validating and analyzing frames. Only if this process fails do they try to create or modify their frames. A successful management-support system must, therefore, be able to answer questions that are used to validate frames and test terminals. If frame creation is important in certain situations then it should be able to to support inductive processes that lead to the generation of new frames. The ability of the system to support these processes will determine its success as a management aid. Moreover, <u>the type of problem the system will be best suited for will depend on the types of questions it is best equipped to answer</u>.

Testing whether a terminal sub-problem exists usually involves looking at data or model values, comparing them with a budget, goal or historical norm and testing whether the differences are significant. Thus, the system must be able to retrieve data, evaluate models and compute functions of data.

Terminals can also be tested by asking yes-no questions with fuzzy discriminating function such as:

"Were profit margins maintained this year?"

"Did unit costs change significantly last year?"

In these cases the system should try and recognize the data and the comparison required and present them to the user. He can then use his judgement to decide whether the sub-problem exists.

Other yes-no questions ask for the existence of certain entities:

"Were overhead budgets exceeded by more than 10% in any plant?"

These questions are often followed by "Which ones?" Thus, the list of identities that pass the test in the yes-no question should be kept by the system for at least the following question just in case it is an identity question based on the previous one.

The system must also be able to answer identity questions as they are often used to isolate the existence of a problem to a particular plant or product. For example:

"Which plants exceeded overhead budgets by more than 10%?"

"Which product had the largest increase in unit cost?"

8.9.4 Frame Building and Validation

Yes-no questions are also asked to validate frames. If answered in the affirmative they are followed by a question asking for data to eliminate a sub-problem or assign terminals in an analysis frame. Thus, whenever a yes-no question has to be answered in the affirmative, the support system should attempt to supply some of the information that may be relevant to the analysis frame that will be invoked. In some cases this can be done quite easily. For example, the question "Do you have profit margin by product?" should be answered by a listing of the margins for all products during the last year rather than by a "Yes, we do.".

Other yes-no questions are asked for system validation. The user wants to know what is available before he plans his problem-solving strategy. The system must, therefore, be able to answer questions about what it can do and about the data and knowledge it contains.

Frame building generally begins with yes-no questions about the problem environment. Answering these may be sufficient but in some cases the user may also wish to build models and the system must be able to allow this as well.

8.9.5 Scanning and Focussing Strategies

Questions that test if terminal sub-problems exist seem to be associated with what Bruner [9] called scanning strategies i.e., the testing of hypotheses that the subject has in mind. The converse, focussing strategies, would seem to correspond to the building up of frames from observed properties. Bruner recognizes that "the task of search imposed upon the user of .. . focussing may become rather severe", and this is even more true in situation analysis than in Bruner's concept attainment tasks. It is little wonder then that subjects preferred to solve the problem by frame analysis and were disconcerted when their frames failed and they were forced to modify or supplement them.

8.10 NEED FOR A GOOD MODEL OF THE WORLD

The prime determinant of efficiency and success in problem analysis seemed to be a good set of frames for corporate profitability. It seemed to provide the subjects with a basic structure within which to operate. The more successful subjects spent a few moments planning the investigation before they started asking questions. This plan was based on their frames and included the data required and, contingent on the values obtained, a general pattern for the problem-solving process.

Subjects who started by asking for data and attempting to make a plan as they went along, generally did poorly. They tended to get swamped by numbers that meant little to them since they did not have a framework to analyze them and had not thought through the implications of different data values.

The need for a good model-based plan of action is hardly a new or surprising result. It is good to see it confirmed, however.

CHAPTER 9

CONCLUSIONS

9.1 BASIC CONCLUSIONS

The basic conclusion that emerges from the analysis presented in the previous chapters is that an English language system, designed along the lines described in Chapter 2, would be useful to managers and can, indeed, be built. Such a system does justice to the complexity of the manager's problem-solving process and although the state of the art does not allow all the facilities required to be provided in an efficient and natural manner, a system can be implemented with sufficient power to provide meaningful assistance to managers. Such a system would be superior to any computer-based management aid available today.

The system will be deductive in nature in that it will support the manager's problem-solving process. It can be used in many ways depending on how the manager approaches the problem but it will do what it is told and will not make suggestions as to how the problem could be attacked.

We found that the vocabulary used by the subjects was limited and that most of their requests were phrased very simply. This result takes on even more significance if we realize that the sentences were obtained from subjects using the system for the first time and without any restrictions on the types of sentences permitted. In real life, new users would be provided with examples of successful sentences and these would guide their usage. As they used the system they would learn the set of sentence types accepted by it and their usage would gravitate preferentially towards this set. These factors would increase the percentage of their sentences accepted by the system. If the set of allowed syntactic types is "habitable", in the sense of Watt [68], in that it is powerful enough for the user to be able to express his requirements in a natural manner, then the percentage of sentences accepted by the system will increase towards a hundred percent with time.

A vocabulary of 1000 to 1500 words backed by a morphological analysis program such as the one described in Chapter 5, and a parser that recognizes, say, twenty sentence types and syntactic conventions, uses the meanings of words for case assignment and has mechanisms to handle conjunctions, relative clauses and other syntactic features described in Chapter 6 should provide sufficient power for a valuable system. It is well within the current state of the art to provide these facilities in an efficient manner.

The incorporation of knowledge into the system is, however, less well understood. Many different kinds of knowledge are needed to give the system power and flexibility. Each type of knowledge has to be encoded in different ways within the structure of the knowledge representation language. Once conventions for representing each category of knowledge are established, however, the process of adding and encoding the knowledge may be large, laborious and tedious but is not intrinsically difficult.

The frame paradigm of problem-solving described in Chapter 8 provides confidence that our analysis of requests and facilities for answering the different types of questions is, indeed, general. According to the paradigm, managers ask questions to:

a. Test whether a potential sub-problem exists.

b. Determine the location of sub-problems.

c. Build and validate frames.

d. Test decision frames.

Thus, if the system is capable of answering questions about its capabilities and contents in addition to providing general problem-solving facilities, by answering questions of the above four types, it will be able to support a wide range of managerial activities.

9.2 SYSTEM CHARACTERISTICS

It is time now to reconsider the system design described in Chapter 2 in light of managerial needs as demonstrated by the experiment and our assessment of the state of the art. Not surprisingly, the features included in the system, were all found to be valuable by the subjects. The values placed on different facilities by the users seemed to be different, in some cases, from those anticipated but no facility was neglected enough to be relegated from the system. A few minor facilities not provided by the system were also felt to be desirable.

The state of the art does, however, limit the quality of the facilities that can be provided in certain cases. On the balance, however, there is adequate capability to build a system that will be very useful to managers.

From among the general characteristics listed in Chapter 2, the English language interface and the ability to request information about the system as well as the corporation were found to be useful. The subjects were able to start working quickly and efficiently with the system and to continue their investigation in a natural manner.

Bomb-proofing, i.e protecting the user from system errors, could not, of course, be tested by the experiment. It is however, strongly recommended on general principles. For this to be possible the system must be written in a language which, in case of system error, generates interrupts that can be processed by user written routines.

A more serious problem may occur when the system, through erroneous understanding of a request or due to error, creates an incorrect answer. The user must believe in the system to use it successfully and although the experiment seems to indicate that people trust output from the system implicitly, there is reason to believe that if the user does detect an error he may become sceptical. This can seriously diminish the utility of the system. If further errors occur he may lose faith completely and abandon the system.

The understanding routine should, therefore, be extremely careful. No part of the sentence should be omitted from analysis and if the system is uncertain about the intent of a request or of a deduction it has made it should ask a clarifying question. If it is reasonably certain its understanding is correct it should generate the answer and indicate the question it has answered.

Data retrieval was the most popular of all the facilities requested by the subjects. It is also relatively simple to provide. The major difficulty in analyzing and complying with requests for data is in the matching of noun groups naming the information required with data names known to the system. This is discussed in Chapter 7 and in Appendix III.

The system limitation most frequently cited by the subjects was the primitive formatting facility of the prototype system. It seems important for effective problem solving to be able to print out data in tables with the figures lined up one above the other and with commas after every three positions to indicate significance. It also seems important to be able to express the answers in thousands or in millions and to change the number of significant digits in them. There is no serious technical difficulty in providing these facilities.

The only mode of data retrieval allowed in the system was to ask for one data item at a time. Some subjects, however, wanted to see sets of data such as the profit and loss statement and the balance sheet and might, as an extreme example, ask for the general ledger. Retrieving and presenting these named sets of data also does not present any significant technical problems and should be included among the facilities offered by the system.

The power of the English language brings with it the disadvantage of verboseness. A variety of conventions have been devised to minimize this, such as pronouns, anaphoric reference and ellipsis. These are integrated into the language but there are others that are used in problem-solving. The system should accept these since users consider them important. The most powerful such device is the defining of models, i.e. naming a frequently used function of data. Models are discussed later in this section. Another such device is to set a series of key specifications that are to be used in all the succeeding requests until they are reset. For example, "Provide the following data for plant 2." This can be implemented quite simply by setting the specifications into special registers that are checked in the process of creating key specifications for data retrieval. It seems desirable, however, to print out the specifications each time they are used since the user may forget he has set them and misinterpret the answers.

While the verbosness and redundancy of English can be tolerated by the manager who needs its power, it can become intolerable for users who require mainly data retrieval and for very experienced users. Someone who uses the system in only an elementary manner should perhaps not be using an English language system but if he ocassionally requires the more powerful facilities he should be provided with a command language that allows him to specify data retrieval requests in an efficient and compact manner. These commands could be preceded by a special character to distinguish them from sentences, and processed by a simple command processor that translates them into the equivalent English sentences. Such a facility can be provided easily and efficiently and could considerably reduce the burden of typing long, routine requests. (See also section 9.4.4.)

One of the more significant results of the experiment was the importance of models to the problem-solving process. Not only did subjects ask for models as naturally as they asked for data, but most of them wanted to define new models and ask what-if questions that require models to answer them. Thus, the ability to build and execute models seems to be an important part of the managerial problem-solving process. It is very difficult, however, to provide conversational model-building facilities. The ability to describe models in English sentences and have the system set up appropriate internal structures is related to the general problem of having computer systems learn from information presented to them. Besides, the knowledge required to build models is very complex and it is difficult to describe models in single sentences. Thus a conversational model building facility would require the ability to understand and process information provided in a number of connected sentences.

Learning from natural language dialog and paragraph comprehension is somewhat beyond the current state of the art.

Since model building facilities are so important to problem-solving and it does not seem feasible to be able to provide them in natural English, the system should attempt to provide them in some other manner. Whenever the user attempts to define a model the system should invoke a special modelling sub-system. This sub-system could initiate a structured interaction with the user during which it asks questions and the user supplies the information needed to build the model in his answers. The sub-system would, of course, make extensive use of system knowledge to frame the questions. The ability to use information gained through structured interaction to specialize information systems has been demonstrated in the IBM System/3 Applications Customizer [31] and other questionnaire-based systems. It may be feasible, therefore, to build such a sub-system with the ability to generate models of a few generic types. In this manner, the user would have access to a fairly powerful model-building facility rather than a rudimentary, conversational model-building system.

What-if questions ask for the value of a target variable given particular values for contributing parameters and states of nature. Such questions can only be answered if a model exists with the target variable as output and the specified parameters and states of nature as inputs. The generation of responses to what-if questions should start, therefore, by looking for an appropriate model. If such a model can be found, the inputs should be picked up from the sentence or supplied by defaults and the answer created. If, however, a model cannot be found, the user should be told so and, if he wishes, led into the model building sub-system.

The subjects seemed to find "percentage increase" the most useful function available to them. Typically, they asked for percentage increases of comparative data over a set of entities. In addition to "percentage" and "increase" the system should provide at least the following functions: "average", "maximum", "minimum", "sum", "difference", "change", "variance" (both accounting and statistical), and "distribution". Functional capabilities are fairly straightforward to provide and the system design should lean towards prolixity rather than parsimony.

The ability to answer yes-no questions and identity questions is extremely important to the success of a managerial question-answering system. Indeed, yes-no questions were the third most popular syntactic type in the sentences obtained from users. These questions are also difficult to answer because special pieces of knowledge are required to understand them. Consider:

"Who is our largest customer?"

"Is Sears our largest customer?"

In these sentences the word "largest" acquires a special meaning, namely "the one who bought the most from us". The utility of such a piece of knowledge is restricted to a narrow range of input requests and a number of such pieces are required. Nevertheless, it seems possible and necessary to provide adequate facilities in these areas.

9.3 PREFERRED ANSWERING STRATEGIES

At various points in this thesis we have mentioned preferred strategies for answering one or another type of question. Some of these improve system efficiency but most of them are designed to provide better information to the user and better support the problem-solving process. We shall describe these preferred strategies in this section and recommend that they be made an integral part of the design of the system.

Our basic assumption underlying answer generation is that people appreciate brevity and tire of repetition. If they have faith in the system and it analyzes their requests carefully and without guessing then there is little need to specify the question in the response. The answers should be, therefore, as brief as possible. If data is asked for, it should be presented without any explanation. If the question is "Who is our largest customer?" the answer should be "Sears", not "Our largest customer is Sears". Defaults and assumptions made by the system should, however, be stated along with the answer on the basis that the user should know all the information used in generating the answer.

This brief style of answering is quite different from the one used in the SOPHIE system [10]: "To minimize the consequences of misrepresentation, the system always responds with an answer which indicates what question it is answering, rather than just giving the numeric answer". This is because the SOPHIE parser can ignore certain words in the input string and may, therefore, understand and answer a question incorrectly. Our philosophy is to be extremely careful about understanding and answering each question. For example, no input word is ever ignored. There is, therefore, less need to indicate the question in the answer.

9.3.1 Yes-No Questions

Questions of the type:

"Do you have sales figures?"

"Can you show me overhead cost?"

should be treated as if the user had requested the data or model specified. These two questions should be answered by providing sales and overhead costs for the most recent year. If it is not possible to provide the information requested the system should respond with the definition of the data or model required.

Questions of the type:

"Is transportation cost included in overhead?"

should be replied to with either a "yes" or with information about where transportation cost is really included. In general, the system the system should try to indicate the correct state of affairs rather than respond to such questions with merely a "no".

In some cases, additional information should also be included with a "yes" answer. For example:

Was actual expense in plant 4 higher than budget?"

If it was, the system should anticipate the following "By how much?" and provide the variance. Similarly,

"Did any product cost exceed budget in 1973?"

should receive a response indicating the variances for the costs that were over budget.

9.3.2 Identity Questions After Yes-No Questions

Yes-no questions asking whether entities with given properties exist are often followed by questions asking for their identities.

"Did any plants exceed their production budget
in 1973?"

"Which ones?"

Since this is a common sequence, also reported by Woods [73], it seems desirable to check the properties of all the relevant entities in answering the yes-no questions, not stopping after the first positive instance, and to keep the list of positive instances in a special position to answer the identity question.

9.3.3 Fuzzy Discriminating Functions

Managers exploring the existence of a sub-problem may ask a yes-no question that requires the system to make a judgement on some data.

"Were profit margins maintained in 1973?"

"Did unit costs increase significantly last year?"

Such questions are identified by "fuzzy" words such as "maintained", "changed" and "same". The system cannot provide the judgement needed to answer these questions. It should, therefore, provide the data and print a message saying that the user should draw his own conclusions.

9.3.4 Free Standing Noun Groups

The SOPHIE system [10] has a default that if a user types in a noun group which is a "measurement" he is assumed to want its value. Some of the experimental subjects, mainly the poorer typists, tended to drop the "What is" before a request for a data item and type just the noun group, optionally followed by preposition groups. The default does, therefore, seem to be a good one and worth adopting.

9.3.5 Definitions

Every entity known to the system should have a prepared definition and description that should be printed out if it is directly asked for or if the user makes an incorrect request related to it. In fact, there probably should be a definition and special messages to respond to different ways in which a request regarding that entity can be erroneously specified. (See section 7.13.)

9.3.6 Questions That Cannot Be Answered

The best of systems will not be able to answer all questions put to them. In case a question cannot be answered the system should politely ask the user to rephrase it. It should also attempt to provide information that tells the user why the question could not be answered so that he may avoid similar problems in the future. For example, the system should indicate whether the sentence could not be parsed or a response created from the parse. If the parser failed because of an unusual construction, this should be pointed out. Similarly, if the request could not be responded to because the data was not available, or if the files were inappropriately structured, or if the entity did not have the required property this should be indicated to the user. Such information will also be useful for compiling the "wish list" of features to be incorporated into the system.

The system may not be able to answer a question because the request was ambiguous or did not provide enough information. In such cases the system should attempt to isolate the problem and tell the user exactly why it could not create the expected response.

9.4 IMPLEMENTATION ISSUES

Since the main result of this thesis is that an English language support system for managers is feasible and one of the obvious directions for future research is to implement such a system, we should touch briefly on implementation issues. First, since the amount of knowledge required, although tractable, is rather large, such a system should be built for specific, limited problem domains. There may, thus, be a support system for budgeting and another for controlling production costs and so on. Each system would be focussed on a particular area but could be used, if needed, by a user who had little knowledge of the area.

Second, such a system would resemble a service rather than a product. It would have to be brought up especially for each particular problem area and it would change and grow with the managers and their jobs and their understanding of the situation. It seems best, at this stage, to relegate the functions of adding knowledge to the system to an intermittent, background, system maintenance phase. Thus, addition of new words, new data or new information to the system will have to be done through a "wish list" which is continuously compiled by the user and ocassionally processed by system programmers. This process of adding to the system will be extremely important to its success and the user would probably pay much more for system updating and maintenance than he would pay as the initial price.

Third, the thinking presented in this thesis has assumed a simple data base structured in the form of arrays. Real world data bases are, however, very much more complex consisting of sequential, indexed sequential, random, inverted and chained files. The retrieval mechanisms from such files will need to be very sophisticated and use knowledge about the structure of the files. Furthermore, certain kinds of questions cannot be answered from an inappropriately structured data base without a record by record search that may cover the entire data base. These questions must be considered inappropriate for the data base and should recieve an "error" response. Such issues need to be resolved before implementation can be considered and would seem to be fruitful areas for further research.

9.5 DIRECTIONS FOR FUTURE RESEARCH

Other than direct implementation of an English language support system for a particular real-world problem area there seem to be two basic general areas for further research; to improve the technology on which the system is based and to improve the design of the system through better understanding of management and managerial needs.

9.5.1 The Parser

We discussed in section 6.1 how the processes of parsing and understanding a user request could be looked at as linking the input sentence into various levels of knowledge. Typically, the parser uses general, syntactic knowledge while the processor uses more specialized, semantic knowledge. Unless the objective is to create a general purpose parser that can be used for a number of English language systems, there is no basic reason for this sequence of knowledge use. As we discussed earlier, in the sub-section on data and model requests, there is duplication in such a sequence. Similar analysis is carried out first with general-purpose and subsequently with special-purpose knowledge. Brown and Burton [7] describe an interesting parser for the SOPHIE system that uses semantic knowledge very early in the analysis of the sentence. This leads to an extremely efficient and compact implementation. The system is still under development but has had some encouraging early tests.

Integration of the understanding process using semantic knowledge with the parsing process seems, therefore, to be a useful direction for further investigation.

9.5.2 Knowledge Representation

On a more basic level, a great deal of research is required in general methods for representing and processing knowledge. The Automatic Programming Group at Project MAC, M.I.T. has been doing some pioneering work in this direction. They plan to produce a very powerful general-purpose parser and interpreter based on their methodology.

9.5.3 Human Problem-Solving

Another basic area for further research is in the development of better models for problem-solving. The frame theory, described in Chapter 8, provides one beginning but a great deal of development and testing is necessary before we have a complex model of problem-solving. Other areas of interest are the stability of frame systems across people and of frame characteristics across problems.

A reactive system that does what the user tells it to is open to the argument that although it can assist the manager in his problem-solving process he is still confined to his conceptual model of the world and is limited by it if it is incorrect or incomplete. If the frame theory does turn out to be a good way to look at problem-solving then the possibility exists of being able to incorporate normative frame structures into the managerial support system. This would make the system active rather than merely reactive in that it would be able to suggest detailed causes and solutions for certain problems. Further, a system that operates along these lines could be written as a consultant. The manager would describe the problem to it and it would use its frames to suggest causes and correlations. Each of these could be explored in collaboration with the manager until he reaches an understanding of the problem. The danger with an active system is that the manager may not understand it, may be threatened by it and, therefore, not use it. Thus, the level of intelligence of the system must be chosen with great care.

9.5.4 Detailed Analysis of User Requirements

The designer of a management support system needs to know the exact functions and facilities that a manager would use as part of the problem-solving process. This kind of research could be continued by conducting experiments like the one described in this thesis on different kinds of data bases and with different management problems.

A variation may be to impose a filter between the user and the experimenter that screens out requests that do not belong to certain sentence types or functional categories. By varying the bandwidth of this filter we could determine how problem-solving performance is affected by restrictions in input language and the available facilities.

Richard Burton [11] found that users would employ anaphoric reference if the system responded to their request within a few seconds but would type a complete sentence if the response time was slower. This suggests that the user loses his trend of thought if the response time exceeds a few seconds. Fast response time is, therefore, important to problem-solving. We need many results of this kind before we can design a support system that understands human beings well enough to be truly responsive.

Nonetheless, the analysis presented in this thesis shows that it is now possible to implement a system that mirrors the complexity of the managerial problem-solving process and allows both new and experienced users to work easily and naturally with it. Powerful technology in natural language processing and knowledge representation and processing now exists and is being strengthened further. The next logical step seems to be to implement such a system for a real situation and learn from an analysis of the actual use that managers make of it. This is probably the most effective way to make progress in responsive support systems for managers.

NOTES

1. The simulation programs for the perfect English language system were written by R. V. Baron.
2. The morphological analysis program used by the prototype system was written by E. R. Banks.
3. G. A. Gorry, in a personal communication, has suggested that frames correspond to the problem-spaces of Newell and Simon [50]. They form an internal representation of the problem and dictate the model and process that will be used to solve it.
4. The idea of frames as conceptual hueristics is due to P.G.W. Keen. Some of his other ideas are also interwoven into Chapter 8.
5. Further investigation of frame structures, their stability and their correlation with attitudes, personality variables and experience will be reported in a forthcoming publication. We will explore the stability of frame structures across managers and whether the same set of terminals appears arranged in different configurations. Experienced managers may tend to have narrower and deeper frames while neophytes may have wider, shallower frames. The pattern and sequence

of terminal filling may also vary with background and experience.

BIBLIOGRAPHY

(1) Ableson, R.P., "The Structure of Belief Systems." in Reference 58 , pp 287-341

(2) Allen, J., "The Talking Computer.", Reports on Research Vol.I No. 2, M.I.T., Cambridge, Mass., April 1974

(3) Allen, J., personal communication.

(4) Black, J. L., "Event Oriented Language User's Guide.", Honeywell Information Systems, Advanced Systems and Technology, April 1971 (Revised August 27 1971)

(5) Bleier, R. E., "Treating Hierarchical Data Structures in the SDC Time-Shared Data Management System (TDMS).", Proc. ACM 22nd Nat. Conf., 1967, MDI Publications Wayne, Pa.

(6) Bobrow, D., and Fraser, J.B., "An Augmented State Transition Network Analysis Procedure.", Proc. IJCAI 1969

(7) Brown, J.S., and Burton R.R., "Multiple Representation of Knowledge for Tutorial Reasoning.", to appear in Bobrow, D., and Collins, A.M., (eds.) "Representation and Understanding: Studies in Cognitive Science." Academic Press, New York., N.Y., to appear Spring 1975

(8) Brown, J.S., Burton R.R., and Bell, A.G., "SOPHIE: A Step Toward a Reactive Learning Environment.", Proc. Nat. ACM Conf., November 1974

(9) Bruner, J. S., Goodnow, J. J. and Austin G. A., "A Study of Thinking", John Wiley, New York, N.Y. 1957

(10) Burton, R.R., "Semantically Centered Parsing System for Mixed Initiative CAI Systems.", presented at the Twelfth Annual Meeting of the Association for Computational Linguistics, Amherst, Mass., July 26-27 1974

(11) Burton, R.R., personal communication.

(12) Carbonell J. R., and Collins A. M., "Natural Semantics in Artificial Intelligence," Proc. IJCAI., 1973

(13) Celce-Murcia, M., "Paradigms for Sentence Recognition.", in System Development Corporation final report No. HRT-15092/7907

(14) Chafe W., "The Meaning and Structure of Language.", Chicago, 1970

(15) Charniak, E., "Towards a Model of Children's Story Comprehension.", AI TR-266, Artificial Intelligence Laboratory, M.I.T. Cambridge, Mass., 1972

(16) Chomsky, N., "Syntactic Structures.", Mouton, The Hague, 1957

(17) Dearborn, D.C., and Simon, H.A., "Selective Perception: a Note on the Departmental Identification of Executives.", in Alexis, M., and Wilson, C.Z., (eds.), "Organizational Decision Making.", Prentice Hall, Englewood Cliffs, N.J., 1967

(18) Feather, H. E., "Charlie II User's Guide.", Honeywell Information Systems, Advanced Systems and Technology, December 15 1971

(19) Fillmore, C. T., "The Case for Case.", in "Universals in Linguistic Theory.", Bach, E., and Harms R., (ed.), Holt, Rinehart and Winston, 1968

(20) Forrester, J.W., "Principles of Systems.", Wright-Allen Press, Cambridge, Mass., 1968

(21) Gorry, G.A., "The Development of Managerial Models.", Industrial Management Review, Vol. 12, No. 2, Winter 1971

(22) Gore, W.J., "Administrative Decision-Making: A Hueristic Model.", John Wiley, New York, N.Y., 1955

(23) Green, B.,F. Jr., "Baseball: An Automatic Question Answerer.", in "Computers and Thought.", Feigenbaum, E. A., and Feldman, J., (Eds.), McGraw Hill, New York, 1963

(24) Hall, W.K., "Strategic Planning Models: Are Top Managers Really Finding Them Useful?", Journal of Business Policy, Vol 3, No. 2, 1972

(25) Hayes, R.H., and Nolan, R.L., "What Kind of Corporate Modelling Functions Best?", Harvard Business Review, May/June 1974

(26) Honeywell Information Systems, "Data Management System User's Guide", Series 6000 WWMCCs Software Order Number DB97, April 1974

(27) Honeywell Information Systems, "DataBASIC System Language Manual.", Honeywell Series 6000/600, Software Order Number DA08, May 1971

(28) Honeywell Information Systems, "Inquiry Command Language User Manual", Database Technology Center, Schenectady, New York, January 1971

(29) Honeywell Information Systems, "I-D-S DATA QUERY System User's Guide." Honeywell Series 6000/600, Software Order Number DB56, February 1973

(30) Honeywell Information Systems, 'Management Data Query User's Guide", Series 6000 WWMCCs Software Order Number DC80, August 1974

(31) International Business Machines, "Application Customizer Service System/3 (Card, Disk, and Model 6) Application Description.", Document GH20-0628-5

(32) International Business Machines, "Information Management System/360 Application Description Manual.", Document H20-0524-1 July 1968

(33) Katz, J.J., "Semantic Theory." in Steinberg, D.D., and Jakobovits, L.A., (eds.) "Semantics.", Cambridge University Press, London, 1971

(34) Kay, M., "The Mind System.", in "Natural Language Processing", Randall Rustin (ed.), Algorithmics Press New York, N.Y. 1973

(35) Krumland R. B., "Concepts and Structures for a Manager's Modelling System," Thesis Proposal, Sloan School of Management, M.I.T., Oct 15 1973

(36) Kucera, H., and Francis, N., "Computational Analysis of Present-Day American English.", Brown University Press, Providence, R.I., 1967

(37) Lindsay, R. K., "Inferential Memory as the Basis of Machines That Understand Natural Language," in "Computers and Thought," Feigenbaum, E. A., and Feldman, J. (eds.), McGraw Hill, New York, 1963

(38) Little, J.D.C., "BRANDAID: An On-Line Marketing-Mix Model.", Working Paper 586-72, Sloan School of Management, M.I.T., Cambridge Mass, February 1972

(39) Little, J.D.C., "Models and Managers: The Concept of a Decision Calculus." Management Science, April 1970, B466-485

(40) Malhotra, A., "Control Structure for a Question-Answering System." Working Paper 693-74, Sloan School of Management, M.I.T., Cambridge, Mass, January 1974

(41) Malhotra, A., " A Descriptive Model of Detailed Problem-Finding Behavior.", Working Paper 695-74, Sloan School of Management, M.I.T., Cambridge, Mass, January 1974

(42) Martin, W.A., "Translation of English into MAPL using Winograd's Syntax and a Semantic Case Grammer", Automatic Programming Memo 11 Project MAC, M.I.T., April 17 1973

(43) Martin, W.A., "Things That Really Matter -- a Theory of Prepositions, Semantic Cases and Semantic Type Checking," Automatic Programming Memo 13 Project MAC, M.I.T, August 15 1973

(44) Martin, W.A., OWL: A System for Building Expert Problem Solving Systems Involving Verbal Reasoning.", Notes for Course 6.871, M.I.T., Cambridge, Mass., 1974

(45) Miller G.A., "The Magic Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information.", in Alexis, M., and Wilson, C.Z., (eds.), "Organizational Decision Making.", Prentice Hall, Englewood Cliffs, N.J., 1967

(46) Minsky M., (ed.), "Semantic Information Processing.", M.I.T. Press, Cambridge Mass., 1968

(47) Minsky, M., "A Framework for Representing Knowledge", Memo No. 306, Artificial Intelligence Laboratory, M.I.T., Cambridge, Mass., December 1973

(48) Mintzberg, H., "The Nature of Managerial Work.", Harper and Row, New York, N.Y., 1973

(49) Montgomery, D.B., "The Outlook for MIS.", Journal of Advertising Research, Vol. 13, Number 3, June 1973

(50) Mosher, F.A., and Hornsby, J.R., "On Asking Questions.", in Bruner, J.S., et al., "Studies in Cognitive Growth.", John Wiley, New York., N.Y., 1966

(51) Newell, A., and Simon, H.A., "Human Problem Solving.", Prentice Hall, Englewood Cliffs, N.J., 1972

(52) Ogden, C. K., "Basic English: International Second Language," Harcourt, Brace and World, New York, 1968

(53) Parzen, E., "Modern Probability Theory and its Applications.", John Wiley, New York, N.Y. 1960

(54) Petrick, S., "A Recognition Procedure for Transformational Grammars.", Unpublished Doctoral Dissertation, Department of Electrical Engineering, M.I.T., Cambridge, Mass., 1965

(55) Pounds W.F., "The Process of Problem Finding.", Industrial Management Review, Vol 11, No. 1, 1969

(56) Raphael, B., "A Computer Program for Semantic Information Retrieval," in Reference 46, pp 33-145

(57) Rubin, A. D., "Hypothesis Formation and Evaluation in Medical Diagnosis.", E.E. Thesis. Department of Electrical Engineering, M.I.T., Cambridge, Mass., May 1974

(58) Schank, R.C., and Colby, K.M., (eds.) "Computer Models of Thought and Language.", W.H. Freeman and Co., San Fransisco, Cal., 1973

(59) Schank, R.,C., "Identification of Conceptualizations Underlying Natural Language.", in Reference 58

(60) Schroder, H.M., Driver, M.J., and Streufert, S., "Human Information Processing.", Holt, Rinehart and Winston, New York, N.Y., 1967

(61) Simmons R.F., "Semantic Networks: Their Computation and Use for Understanding English.", in Reference 58

(62) Simon, H.A., "The New Science of Management Decision.", Harper and Brothers, New York, N.Y., 1960

(63) Teitelman, W., " Automatic Programmering: The Programmer's Assistant.", FJCC, December 1972

(64) Thorne, J., Bradley, P., and Dewar, H., "The Syntactic Analysis of English by Machine.", in Michie, D., (ed.) "Machine Intelligence 3." Elsevier

(65) Sussman, G. J., "Some aspects of Medical Diagnosis.", Working Paper 56, Artificial Intelligence Laboratory, M.I.T., Cambridge, Mass., December 1973

(66) Urban G.L. and Karash, R., "Evolutionary Model Building.", Journal of Marketing Research, Vol. VIII, February, 1971

(67) Urban, G.L., "PERCEPTOR: A Model for Product Design.", Working Paper 689-73, Sloan School of Management, M.I.T., Cambridge, Mass, December 1973

(68) Watt W. C., "Habitability.", American Documentation Volume 19, No.3, July 1968

(69) Wertheimer Max, "Productive Thinking." Harper & Brothers Publishers, New York, N.Y., 1945

(70) Woods, W.A., "Augmented Transition Networks for Natural Language Analysis.", Report No. CS-1, Aiken Computation Laboratory, Harvard Univ., Dec., 1969

(71) Woods, W. A., "Procedural Semantics for a Question-Answering Machine.", Proceedings of the FJCC 1968

(72) Woods, W. A., "An Experimental Parsing System For Transition Network Grammars.", in "Natural Language Processing", Randall Rustin (ed.), Algorithmics Press New York, N.Y. 1973

(73) Woods, W.A,, Kaplan, R.M., and Nash-Webber "The Lunar Sciences Natural Language System:

Final Report.", Bolt, Beranek and Newman, Cambridge, Mass., June 15 1972

(74) Wilks, Y., "An Artificial Intelligence Approach to Machine Translation.", in Reference 58

(75) Winograd, T., "An A.I. Approach to English Morphemic Analysis.", Memo No. 241, Artificial Intelligence Laboratory, M.I.T., Cambridge, Mass., February 1971

(76) Winograd, T., "Understanding Natural Language.", Academic Press, New York, 1972

(77) Zwicky, A.M., et al, "The MITRE Syntactic Analysis Procedure for Transformational Grammars.", Proc FJCC, 1965

APPENDIX 1

DETAILS OF THE EXPERIMENT

The experiment consisted of three parts. The subject began by filling out a biographical questionnaire and completing a Semantic Differential Test that probed his attitudes towards computers and computer systems. Next, he submitted to a concept attainment test. This duplicated, as far as possible, the test described by Bruner [9]. Three trials were conducted and during each of them the subject was allowed to work from an illustration showing the cards arranged in an orderly manner. This was reproduced from [9].

Finally, the subject read the problem scenario and the instructions using the system and proceeded to solve the problem using the simulated perfect English language question-answering system.

Completing the questionnaire and the Semantic Differential Test took between ten and fifteen minutes. The concept attainment test took between ten and thirty minutes and the problem solving between forty five and ninety minutes.

Typical times for the entire experiment ran about ninety minutes.

The following pages reproduce the materials used in the experiment.

BIOGRAPHICAL DATA

1. Name:

2. Age:

3. Sex:

4. Years you have used English:

5. Nationality:

6. How many years have you used computers:

7. Education:

   a. When did you get your degree: B.S. M.S. Ph.D.
   b. Major field: Undergraduate:
      Graduate:
   c. Semesters of: Management education:
      Engineering education:
   d. Semester hours of: Accounting:
      Production:
      (A semester hour is one hour of class a week for one semester)

8. Work Experience:
   a. Total number of years:
   b. Years of line experience*:
   c. Years of staff experience*:
   d. Years of experience in: Management:
      Production:
      Engineering:

9. Your area of expertise:

* A line job is characterized by the need to make decisions under time pressure with direct responsibility for the execution of decisions.
A staff job is advisory in nature and normally has no direct operating responsibility.

SEMANTIC DIFFERENTIAL TEST

A problem refers to the difference between some existing situation and some desired situation.

Problem Finding refers to the process of finding these differences.

Problem Solving refers to the process of choosing alternatives to reduce these differences.

INSTRUCTIONS

On top of each of the pages that follow is named an "instrument" that has use in Problem Solving. Below each "instrument" is a list of 10 dimensions. For each dimension, there are 7 possible values. Rate the "instrument" along each of the dimensions by putting a check mark in one of the 7 spaces provided for each dimension.

COMPUTER SYSTEM

COMPLETE	___	___	___	___	___	___	___	INCOMPLETE
TIMELY	___	___	___	___	___	___	___	UNTIMELY
MEANINGLESS	___	___	___	___	___	___	___	MEANINGFUL
UNIMPORTANT	___	___	___	___	___	___	___	IMPORTANT
UNCONSTRAINED	___	___	___	___	___	___	___	CONSTRAINED
WEAK	___	___	___	___	___	___	___	POWERFUL
OPAQUE	___	___	___	___	___	___	___	TRANSPARENT
PASSIVE	___	___	___	___	___	___	___	ACTIVE
FAST	___	___	___	___	___	___	___	SLOW
COMPLEX	___	___	___	___	___	___	___	SIMPLE

This page was reproduced five times with the headings: COMPUTER SYSTEM, STAFF ASSISTANT, REPORTS, MANAGEMENT INFORMATION SYSTEM, and DATA.

INSTRUCTIONS

On top of each of the pages that follow is named an "instrument" that has use in <u>Problem Finding</u>. Below each "instrument is a list of 10 dimensions. For each dimension, there are 7 possible values. Rate the "instrument" along each of the dimensions by putting a check mark in one of the 7 spaces provided for each dimension.

COMPUTER SYSTEM

COMPLETE ___ ___ ___ ___ ___ ___ ___ INCOMPLETE

TIMELY ___ ___ ___ ___ ___ ___ ___ UNTIMELY

MEANINGLESS ___ ___ ___ ___ ___ ___ ___ MEANINGFUL

UNIMPORTANT ___ ___ ___ ___ ___ ___ ___ IMPORTANT

UNCONSTRAINED ___ ___ ___ ___ ___ ___ ___ CONSTRAINED

WEAK ___ ___ ___ ___ ___ ___ ___ POWERFUL

OPAQUE ___ ___ ___ ___ ___ ___ ___ TRANSPARENT

PASSIVE ___ ___ ___ ___ ___ ___ ___ ACTIVE

FAST ___ ___ ___ ___ ___ ___ ___ SLOW

COMPLEX ___ ___ ___ ___ ___ ___ ___ SIMPLE

This page was reproduced five times with the headings: COMPUTER SYSTEM, STAFF ASSISTANT, REPORTS, MANAGEMENT INFORMATION SYSTEM, and DATA.

INSTRUCTIONS FOR THE CONCEPT ATTAINMENT TEST

1. THERE ARE 81 CARDS DISPLAYED ON THE CHART PROVIDED TO YOU.
2. EACH OF THESE CARDS CONTAINS FOUR ATTRIBUTES.
3. EACH ATTRIBUTE ON EACH CARD HAS ONE OF THREE POSSIBLE VALUES.

   NUMBER OF FIGURES: 1,2 OR 3

   COLOR OF FIGURES: RED, BLACK OR GREEN

   TYPE OF FIGURES: SQUARE, CIRCLE OR CROSS

   NUMBER OF BORDERS: 1,2 OR 3
4. YOU WILL BE GIVEN A CARD AS AN EXAMPLE OF THE CONCEPT THAT THE EXPERIMENTER HAS IN MIND.
5. THE CONCEPT WILL CONSIST OF SOME OF THE ATTRIBUTES OF THE EXAMPLE. FOR INSTANCE, IF THE EXAMPLE IS

   "TWO GREEN SQUARES WITH THREE BORDERS",

   THEN "GREEN SQUARES" IS A VALID CONCEPT

   AND "TWO SQUARES WITH THREE BORDERS" IS A VALID CONCEPT

   BUT "GREEN SQUARES OR RED CROSSES" IS NOT A VALID CONCEPT

   BECAUSE IT CONTAINS AN ATTRIBUTE NOT IN THE EXAMPLE.
6. THE EXPERIMENTER WILL TELL YOU WHETHER A SELECTED CARD IS AN EXAMPLE OF THE CONCEPT.
7. AFTER EACH SELECTION YOU MAY TEST WHETHER YOU HAVE ARRIVED AT THE CONCEPT.
8. ONLY ONE TEST IS PERMITTED AFTER EACH TEST HOWEVER.
9. ATTEMPT TO DISCOVER THE CONCEPT AS EFFICIENTLY AS POSSIBLE.

THE PROBLEM SCENARIO

The Battery Company is an established manufacturer of lead batteries with head offices located in the mid-west. It has four plants where the actual manufacturing is carried out. These are spread out over the continental United States.

The Battery Company manufactures five basic battery types for various purposes. Each battery type is identified by a product number.

The Battery Company sells mainly in bulk to five major customers located all over the United States. Customers place long range contracts with The Battery Company for specified quantities of a certain product. The Battery Company supplies against these contracts on the receipt of orders from customer branches. Each branch is expected to order from the plant closest to it. In general, a given plant supplies customer branches in a set of states surrounding it.

Each plant manufactures the products it supplies. Only in rare cases of shortages and lack of facilities to manufacture a specialized unit will batteries be supplied from other than the closest plant.

Plants are expected to meet budgets on direct costs and overheads. Performance against budget as well as customer service are the main criteria for plant manager evaluation. Plants are not run as profit centers because prices on contracts are negotiated by the head office. A list of standard prices exists as a guideline for negotiating contracts.

It is February 1974 and as President of The Battery Company you are a little concerned at the results for 1973 that you have just received. Despite a 20% increase in sales over 1972, profits decreased by 1%.

You feel that the decrease in profit could be due to a combination of three causes: increase in overhead expenses, decrease in contribution (or profit) margins (difference between selling price and direct manufacturing cost) or a change in product mix toward less profitable units. Alternatively, you would like to know how the additional revenues from increased sales were spent. You would like to investigate the cause of the decreased profit using the The Information System. Depending on what you find, you will take a decision to enforce strict control on the pricing of contracts, review and reset list prices which are supposed to serve as guidelines for contract pricing, or introduce a program of cost control. The purpose of this exercise is to determine which decisions are appropriate under the circumstances.

As sales growth has been very healthy, you are inclined to disregard competitive actions in your analysis. You also assume that the cost and other data contained in the system are accurate.

The following sections contain a description of The Information System ; what it can do and how to use it.

THE INFORMATION SYSTEM

The Information System contains data on sales, costs, prices and other indicators of The Battery Company's operations. It is capable of answering questions posed to it in simple English about the contents of the database and functions of these contents such as "profit" or "average price for product 3". In addition, the system is capable of answering questions about itself, i.e. it can enumerate the data items it contains, explain the procedures embedded in the functions, etc.

The system can be queried much as you would use an assistant to answer questions, prepare reports, etc. It will provide appropriate responses to requests it does not understand or cannot reply to. A typical dialog with the system may be:

(User input is in lower case, system response in upper case.)

Q: What data do you have regarding product costs?

A: I HAVE ACTUAL AND BUDGETED COSTS FOR EACH
PRODUCT AT EACH PLANT.

Q: What was the unit manufacturing cost of product 3
in plant 4 ?

A: $18.23
DEFAULT YEAR 1973
(Since the time period was unspecified the system assumes
a reasonable value)

Q: What was the list price for product 3 in 1973?

A: $21.00

Q: Do you have a model for contribution margin ?

A: YES

Q: How does it work ?

A: IT COMPUTES THE LIST (STANDARD) PRICE MINUS ACTUAL
MANUFACTURING COST FOR THE GIVEN PRODUCT.

Q: What was the contribution margin for product 3 at plant 2?

A: $1.20
DEFAULT YEAR 1973

Q: What was the avreage cost of product 1?

A: SORRY, I DON'T KNOW THE WORD "AVREAGE".

Q: What was the average cost for product 1?

A: $18.67
DEFAULT YEAR 1973

Q: What was the average budgeted cost for product 1?

A: $16.00
DEFAULT YEAR 1973

HOW TO USE THE SYSTEM

After the system is loaded by the experimenter you can type into it in English and expect intelligible English responses. Questions about the system, its contents and its use can also be asked in the same way.

The only special conventions you need to remember are that each request must be followed by a special character called "Control S" which prints as ↑S and is struck by keeping the "CTL" key pressed while striking "S". This sends your request to the system. The system types out "REQUEST SENT" followed after a few moments by a response to the request.

Striking the delete (DEL) key at the extreme right of the keyboard erases the last letter. Typing ↑K (in a manner analogous to ↑S) displays the current state of your message. After it is displayed, you may send it as is by a ↑S, or you may amend it before sending it. The display of the current message will contain some "-" signs. These are used to pad out the message buffer and are not part of your message. Please ignore them. Similarly, the output of the system will contain ocassional "↑L" signs. These should also by ignored.

An example may be useful here. "(DEL)" represents the delete character.

>

How much dia(DEL)d wee(DEL) ↑K

HOW MUCH DID WE ---sell to Sears in 1972 ? ↑K

HOW MUCH DID WE SELL TO SEARS IN 1972 ? ↑S

REQUEST SENT

$ 2453478.72 ↑L

>

The system will be ready and waiting for a question when you start your problem solving session. After responding to your request, the system will print a ">" to indicate it is waiting for another question.

APPENDIX II

THE REQUESTS MADE BY THE SUBJECTS

The requests made by the subjects are listed below. They are presented as they were typed in except for the correction of spelling errors which would have been caught by the morphological analysis routine. Many subjects used the ↑S character that transmitted their request to the system as terminal punctuation mark. This has been replaced by the appropriate terminal punctuation.

The numbering scheme reflects the units in which information was sent to the system by the subject. Usually this was a sentence at a time but occasionally it consisted of more than one sentence strung together.

SUBJECT 1

The subject had ten years of management experience but no formal management training.

1. WHAT WAS THE PERCENTAGE OF OVERHEAD COSTS TO TOTAL SALES

FOR THE LAST FIVE YEARS?

2. WHAT WERE THE PROFIT MARGINS FOR EACH PRODUCT FOR THE LAST 5 YEARS?

3. WHAT ARE THE OVERALL PROFITS ON OPERATIONS FOR THE PAST 5 YEARS?

4. WHAT WERE THE GROSS SALES FIGURES FOR THE PAST FIVE YEARS?

5. WHAT ARE PROFIT MARGINS AS A PERCENTAGE OF SALES FOR EACH MANUFACTURING INSTALLATION?

6. WHAT IS THE PROFIT CONTRIBUTION OF EACH MANUFACTURING INSTALLATION?

7. OK.

8. WHAT DATA DO YOU HAVE ON OPERATIONS AS A PERCENTAGE OF GROSS SALES?

9. YES AND FOR EACH PLANT.

10. CAN YOU CALCULATE PERCENTAGES?

11. WHAT ARE THE PRODUCTION COSTS AS A PERCENTAGE OF SALES?

12. WHAT ARE THE DEVIATIONS OF PRODUCTION COST FROM ACTUAL?

13. CANCEL THIS QUESTION.

14. WHAT IS THE RATIO OF ACTUAL COST TO BUDGETED COST FOR EACH PRODUCT?

15. WHAT IS THE PERCENTAGE CHANGE IN SALES FOR EACH PRODUCT FOR THE PAST YEAR?

16. WHAT ARE VARIABLE COSTS FOR MANUFACTURING OPERATIONS?

17. HAVE TRANSPORTATION COSTS INCREASED DURING THE PAST YEARS?

18. WHAT ARE THE RATIOS OF PRODUCTION COSTS TO SALES?

19. WHAT HAS THE AVERAGE SELLING PRICE FOR EACH PRODUCT BEEN FOR THE PAST TWO YEARS?

20. WHAT QUANTITIES WERE PRODUCED FOR EACH PRODUCT FOR THE PAST TWO YEARS?

SUBJECT 2

The subject was a Ph.D. student at the Sloan School of Management, M.I.T. He had some four years experience as a staff manager.

1. PLEASE DISPLAY OVERHEAD COSTS FOR ALL PLANTS FOR 1972 AND 1973.

2. WHAT WAS BUDGETED OVERHEAD FOR ALL PLANTS FOR 1972?

3. WHAT WERE SALES AND PROFITS FOR 1973?

4. DO YOU HAVE VARIABLE BUDGETS?

5. DO OVERHEAD COSTS VARY WITH VOLUME?

6. I BELIEVE YOUR OVERHEAD VARIANCE ACCOUNTS FOR YOUR LOWER THAN EXPECTED PROFITS.

7. I SUPPOSE I SHOULD.

8. WHAT WERE CONTRIBUTION MARGINS BY PRODUCT FOR 1972 AND 1973?

9. HOW ARE CONTRIBUTION MARGINS CALCULATED?

10. WHAT IS THE DIFFERENCE BETWEEN LIST PRICE AND AVERAGE SELLING PRICE?

11. FOR 1972?

12. WHAT IS THE DIFFERENCE BETWEEN PLANT 1 AND PLANT 3 AND PLANT 2 AND PLANT 4?

13. WHAT ARE THE COMPONENTS OF OVERHEADS FOR THE PLANTS?

14. IS TRANSPORTATION COST PART OF OPERATING EXPENSES?

15. WHERE DOES TRANSPORTATION COST GET INCLUDED?

16. GIVE ME THE CONSTITUENTS OF OVERHEADS FOR EACH PLANT?

17. I WOULD LIKE TO END THE INTERVIEW.

SUBJECT 3

The subject was a school teacher with no management experience. At the time of the experiment he was a masters student at the Sloan School of Management, M.I.T.

1. LIST SALES FOR PRODUCT 1 THROUGH PRODUCT 5 FOR THE LAST TWO YEARS?

2. LIST PRICES OF SINGLE UNIT PRICES FOR BOTH 72 AND 73.

3. WHAT WAS COST OF PRODUCING EACH PRODUCT FOR BOTH 1972 AND 1973?

4. PRODUCTION COST FIRST FOR ONE UNIT.

5. DID ONE PLANT ASSUME MORE PRODUCTION OF BATTERIES FROM THE OTHER PLANTS IN 1973?

6. IN 1972?

7. WHAT WAS THE RATE OF INCREASE OF SHIPPING COST BETWEEN 1972 AND 1973?

8. ARE SHIPPING COSTS REFLECTED IN PRODUCTION COSTS?

9. DO YOU HAVE ANY INFORMATION ON CUSTOMER SATISFACTION?

10. WHAT IS THE PERCENTAGE OF REPEAT CUSTOMERS IN 1973 AND

1972 AND IN 1971 AND 1972?

11. WHAT WAS THE UNIT PRICE IN 1973?

12. WHAT'S RATE OF UNIT COST FOR EACH YEAR AND THE RATIO OF THIS PRODUCTION INCREASE TO PRODUCT PRICE?

13. WHAT IS THE PERCENT OF INCREASE OF EACH PRODUCT FOR EACH YEAR STUDIED?

14. WHAT WAS PERCENT DECREASE IN SALES FOR LAST 5 YEARS?

15. CANCEL THIS QUESTION.

16. WHAT WAS % OF PROFIT FOR EACH OF LAST 5 YEARS?

17. INCREASE OVER LAST YEAR.

18. WHAT WERE OVERHEAD COSTS FOR LAST 5 YEARS?

19. WHAT IS LIST PRICE VS SELLING PRICE FOR LAST 2 YEARS?

20. WHAT'S DIFFERENCE BETWEEN LIST PRICE AND AVERAGE QUOTATION PRICE?

21. PRODUCT 4 AND 5 SHOW GREATEST DIFFERENCE BETWEEN LIST AND QUOTATION PRICES, WHY?

22. DO YOU HAVE LIST OF CHANGES IN SALES FORCE FOR EACH BRANCH?

23. IT SHOULD BE INCLUDED.

24. GIVE ME A BREAKDOWN OF ITEMS IN YOUR OVERHEAD.

25. INCLUDE EACH OF THESE BY PLANTS.

26. COMPARE OVERHEAD COSTS FOR LAST 5 YEARS.

27. DO YOU HAVE FURTHER BREAKDOWNS OF OVERHEAD ATTRIBUTABLE TO EACH PRODUCT WITHIN PLANTS?

28. WHAT % OF EACH PRODUCT IS SOLD FROM EACH PLANT FOR EACH OF THE LAST 5 YEARS?

29. WHAT IS TOTAL VOLUME FOR EACH PLANT (SALES) FOR EACH OF THE 5 YEARS?

30. COMPARE PLANT OVERHEAD COSTS WITH TOTAL OVERHEAD COSTS FOR LAST 2 YEARS, 1973 AND 1972.

31. LIST INCREASES IN OVERHEAD FOR EACH PLANT FOR LAST FIVE YEARS.

32. COMPARE OVERHEAD COSTS FOR PLANTS FOR LAST 5 YEARS.

33. WHAT ARE SALARY INCREASES FOR EACH PLANT FOR LAST TWO YEARS?

34. LIST INCREASE IN INTEREST COSTS FOR LAST TWO YEARS.

35. LIST INVENTORY OF PRODUCT AT END OF 1971 AND 1972.

SUBJECT 4

The subject was a Ph.D. student at the Sloan School of Management, M.I.T. He had three years experience as a high school teacher.

1. LIST ALL DATA ITEMS YOU KNOW ABOUT.

2. HOW MANY PLANTS ARE THERE?

3. FOR 1973 LIST THE SALES OF PRODUCTS PRODUCED BY PLANT ONE, BROKEN DOWN BY PRODUCT.

4. FOR 1973 AND FOR PRODUCTS 1 THROUGH 5 LIST PRICES AND PERCENTAGE PROFIT MARGIN.

5. FOR 1973 AND PLANT 1 LIST DIRECT MANUFACTURING EXPENSES BY PRODUCT AND ALSO TOTAL OVERHEAD.

6. ARE THE PRICES THE SAME FROM PLANT TO PLANT?

7. BY PLANT, LIST OVERHEAD FIGURES FOR 1972 AND 1973.

8. FOR EACH PLANT, LIST THE RATIO OF OVERHEAD TO SALES IN 1972 AND 1973.

9. IN THE FUTURE, PLEASE EXPRESS NUMBERS OF OVER 100000 IN TERMS OF UNITS OF MILLIONS, AND NUMBERS OVER 100 BUT LESS THAN 100000 IN UNITS OF THOUSANDS.

10. LIST PRODUCTION COSTS BY PLANT FOR 1972 AND 1973.

11. FOR EACH PRODUCT, LIST THE PROFIT PERCENTAGE FOR 1972 AND 1973?

12. FOR EACH PRODUCT LIST THE RATIO OF TOTAL SALES TO TOTAL COST IN 1972 AND IN 1973.

13. WHAT IS THE DEFINITION OF THE PROFIT FOR A PRODUCT?

14. WERE THE PRICES THE SAME IN 1972 AND 1973?

15. WHY DID YOU GIVE ME PRICES OF $ 17, 18, 19.25, 20.25, AND 18.0 EARLIER?

16. FOR THE LAST TWO YEARS, WHAT IS THE ACTUAL PROFIT MARGIN PERCENTAGE FOR EACH PRODUCT?

17. DO YOU HAVE INFORMATION ABOUT TRANSPORTATION COST?

18. WHAT WAS TRANSPORTATION COST BY PLANT FOR THE LAST TWO YEARS?

19. IS TRANSPORTATION COST INCLUDED IN OVERHEADS?

20. WHAT ARE THE COMPONENTS OF THE VARIOUS COSTS YOU KNOW ABOUT?

21. FOR EACH PLANT GIVE THE RATIO OF 1973 TO 1972 FIGURES FOR EACH TYPE OF PRODUCTION COST AND OVERHEAD COST.

22. WHY WAS THERE SUCH A GREAT INCREASE IN OPERATING COST IN PLANT 0?

23. PRINT EVERY PIECE OF INFORMATION YOU HAVE CONCERNING PLANT 0 IN 1972 AND 1973.

24. DISREGARDING PLANT 0 TOTALLY, WHAT IS THE DIFFERENCE IN TOTAL PROFIT BETWEEN 1972 AND 1973?

25. WHAT IS THE DIFFERENCE IN PROFIT PERCENTAGE?

SUBJECT 5

The subject had an M.B.A. from the Harvard Business School and five years of experience; one year as an engineer and four as a manager.

1. WHAT WAS THE PERCENTAGE INCREASE IN OVERHEAD COSTS BETWEEN 1972 AND 1973?

2. WHAT WAS THE PERCENTAGE INCREASE IN THE AVERAGE PRICE PER PRODUCT?

3. WHAT WAS THE AVERAGE INCREASE IN THE COST PER PRODUCT?

4. HOW DID THE PRODUCT MIX CHANGE?

5. WHAT WERE THE GROSS MARGIN ON EACH PRODUCT

IN 1972 AND 1973?

6. WHAT IS PROFIT AS A PERCENT OF SALES IN 1972 AND 1973?

7. ARE TRANSPORTATION COSTS INCLUDED IN OVERHEAD OR COST OF GOODS SOLD?

8. WHAT COMPONENTS OF THE OVERHEAD COSTS GO UP MORE THAN 2% ?

9. WHAT WAS OVERHEAD COST AS A PERCENT OF SALES IN 1972 AND 1973?

10. WHAT WAS THE INCREASE IN INTEREST COST DUE TO?

11. WHAT WOULD HAVE 1973 PROFITS HAVE BEEN COMPARED TO 1972 IF THE PRODUCT MIX HAD NOT CHANGED IN THOSE TWO YEARS?

12. HOW MUCH DID AMOUNT BORROWED GO UP BETWEEN 1972 AND 1973 AND HOW MUCH DID THE AVERAGE INTEREST RATE GO UP?

13. WHAT PERCENT OF OVERHEAD COST IS INTEREST COST AND WHAT PERCENTAGE IS OPERATING COSTS?

14. HOW MUCH DID OPERATING COSTS GO UP BETWEEN 1972

AND 1973?

15. WHAT WERE THE FIVE LARGEST DOLLAR INCREASES IN OPERATING COSTS BETWEEN 1972 AND 1973?

16. HOW MUCH WAS THE DOLLAR INCREASE IN OPERATING COSTS AND INTEREST COSTS?

17. I KNOW WHAT THE PROBLEM IS.

SUBJECT 6

The subject was in the first semester of the Sloan Fellows program at the Sloan School of Management, M.I.T. He had five years of management experience but no previous training in management.

1. WHAT IS THE TOTAL SALES FOR THE PAST THREE YEARS?

2. WHAT WAS THE NET PROFIT IN 72 AND 73?

3. WHAT WAS THE TOTAL COST OF RAW MATERIAL IN 71, 72, 73?

4. WHAT WAS THE DIVIDEND PAID IN 1971, 1972, 1973?

5. WHAT IF ANY ARE OUTSTANDING LOANS, 1971, 1972, 1973?

6. WHAT WAS THE INTEREST RATE, 1971, 1972, 1973?

7. WHAT ARE THE OUTSTANDING SHARES?

8. ANY EQUIPMENT PURCHASED FOR LONG TERM DEPRECIATION?

9. DO YOU HAVE INFORMATION ABOUT WHAT THESE LOANS ARE FOR?

10. DO YOU HAVE LABOR COST FOR FINISHED PRODUCTS?

11. WHAT WAS THE TOTAL LABOR COST FOR 1971, 1972, 1973?

12. ARE WE FACING INFLATION?

13. WHAT ARE WE DOING WITH THE $16 MILLION LOAN?

14. OK I THINK I KNOW WHAT THE PROBLEM IS.

SUBJECT 7

The subject was a Ph.D. student at the Sloan School of Management, M.I.T. He had no work experience other than summer jobs.

1. WHAT WAS TOTAL OVERHEAD IN 1973?

2. WHICH YEARS DO YOU HAVE COSTS FIGURES FOR?

3. WHAT WERE THE OVERHEAD COSTS IN EACH OF THE LAST FIVE YEARS?

4. WHAT WERE TOTAL SALES IN EACH OF THE PAST FIVE YEARS?

5. WHAT WAS THE MOST PROFITABLE PRODUCT IN 1973?

6. YES.

7. WHAT WERE THE PROFIT MARGINS ON EACH PRODUCT IN 1972?

8. HOW MUCH OF EACH PRODUCT WAS PRODUCED IN 1972 AND IN 1973?

9. HOW MUCH DID THE INVENTORY LEVEL OF EACH PRODUCT CHANGE IN 1973 FROM 1972?

10. WHAT WERE THE SALES FOR EACH PRODUCT IN 1972 AND IN 1973?

11. WHAT WERE PROFITS IN EACH OF THE LAST FIVE YEARS?

12. WHAT WERE TOTAL SALES IN THE LAST 5 YEARS?

13. CANCEL THAT QUESTION.

14. WHAT WERE SALES OF EACH PRODUCT IN 1971?

15. WHAT WERE THE PERCENTAGE INCREASES IN SALES FOR EACH PRODUCT IN 1972 AND 1973?

SUBJECT 8

The subject was an industrial engineer with sixteen years of management experience.

1. WHAT IS THE OVERHEAD COST FOR EACH TYPE OF BATTERY?

2. OVERHEAD FOR 1972.

3. DIFFERENCE IN OVERHEAD FROM 1972 TO 1973?

4. LIST THE PRODUCT MIX FOR 1972 AND 1973.

5. GIVE ME THE PROFIT MARGIN ON EACH PRODUCT FOR 1972 AND 1973.

6. WHAT IS THE COST OF EACH PRODUCT FOR 1972 AND 1973?

7. WHAT WERE SALES FOR EACH PRODUCT IN 1972 AND 1973?

8. GIVE ME THE BUDGET FOR EACH PLANT AND THE OVERRUN IF ANY.

9. PRODUCTION COSTS.

10. WHAT IS THE OVERHEAD BUDGET FOR EACH PLANT?

11. WHAT OVERHEAD COSTS WERE INCURRED BY EACH PLANT?

12. WHAT IS THE PERCENT OVERHEAD OVERRUN AT EACH PLANT?

13. GIVE ME SALES PERCENT INCREASE AT EACH PLANT FOR 1973 OVER 1972.

SUBJECT 9

The subject was a Ph.D. student at the Sloan School of Management, M.I.T. He had no work experience other than summer jobs.

1. WHAT INFORMATION DO YOU HAVE ON COMPETITION?

2. DO YOU HAVE ANY INFO ON PRODUCTION COSTS?

3. TABLE DIRECT COST, OVERHEAD COST, AND CONTRIBUTION

PER UNIT SOLD FOR 1973.

4. HOW DO YOU DEFINE MARGIN?

5. TABLE SALES IN UNITS SOLD, PRODUCT MIX, DIRECT COST PER UNIT AND OVERHEAD COST PER UNIT FOR THE LAST 4 YEARS.

6. TABLE UNIT SALES, DIRECT MANUFACTURING COST, MARGIN AND PRODUCT MIX FOR THE LAST 2 YEARS BY PRODUCT.

7. DO YOU PERFORM MATHEMATICAL COMPUTATIONS?

8. PLEASE COMPUTE THE FOLLOWING: PERCENT CHANGE IN UNIT SALES, PERCENT CHANGE IN UNIT PRODUCTION COST FROM 1972 TO 1973.

9. BY PRODUCT, PLEASE.

10. DO YOU HAVE A FORECASTING MODEL FOR DEMAND?

11. DO YOU HAVE ANY MODEL AT ALL?

12. LIST THE FUNCTIONS YOU CAN PERFORM.

13. ARE THERE ANY VARIANCES BETWEEN ACTUAL PRICES CHARGED OUR CUSTOMERS AND THE GUIDELINE PRICES?

14. PLEASE TABLE THEM FOR PRODUCT 4 FOR THE 5 MAJOR CUSTOMERS.

15. LIST ACTUAL SALES PRICE AND GUIDELINE PRICE BY PRODUCT.

16. DO YOU HAVE A MODEL TO MAXIMIZE CONTRIBUTION TO THE COMPANY SUBJECT TO PRODUCTION AND OTHER CONSTRAINTS?

17. WHY WERE THE QUOTATION PRICES LOWER THAN LIST PRICES IN 1973?

18. HAVE THEY BEEN THIS WAY FOR THE PAST YEARS TOO?

19. NO. PLEASE GIVE ME THE OVERHEAD COST FOR 1972.

21. TABLE PROFIT BEFORE TAX FOR 1972 AND 1973.

22. COMPUTE PROFIT FOR 1972 AND 1973 ACCORDING TO THE FOLLOWING FORMULA: ACTUAL UNIT SALES BY PRODUCT TIMES LIST PRICE MINUS PRODUCTION COST FOR THE PRODUCT SUMMED OVER ALL PRODUCTS LESS OVERHEAD COST FOR THE YEAR.

23. I THINK I UNDERSTAND THE PROBLEM. THANK YOU.

SUBJECT 10

The subject was a Ph.D. student at the Sloan School of Management, M.I.T. He had no work experience other than summer jobs.

1. WHAT WAS THE CONTRIBUTION MARGIN OF PRODUCT 1 IN 1973?

2. WHAT WERE THE ACTUAL AND BUDGETED CONTRIBUTION MARGINS OF PRODUCTS 1, 2, 3, 4, 5 IN 1973?

3. WHAT WERE THE CONTRIBUTION MARGINS FOR PRODUCTS 1, 2, 3, 4, 5 IN 1972?

4. GIVE THE PRODUCT MIX IN 1972 AND 1973.

5. GIVE THE ACTUAL AND BUDGETED OVERHEAD COSTS IN 1973 AND THE ACTUAL OVERHEADS IN 1972 FOR EACH PLANT.

6. GIVE TOTAL CONTRIBUTION FIGURE FOR 1972 AND 1973.

7. CANCEL.

8. GIVE TOTAL PROFIT FIGURE IN 1973 AND 1972.

9. GIVE LIST AND ACTUAL PRICES FOR ALL PRODUCTS IN 1973.

10. GIVE ACTUAL PRICES FOR ALL PRODUCTS IN 1972.

11. DO YOU HAVE A BREAKDOWN OF OVERHEAD COSTS?

12. GIVE THE BREAKDOWN OF ACTUAL AND BUDGETED OVERHEAD COSTS FOR PLANTS 0, 2, 4.

13. GIVE ACTUAL AND BUDGETED OPERATING COSTS FOR ALL PLANTS, AND ACTUAL AND BUDGETED MANAGEMENT SALARIES AND INTEREST COSTS.

14. GIVE THE BUDGETED PROFIT.

15. DO YOU HAVE DATA ON TRANSPORTATION COSTS?

16. DO YOU HAVE THE DATA BY PLANT?

17. GIVE BUDGETED AND ACTUAL TRANSPORTATION COST BY PLANT.

18. GIVE BUDGETED AND ACTUAL SALES REVENUE.

19. GIVE BUDGETED AND ACTUAL INVENTORY.

20. GIVE BUDGETED AND ACTUAL SELLING COSTS.

SUBJECT 11

The subject was the manager of an operations research department with five years of staff experience.

1. HOW FAR BACK DOES YOUR INFORMATION GO?

2. WHAT WAS THE % OF OVERHEAD IN EACH OF THE LAST FIVE YEARS?

3. PERCENT OF OVERHEAD TO SALES.

4. WERE THERE ANY CHANGES IN THE PRODUCT MIX IN TERMS OF SALES DOLLARS?

5. WHAT WERE THE PROFIT MARGINS OF THE FIVE BATTERIES IN THE LAST TWO YEARS?

6. WHAT ARE THE HANDLING COSTS ASSOCIATED WITH EACH PRODUCT AND DID THEY CHANGE OVER THE LAST TWO YEARS?

7. HANDLING COSTS ARE COSTS ASSOCIATED WITH PRODUCTS THAT ARE NOT REFLECTED IN DIRECT MFG COSTS.

8. WHAT ARE THE ACTUAL SELLING PRICES OF THE FIVE BATTERIES?

9. HOW MUCH WAS THE ADDITIONAL REVENUE RECEIVED FROM THE 20% SALES INCREASE AND WHERE WAS IT SPENT?

10. THE INTENT OF MY QUESTION IS TO FIND OUT IF YOU KNOW IF YOUR ACCOUNTING METHODS CAN RELATE THE CHANGES IN SALES TO CHANGES IN YOUR EXPENSE STRUCTURES. DOES THIS HELP?

11. PLEASE GIVE ME CHANGES IN EACH TYPE OF COST ASSOCIATED WITH EACH PRODUCT.

12. IN AS MUCH AS ALLOCATING COSTS IS A TOUGH JOB I WOULD LIKE TO HAVE THE TOTAL COSTS RELATED TO EACH PRODUCT. I MEAN I WOULD LIKE THE COST OF EACH PRODUCT BROKEN DOWN ON A DIRECT AND INDIRECT BASIS.

13. WHAT WAS THE TOTAL PRODUCTION COSTS IN THE MOST RECENT TWO YEARS?

SUBJECT 12

The subject was a Ph.D. student at the Sloan School of Management, M.I.T. He had four years of management experience.

1. HAVE ANY PLANTS BEEN SUPPLYING BATTERIES TO OTHER THAN

NORMAL CUSTOMERS IE OUTSIDE OF THEIR NORMAL SALES DISTRICT?

2. PLEASE DISPLAY OVERHEAD FIGURES (ACTUAL AND BUDGET) FOR ALL PLANTS FOR THE PAST FOUR YEARS.

3. WHICH PLANTS WERE OVER BUDGET ON OVERHEAD BY MORE THAN 5% ?

4. PLEASE DISPLAY THE OVERHEAD BUDGET VARIANCE IN PERCENT AND ABSOLUTE $ FOR PLANTS 0, 2, AND 4.

5. WHICH PLANTS WERE OVER BUDGET ON FIXED COSTS BY MORE THAN 5 % ?

6. DISPLAY THE PROFITABILITY OF EACH PLANT AS A PERCENT OF SALES.

7. DISPLAY PROFIT FOR EACH PLANT DIVIDED BY PLANT SALES.

8. DISPLAY SALES REVENUES FOR ALL PLANTS FOR THE PAST FOUR YEARS.

9. DISPLAY AVERAGE COMPANY WIDE PROFITABILITY FOR THE LAST FOUR YEARS (%).

10. YES.

11. WHY IS THERE SUCH A DIFFERENCE BETWEEN THE COMPANY WIDE AVERAGE PROFITABILITY AND THE PROFITABILITY OF THE INDEPENDENT PLANTS?

12. I SUGGEST WE GET RID OF PLANT ZERO!

13. HAS PRODUCT MIX CHANGED IN ANY PLANT WHOSE PROFITABILITY HAS FALLEN OFF?

14. HAS PRODUCT MIX CHANGED BY MORE THAN 1 % IN ANY PLANT WHOSE PROFITABILITY HAS DECREASED?

15. DISPLAY THE DIRECT COST VARIANCE (ABSOLUTE $ AND %) FOR ALL PLANTS.

16. HAS THERE BEEN A DECREASE IN CONTRIBUTION MARGINS FOR ANY PRODUCT?

17. DISPLAY THE PERCENTAGE OVERHEAD GROWTH FOR EACH PLANT FOR THE PAST FOUR YEARS.

18. DISPLAY THE OVERHEAD DIVIDED BY SALES (%) FOR EACH PLANT.

19. WHY ARE THE OH FIGURES FOR PLANTS 2 AND 4 HIGHER THAN FOR 1 AND 3?

20. HAS THE PROFITABILITY OF ANY PLANT DECREASED?

21. WHICH ONE(S)?

22. DISPLAY THE MARGINS FOR PLANT 2 FOR THE PAST 4 YEARS.

23. DISPLAY THE DIFFERENCE BETWEEN LIST PRICE AND ACTUAL COSTS (DIRECT + OVERHEAD) DIVIDED BY LIST PRICE FOR PLANT 2 FOR THE PAST FOUR YEARS.

24. YES.

SUBJECT 13

The subject was a C.P.A with eight years of experience in accounting.

1. GIVE ME THE BREAKDOWN OF OVERHEAD EXPENSES FOR THE YEARS 1972 AND 1973.

2. GIVE ME COMPARATIVE NUMBERS FOR OPERATING COSTS FOR THE YEARS 1972 AND 1973.

3. WHAT WAS THE PROFIT MARGIN FOR THE YEAR 1972?

3. YES.

4. WHAT WAS THE SALES REVENUE BY PRODUCT FOR THE YEAR 1972?

5. GIVE ME THE SAME REVENUE FIGURES FOR THE YEAR 1973.

6. GIVE ME THE ACTUAL COST VS BUDGETED COST FOR EACH PRODUCT FOR THE YEARS 1972 AND 1973.

7. YES.

8. GIVE ME COMPARATIVE FIGURES FOR MANAGEMENT SALARY, INTEREST COSTS, AND DEPRECIATION FOR 1972 AND 1973.

9. WHAT WERE THE GROSS PROFIT FIGURES FOR THE YEARS 1972 AND 1973?

10. WHAT WERE THE COMPARATIVE FIGURES FOR SALES REVENUE VS DIRECT COSTS FOR THE YEARS 1972 AND 1973?

11. YES.

12. GIVE ME A BREAKDOWN OF DIRECT COSTS AND OVERHEADS FOR EACH PLANT IN THE YEARS 1972 AND 1973.

13. GIVE ME A BREAKDOWN OF BUDGETED DIRECT COSTS AND OVERHEADS FOR EACH PLANT FOR THE YEARS 1972 AND 1973.

14. GIVE ME PLANT 0 PRODUCTION COST FIGURE FOR THE YEARS 1972 AND 1973.

15. BY WHAT PERCENT DID THE OVERHEAD EXPENSES IN 1973 INCREASE OVER THOSE IN 1972?

16. WHAT WERE THE COMPARATIVE FIGURES FOR OVERHEAD EXPENSES FOR THE YEARS 1972 AND 1973?

17. GIVE ME DETAILS OF HOW THE ADDITIONAL SALES REVENUE IN 1973 WAS SPENT.

18. WHAT WAS THE PRODUCT MIX IN THE SALES FOR THE YEARS 1972 AND 1973?

SUBJECT 14

The subject was a masters student at the Sloan School of Management, M.I.T. He had no work experience.

1. WHAT WERE THE SELLING PRICES OF EACH PRODUCT?

2. WHAT WERE AVERAGE MANUFACTURING COSTS FOR EACH PRODUCT?

3. WHAT WERE UNIT PRODUCTION COSTS FOR EACH PRODUCT IN THE PREVIOUS YEAR?

4. WHAT WERE THE RELATIVE PERCENTAGES SOLD OF EACH PRODUCT IN 1972 AND 1973?

5. WHAT WERE AVERAGE QUOTATION PRICES FOR EACH PRODUCT IN 1972?

6. WHAT WERE BUDGETED COSTS FOR EACH PRODUCT IN 1972 AND 1973?

7. BOTH.

8. DO YOU HAVE BUDGETED PRODUCTION COSTS ON A PER UNIT BASIS?

9. WHAT QUANTITY OF PRODUCT 1 WAS SOLD BY ALL PLANTS IN 1973?

10. WHAT WERE CONTRIBUTION MARGINS FOR EACH PRODUCT

IN 1972 AND 1973?

11. WHAT ARE THE RELATIVE PERCENTAGES OF EACH PRODUCT SOLD BY EACH PLANT?

12. WHAT ARE THE RELATIVE PERCENTAGES OF SALES BY EACH PLANT?

13. DO YOU HAVE LIST PRICES FOR EACH PRODUCT?

14. WHAT WERE THEY IN 1972 AND 1973?

15. GIVE ME A BREAKDOWN OF DIFFERENCE BETWEEN LIST AND AVERAGE QUOTED PRICE FOR EACH PRODUCT FOR 1972 AND 1973.

SUBJECT 15

The subject was an undergraduate student in the department of Electrical Engineering and Computer Science, M.I.T. He had no work experience.

1. HOW MANY PLANTS ARE THERE?

2. WHICH OF THE FOUR PLANTS HAD THE LARGEST VALUE FOR TOTAL SALES IN 1973?

3. AT PLANT 2, WHICH PRODUCT ACCOUNTED FOR THE LARGEST PERCENTAGE OF TOTAL SALES IN DOLLARS?

4. DOES PRODUCT 2 ALSO ACCOUNT FOR THE LARGEST PERCENTAGE AT PLANT 4?

5. WHAT WAS THE TOTAL OVERHEAD OF PRODUCTION FOR PRODUCT 2 AT PLANT 2 IN 1973?

6. SUBSTITUTE "DIRECT MANUFACTURING COST" FOR "OVERHEAD OF PRODUCTION" IN PREVIOUS INPUT.

7. WHAT IS THE NUMBER OF UNITS OF PRODUCT 2 PRODUCED AT PLANT 2 IN 1973 TIMES THE UNIT COST OF PRODUCT 2?

8. DEFINE THE TERMS "UNIT COST" AND "UNIT PRICE".

9. WHAT WAS THE NUMBER OF UNITS OF PRODUCT 2 PRODUCED AT PLANT 2 IN 1973 TIMES THE UNIT PRICE OF PRODUCT 2?

10. HOW IS PROFIT COMPUTED?

11. CAN YOU PRODUCE A PROFIT FIGURE FOR A SPECIFIC PRODUCT AT A SPECIFIC PLANT IN 1973?

12. PRINT A TABLE CONTAINING UNIT COST AND UNIT PRICE FOR EACH PRODUCT AT PLANT 2 IN 1973.

13. COMPUTE UNIT COST FOR EACH OF THE PRODUCTS IN 1972.

14. WHICH UNIT PRICES WERE DIFFERENT IN 1972? PRINT THEIR VALUES.

15. WHAT WERE THE TOTAL OVERHEAD COSTS AT PLANT 2 IN 1972 AND 1973?

16. WHAT WERE TOTAL REVENUES AT PLANT 2 IN 1972 AND 1973?

17. HOW IS OVERHEAD COST COMPUTED?

18. LIST THE FIXED, NON-MANUFACTURING EXPENSES.

19. FOR EACH OF THE FACTORS JUST LISTED GIVE THE TOTAL VALUE INCURRED AT PLANT 2 IN 1972 AND 1973.

20. AT PLANT 2 LIST THE OPERATING COST INCURRED IN 1972 AND 1973.

21. FOR DEPRECIATION MANAGEMENT SALARY AND INTEREST COST

LIST THE AMOUNTS INCURRED IN 1972 AND 1973.

22. WHAT WAS THE OPERATING COST AT EACH PLANT?

CANCEL.

23. WHAT WAS THE PERCENT CHANGE IN OPERATING COST AT EACH PLANT FROM 1972 TO 1973?

24. IN 1973 WHAT PERCENTAGE OF THE DIRECT MANUFACTURING COST WAS ACCOUNTED FOR BY OPERATING COST?

25. WHAT WAS THE CHANGE IN TOTAL MANUFACTURING COST FROM 1972 TO 1973?

26. WHAT WAS THE PERCENT CHANGE IN TOTAL MANUFACTURING COST FROM 1972 TO 1973?

27. WHAT WAS THE PERCENT CHANGE IN TOTAL REVENUES FROM 1972 TO 1973?

28. WHAT WAS THE PERCENT CHANGE IN TOTAL OVERHEAD COSTS FROM 1972 TO 1973?

29. DEFINE P-COST TO BE THE SUM OF OVERHEAD COST AND MANUFACTURING COST. WHAT PERCENTAGE OF THE P-COST

IS ACCOUNTED FOR BY OVERHEAD COST?

30. FOR WHAT YEAR WAS THAT FIGURE?

31. GIVE ME THE SAME FIGURE FOR 1972.

32. HOW IS PROFIT COMPUTED?

33. HOW IS TOTAL COST COMPUTED?

34. ARE PRODUCTION COST AND MANUFACTURING COST THE SAME?

SUBJECT 16

The subject had a masters degree from the Sloan School of Management, M.I.T. He had eighteen months experience as a staff manager.

1. HELLO!

2. GIVE ME TWO TABLES, THE CONTRIBUTION MARGIN FOR ALL PRODUCTS IN EACH PLANT FOR THE YEARS 1972 AND 1973.

3. GIVE ME THE TOTAL SALES FOR 1972 AND 1973.

4. GIVE ME THE SALES VOLUME BY PRODUCT FOR THE

YEARS 1972 AND 1973.

5. GIVE ME THE FOLLOWING PROPORTIONS:
THE SALES OF PRODUCTS ONE, TWO AND FIVE DIVIDED BY THE TOTAL SALES FOR 1972 AND 1973.

6. GIVE ME THE AVERAGE COSTS AND THE BUDGETED COSTS FOR THE FIVE PRODUCTS FOR 1973 AND 1972.

7. UNIT COSTS.

8. GIVE ME THE DISTRIBUTION OF THE SALES OF PRODUCT FOUR BY PLANT.

9. DISTRIBUTION OF THE SALES OF PRODUCT 4 BY PLANT FOR THE YEAR 1972.

10. GIVE ME THE BUDGET FOR PLANT 4.

11. GIVE ME THE DIRECT COSTS AND THE OVERHEADS FOR 1972 AND 1973.

12. WAS THE ACTUAL OVERHEAD EXPENSE IN PLANT 4 HIGHER THAN THE BUDGETED AMOUNT IN 1973?

13. BY HOW MUCH?

14. SUPPOSE THE SALES IN 1973 HAD REMAINED UNCHANGED, WOULD THE PROFIT PICTURE HAVE ALTERED IF THE SELLING PRICE OF PRODUCT 1 HAD BEEN INCREASED TO ALLOW A PROFIT MARGIN OF $5.5, AND BY HOW MUCH? NEXT, WOULD THE SALES HAVE ALTERED SIGNIFICANTLY IF THERE HAD BEEN THIS PRICE INCREASE?

15. EVEN THOUGH THE PLANTS ARE NOT OPERATED AS PROFIT CENTERS, COULD YOU TELL ME THE CONTRIBUTION TO PROFITS FROM EACH PLANT FOR THE YEARS 1972 AND 1973?

16. GIVE ME THE SALES BY PRODUCT FOR PLANT TWO FOR THE YEARS 1972 AND 1973.

17. GIVE ME THE PROPORTIONAL INCREASE IN THE SALES OF THE VARIOUS PRODUCTS.

18. GIVE ME THE PRICES FOR THE VARIOUS PRODUCTS FOR THE LAST TWO YEARS.

19. CAN YOU GIVE ME THE PROPORTIONAL CONTRIBUTIONS TO PROFITS OVER THE LAST TWO YEARS FOR EACH PRODUCT?

20. NO.

SUBJECT 17

The subject had a masters degree from the Sloan School of Management, M.I.T. He had eight years of experience as a chemical engineer and two years of experience as a staff manager.

1. PLEASE GIVE ME THE SALES FOR 1969 70 71 72 AND 73.

2. TOTAL PROFIT MARGIN FOR 69 70 71 72 AND 73.

3. TOTAL PROFIT.

4. PROFIT MARGINS FOR EACH PRODUCT?

5. SALES FROM EACH PLANT DURING 73.

6. SALES FROM EACH PLANT DURING 72.

7. THE RATIO OF PRODUCTS COSTING $6.25 AND $5.00
FROM EACH PLANT DURING 72 AND 73.

8. CAN YOU GIVE ME DATA ON PRODUCT MIX FROM EACH PLANT?

9. GIVE ME THE OVERHEAD COSTS FROM EACH PLANT DURING 72 AND 73.

10. GIVE ME THE RATIOS OF OVERHEAD COSTS AND SALES FROM EACH PLANT FOR 72 AND 73.

11. GIVE ME THE RATIOS OF OVERHEAD COSTS AND SALES FOR PLANTS 1 2 3 4 FOR 72 AND 73.

12. FOR 72 AND 73.

13. GIVE RATIOS OF MANUFACTURING COSTS TO SALES FOR PLANTS 1 2 3 AND 4 FOR 72 AND 73.

14. GIVE PERCENTAGE CHANGE IN SALES FOR EACH PLANT FOR YEARS 72 AND 73.

15. GIVE PERCENTAGE CHANGE IN OVERHEAD COSTS FOR ALL PLANTS FOR YEARS 72 AND 73.

SUBJECT 18

The subject was a production manager with ten years of line experience and two years of staff experience.

1. WHAT IS TOTAL REVENUE FOR COMPANY ?

2. WHAT WAS THE COST OF GOODS SOLD?

3. I WANT THE SUM.

4. WHAT WAS THE NET INCOME?

5. WHAT IS THE COST FOR EACH PRODUCT IN EACH PLANT?

6. UNIT COST.

7. WHAT WAS THE ACTUAL UNIT COST CHANGE PER PRODUCT IN 1973 OVER 1972?

8. HOW MUCH DID OVERHEAD EXPENSE INCREASE IN 1973 OVER 1972 IN EACH PLANT?

9. WHAT IS PLANT 0?

10. WILL OUR CUSTOMERS PAY MORE FOR THE PRODUCT?
    CANCEL.

11. WHAT WAS THE VOLUME INCREASE PER PRODUCT IN 1973 OVER 1972?

12. WHAT WAS OVERHEAD INCREASE PER LOCATION IN

12. WHAT WAS OVERHEAD INCREASE PER LOCATION IN 1973 OVER 1972?

13. WHO ARE MY CUSTOMERS AND WHAT ARE THEIR VOLUMES PER CUSTOMER?

14. WHAT IS THE PRICE OF EACH PRODUCT?

SUBJECT 19

The subject was in the first semester of the Sloan Fellows program at the Sloan School of Management, M.I.T. He had five years of experience as a staff manager mainly in the computer field. Some of his experience was with a psuedo English language system.

1. DISPLAY FOR 1972 AND 1973.

2. SALES, OVERHEAD, SELLING PRICE, OVERHEAD, DIRECT MANUFACTURING COST, AND PROFIT MARGIN FOR ALL TYPES.

3. REMEMBER THIS REQUEST (CALL IT REQUEST A).

4. CAN YOU FORMAT REPORTS?

5. PLEASE RESPOND TO REQUEST A FOR YEARS 1972 AND 1973.

6. DISPLAY SALES.

7. DISPLAY SALES FOR YEARS 1972 AND 1973 BY BATTERY TYPES.

8. CALL CHAS THE RATIO (OVERHEAD/SALES).

9. CONGRATULATIONS.

10. PLEASE RETAIN THE RESULTS OF SPECIFICATIONS
UNTIL I CHANGE THEM.

11. DISPLAY FOR YEARS 1972 AND 1973 SALES AND CHAS
BY BATTERY TYPE.

12. DISPLAY ((SALES IN 1972 - SALES IN 1973)/SALES IN 1972).

13. REMEMBER TO RETAIN SPECIFICATIONS OF PREVIOUS REQUESTS.

14. CALL LAST DISPLAYED QUANTITY "SALES GROWTH".

15. DISPLAY SALES GROWTH FOR ALL TYPES.

16. DISPLAY AVERAGE COST FOR 1972 AND 1973.

17. PRODUCTION COST AVERAGED OVER SALES.

18. AGAIN BY PRODUCT PLEASE.

19. DISPLAY COST OF GOODS SOLD FOR PRODUCT 1.

20. WHAT IS THE DIFFERENCE BETWEEN "PRODUCTION COST" AND "DIRECT MANUFACTURING COST"?

21. NO THEY ARE'NT.

22. GIVE ME DEFINITION OF MARGIN.

23. STANDARD COSTS?

24. LET SCVAR BE DIFFERENCE BETWEEN STANDARD COSTS AND PRODUCTION COSTS.

25. DISPLAY SCVAR AND SALES GROWTH FOR 1972 AND 1973. CANCEL.

26. DISPLAY SCVAR FOR ALL PRODUCTS AND ALL YEARS.

27. WHAT ARE MY EXPENSE CATEGORIES?

28. DISPLAY OVERHEAD.

29. LET ALLOC BE
((OVERHEAD/PRODUCTION COST)*TOTAL PRODUCTION COST)
FOR EACH PRODUCT.

SUBJECT 20.

The subject was a Ph.D. student at the Sloan School of Management, M.I.T. He had four years of experience as a mechanical engineer.

1. WHAT DATA DO YOU HAVE REGARDING OVERHEAD EXPENDITURES ?

2. WHAT DATA DO YOU HAVE REGARDING PRODUCTION COST ?

3. WHAT DATA DO YOU HAVE REGARDING PRODUCT MIX ?

4. DO YOU HAVE PRODUCTION COST PER UNIT FOR
EACH TYPE OF PRODUCT ?

5. PRINT PRODUCTION COST PER UNIT FOR PRODUCT 1.

6. PRINT LIST PRICE FOR PRODUCT 1.

7. PRINT TOTAL MANUFACTURING COST FOR PRODUCT 1.

8. PRINT TOTAL MANUFACTURING COST PER UNIT OF PRODUCT 1.

9. PRINT OVERHEAD COST PER UNIT OF PRODUCT 1.

10. WHAT WAS THE AVERAGE BUDGETED COST PER UNIT OF PRODUCT 1?

11. WHAT DOES THE AVERAGE BUDGETED COST PER UNIT INCLUDE ?

12. PRINT BUDGETED COST PER UNIT OF PRODUCTS 2, 3, 4.

13. PRINT DIRECT PRODUCTION COSTS PER UNIT FOR ALL PRODUCTS.

14. PRINT LIST PRICES PER UNIT FOR ALL PRODUCTS.

15. WHAT WAS THE EXPECTED CONTRIBUTION MARGIN FOR ALL PRODUCTS PER UNIT ?

16. WHAT WAS THE ACTUAL CONTRIBUTION MARGIN FOR ALL PRODUCTS PER UNIT?

17. WHAT WAS THE AVERAGE SELLING PRICE PER UNIT FOR ALL PRODUCTS?

18. WHAT WERE EXPECTED OVERHEAD COSTS?

19. WHAT WAS ACTUAL OVERHEAD COST?

20. WHAT WAS THE PLANNED PRODUCT MIX?

21. WHAT WAS ACTUAL PRODUCT MIX?

22. PRINT PRODUCTION COSTS PER UNIT AND PER PLANT FOR ALL PRODUCTS.

23. DO YOU HAVE A LIST OF OVERHEAD COST FOR EACH PLANT SEPARATELY ?

24. PRINT THIS LIST.

25. WHAT IS PLANT 0 ?

26. DO YOU HAVE A LIST OF PRODUCTION COST ITEMIZED PER TYPE OF DIRECT COSTS?

27. WHAT WAS THE BUDGETED DIRECT MATERIAL COST ?

28. WHAT WAS DIRECT MATERIAL COST?

29. WHAT WAS LABOR COST ?

30. WHAT WAS TRANSPORTATION COST?

31. WHAT WAS MATERIAL COST IN 1972?

32. WHAT WAS LABOR COST IN 1972?

33. WHAT WAS TRANSPORTATION COST IN 1972?

34. DO YOU HAVE RECORDS ON SALES PER MAJOR CUSTOMER
IN 1972 AND 1973?

SUBJECT 21

The subject was a masters student at the Sloan School of Management, M.I.T. He had fifteen months of experience as an econometric consultant using a psuedo English language system.

1. LIST DATA AVAILABLE.

2. PRINT THE UNIT COST FOR BATTERY TYPE 1 AT EACH PLANT.

3. LIST ACTUAL AND BUDGETED UNIT COSTS FOR PRODUCT 1
FOR 65 TO 73.

4. LIST THE DATA FOR THE LAST 5 YEARS FOR EACH PRODUCT
BY UNIT COST.

5. DEFINE EQUATION
DISCOUNT(X) =
(LIST PRICE(X)-SELLING PRICE(X))/(LIST PRICE(X)).

6. SOLVE DISCOUNT (PRODUCT 1).
PRINT DISCOUNT (PRODUCT 1).

7. SOLVE DISCOUNT (PRODUCT 2), THEN PRINT DISCOUNT (PRODUCT 2).

8. SOLVE DISCOUNT (PRODUCT 3), THEN PRINT ANSWER.

9. SOLVE DISCOUNT (PRODUCT 4), PRINT ANSWER.

10. SOLVE DISCOUNT (PRODUCT 5), PRINT ANSWER.

11. PRINT PROFIT MARGIN FOR EACH PRODUCT FOR 72 AND 73.

12. DEFINE
%SALES(X)=(TOTAL SALES PRODUCT(X))/(TOTAL COMPANY SALES).

13. SOLVE %SALES(X) FOR EACH PRODUCT FOR 72 AND 73.

14. PRINT THE NUMBER OF UNITS OF EACH PRODUCT PRODUCED BY PLANT.

15. PRINT TOTAL SALES VOLUME BY PLANT.

16. LIST PROFIT MARGINS BY PLANT.

17. LIST PRODUCTION COSTS BY PLANT.

18. LIST OVERHEAD COSTS BY PLANT.

19. DEFINE

    %CHOVERHEAD(T) =

        (OVERHEAD(T) - OVERHEAD(T-1))/(OVERHEAD(T-1)).

20. PRINT %CHOVERHEAD(73) FOR EACH PLANT.

21. WHY ARE THERE 5 PLANTS?

22. WHAT WERE THE MAJOR INCREASES IN OVERHEAD IN PLANT 1?

23. GIVE DOLLAR FIGURES FOR OVERHEAD EXPENSES FOR PLANT 1.

24. ITEMIZE OVERHEAD COSTS FOR PLANT 1.

25. DEFINE %CH (ITEM T)=(ITEM(T)-ITEM(T-1))/(ITEM(T-1)).

26. LET ITEM BE DEPRECIATION, AND T BE 73.

27. PRINT THE LAST ANSWER.

28. LET ITEM BE OPERATING COST.

29. LET MANAGEMENT SALARIES BE ITEM.

30. LET ITEM BE INTEREST COST.

31. LET ITEM BE OPERATING COST BY PLANT.

32. WHAT MAKES UP OPERATING COSTS?

33. LET ITEM BE ENTERTAINMENT EXPENSES.
PRINT FOR TOTAL, AND EACH PLANT.

34. LET ITEM BE INTEREST COST BY PLANT.

SUBJECT 22

The subject had an M.B.A. and five years of experience. Two of these were as an engineer and three as a manager.

1. WHAT WERE THE OVERHEAD EXPENSES IN 1973?

2. WHAT WAS THE PERCENTAGE INCREASE IN OVERHEAD COST, 1973 VS 1972?

3. WHAT WAS THE PERCENTAGE INCREASE IN FREIGHT AND DISTRIBUTION COSTS FOR THE SAME PERIOD?

4. WHAT WAS THE ACTUAL VALUE OF FREIGHT AND DISTRIBUTION COSTS IN 1973?

5. WAS THERE AN INCREASE IN TRUCKERS FEES IN 1973?

6. ARE ALL INCREASES FROM FREIGHT CARRIERS PASSED ON TO THE CUSTOMER?

7. IS TRANSPORTATION COST INCLUDED IN OVERHEAD?

8. WHAT WERE THE SALES BY PRODUCT (5 PRODUCTS) FOR 1972 AND 1973?

9. WHAT WAS THE TURNOVER BY PRODUCT FOR 1972 AND 1973?

10. DIVIDE COST OF SALES BY AVERAGE INVENTORY FOR EACH YEAR

FOR EACH PRODUCT AND GIVE US THE RESULT.

11. FOR 1972 AND 1973.

12. WHAT WAS THE PROFIT MARGIN FOR EACH PRODUCT FOR 1972 AND 1973?

13. WHAT WAS THE PERCENTAGE OF TOTAL SALES FOR EACH PRODUCT FOR 1972 AND 1973?

14. WHAT COST ITEMS ARE INCLUDED IN OVERHEAD COST?

15. WHAT WERE THE OVERHEAD COSTS FOR 1972 AND 1973 FOR EACH PLANT?

16. CAN YOU GIVE THE PERCENT OF TOTAL OVERHEAD COST OF EACH PLANT FOR 1972 AND 1973?

17. WHAT WAS THE PERCENT CHANGE 1972 VS 1973 FOR EACH PLANT?

18. DO YOU HAVE A MODEL FOR MEASURING CUSTOMER SERVICE?

19. DO YOU HAVE A COUNT OF THE NUMBER OF SALES REQUESTS AND THE NUMBER OF REQUESTS FILLED?

SUBJECT 23

The subject was an engineer with a degree in operations research and ten years of experience as a production engineer. He was taking his first, formal management courses at the time of the experiment.

1. WHAT TYPES OF DATA DO YOU HAVE?

2. IS REGION RECORDED BY PRODUCT?

3. IS REVENUE RECORDED BY PRODUCT?

4. WHAT ARE REVENUES FOR EACH PRODUCT?

5. WHAT ARE SALES BY PLANT?

6. WHAT ARE SALES BY PLANT BY PRODUCT?

7. CAN YOU SUBRACT 1972 SALES BY PLANT BY PRODUCT FROM 1973 SALES BY PLANT BY PRODUCT?

8. SUBTRACT 1972 SALES BY PLANT BY PRODUCT FROM 1973 SALES BY PLANT BY PRODUCT.

9. DID ANY PRODUCT COSTS EXCEED BUDGET IN 73?

10. BY PLANT BY PRODUCT WHICH COSTS EXCEEDED BUDGET?

11. WHICH PRODUCT OF THE FIVE HAD THE LARGEST PERCENTAGE VARIANCE?

12. IN 1972 WHICH PRODUCT OR PRODUCTS HAD LARGEST VARIANCES?

13. WHAT WERE 1972 AND 1973 PROFIT MARGINS BY PRODUCT?

14. CAN YOU GIVE UNIT COSTS BY PLANT BY PRODUCT?

15. WHAT WERE ACTUAL COSTS PER UNIT FOR PLANT TWO?

16. WHAT WERE UNIT COSTS FOR 1972?

17. WHAT WAS PRODUCT MIX BY PERCENT IN 72?

18. WHAT WAS PRODUCT MIX BY PERCENT IN 73?

19. WERE PRICES RAISED IN 1973 OVER 1972?

20. WHAT WERE 1972 AND 1973 PRICES FOR EACH PRODUCT?

APPENDIX III

THE KNOWLEDGE REQUIRED TO RESPOND TO THE SUBJECTS' REQUESTS

To provide some idea of the knowledge required in a domain-specific management-support system this appendix attempts to describe all of the knowledge required to understand and respond to the 496 requests received from the subjects. The knowledge is described by itself and as the contents of a hypothetical system capable of responding to the subjects' requests. Although similar in nature, this system should not be confused with the system described in the earlier part of the thesis. The knowledge required to parse the requests is not included. The parsing problem is well known and has been studied extensively [7,34,43,61,72,74,76]. The capabilities of our parser are described in Chapter 6. Thus, more accurately, this appendix describes the knowledge required to process the parsed requests. Since it attempts to be complete there is some overlap with earlier sections.

Generically, system knowledge is of two main types: properties of objects and events and procedures for performing different types of actions. OWL [44], the knowledge representation language, was used in the prototype mainly for the former purpose. Procedural knowledge is encoded into various sections of the processor. Philosophically, there is a duality between descriptive and procedural knowledge but certain types of knowledge are more efficiently represented in one form rather than the other. We have attempted to encode knowledge in a descriptive fashion as far possible.

To provide a flavor of the kind of processing carried out by the system and the knowledge it is based on, we describe below the analysis of the first request made by the first experimental subject. Some of the analysis will become clearer in light of the knowledge of various types described in the following sections. The request was:

"What was the percentage of overhead cost
to total sales for the last five years?"

The parse of the sentence indicates that it is a wh-question and, thus, asks for some kind of information. The system looks at the verb and finds it to be a kind of "be" verb. This indicates that the following main noun group will determine the nature of the information required. The system looks at this and finds it to be "the percentage". The "a-k-o" (short for "a kind of") property of the terminal noun is tested and found to have the value "function". The system now knows that a function of data or model values is required and subsequent analysis is based on this inference.

The system tests other properties of "percentage" to find the number of arguments required and where these may be found in the sentence. It finds that two arguments are required. It also finds a number of sentence patterns that may be used to specify them. It tests the request against each of these patterns and finds that it fits a pattern that expects the arguments in prepositional groups (PGs) starting with "of" and "to". It takes, therefore, "overhead cost" and "total sales" as the first and second arguments. Other PGs are tested for information related to the retrieval of data and "for the last five years" is found.

"Overhead cost" and "total sales" are now processed by the name matching routine. It recognizes them as noun groups referring to data known to the system and returns "OVERHEAD-COST" and "SALES" as the equivalent data names. The name matching routine is described in III.5. The PG "for the last five years" is processed by the key value assignment routine, described in III.6. It returns the list "year-1969 year-1970 year-1971 year-1972 year-1973" as the value assigned to the key "year". Since this is a request for a function of data, the system attempts to retrieve the value of the inputs to the function. It sends the data names and the key values obtained from the sentence to a retrieval routine. This finds that "OVERHEAD-COST" is stored by plant and by year. Since there is no specification for plant in the sentence it uses a general default and sums over all plants. It returns, therefore, a list of five numbers: the overhead costs for each of the five years. Similarly, since there are no specifications for plant, product or customer it sums "SALES" over them and returns another list of five sales figures. These two lists, along with the names of the inputs and the key specifications for each value, are sent to a control program that invokes the function routines and formats the data. It notices that the "percentage" is required for two equally long lists of numbers and invokes the "percentage" routine five times, once for each pair of corresponding numbers in the lists. Finally, it takes the five percentages and formats an answer that looks like:

PERCENTAGE: OVERHEAD-COST TO SALES

YEAR-1973 12.30 %
YEAR-1972 11.90 %
YEAR-1971 13.20 %
YEAR-1970 14.30 %
YEAR-1969 15.20 %

The system is now ready for the next question.

Every noun group that is contained in the subjects' requests can be classified into one of the following categories: data names, model names, names of functions of data or model values, names of keys over which data is stored and names of entities known to the system. The entities include, other than those named above, the system, The Battery Company and the user. Each of these categories is discussed exhaustively in one or more of the following sections.

III.1 DATA KNOWLEDGE

The subjects' requests ask for data by name and by using the appropriate verb. The data requested can be exhaustively classified into revenues, costs, prices, inventories, loans and average interest rates. Each of these words, as well as "sales", has a "a-k-o" property that is set to "data". The verbs "spend", "sell" and "produce" possess "noun-verb" properties that point to "cost", "sales" and "production" respectively. The verb "incur" indicates a cost, but the name of the cost must be completely specified in the sentence.

The only kind of revenues requested are sales. Aggregate costs are of two basic kinds: production costs and overheads. Production costs are broken down into direct labor, direct material and transportation cost. Some of the requests also ask for standard costs and unit production costs. Unit production cost is broken down into unit labor, unit material and unit transportation costs. Unit costs are the average direct costs of producing one battery. Standard costs are also subdivided into standard material cost, standard labor cost and standard transportation cost. Overhead costs are broken down into operating costs, interest cost, depreciation and management salaries. This information is also used to interpret and answer questions about the components of different types of data and where specific costs get included.

The requests also ask for list prices, at which the product is supposed to be sold, and average quotation prices at which it is actually sold. Some ask for inventories which are divided into product and material inventories.

All data is stored for the last five years of operation and as actual and budget, except for unit production costs for which standard costs serve as budget.

Figure III.1 presents all the data items known to the system in the form of a tree. All the words that are used to refer to data can be classified either as direct data names such as "cost" and "inventory" or as adjectives and classifiers such as "unit" that futher specialize the data name or, finally, as noun groups that the system processes as being equivalent in meaning to a known data name.

Each data item requested is specified as a noun group in the sentence. The name matching routine analyzes the noun group and attempts to match it to one of the data names mentioned above. Details of the matching process are described in III.5. If a match cannot be found the system replies that it does not have information about the noun group.

Revenues, costs, prices, inventories and loans are

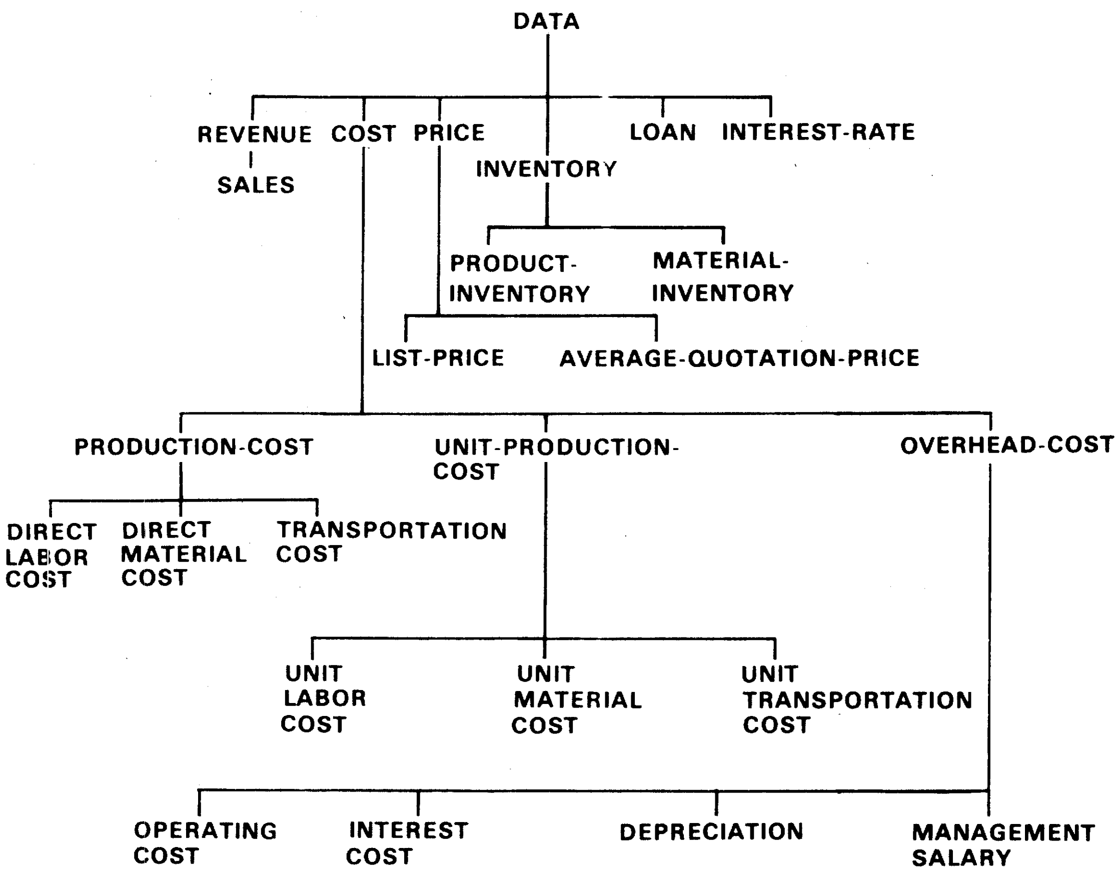


Figure III.I The Data Items Known To The System stored in dollars. Interest rates are stored as percentages. Sales are stored in an array by customer, plant, product and year. These are called its keys. Production costs and its components are stored in arrays by plant, product and year. Unit production costs are stored by product and year. Overhead costs and its components are stored by plant and year.

To retrieve a piece of data, key information for its retrieval must be specified in the sentence. Assignment of key values from information contained in the sentence also requires analysis of noun groups that specify plant, product, customer or year values. The key value assignment program analyzes these noun groups and sets up the values in appropriate registers. It is described in section III.6.

If all the key values are not specified, a set of defaults is used to fill in the missing information. For all data, except prices and unit and standard costs, a missing key, other than year, is summed over. For example, OVERHEAD-COST is stored by plant and year. If the request is for "overhead cost for 1973" the system provides the sum of overhead costs over all plants for 1973. For prices and unit and standard costs, if the product specification is missing, the data is provided for all products. If the year is omitted in a request for data it is assumed to be the last year for which the system has information, namely 1973. This is printed out as part of the answer.

The system contains a definition for each data item. If an explicit definition is not provided for a piece of data, knowledge about the keys on which it is stored and its components, if any, can be used to create a definition.

Requests for data are checked to contain keys on which the data is not stored. If this occurs, an error message is printed out accompanied by the definition of the data requested.

Other than the values of various types of data, some of the subjects' requests ask for information about their nature. Questions asking for the components of an aggregate cost or where a named cost gets included use the following nine constructions in the subjects' requests:

"What are the components of overhead cost?"

"Is transportation cost part of operating expenses?"

"Where does transportation cost get included?"

"Are shipping costs reflected in production costs?"

"Is transportation cost included in overheads?"

"What are the components of the various costs you know about?"

"Are transportation costs included in overhead or cost of goods sold?"

"What are my expense categories?"

"What makes up operating costs?"

Requests that inquire whether a certain piece of data exists can be answered by invoking the name matching routine. The subjects' requests use the following constructions to ask for available data:

Constructions Involving "data":

"What data do you have on operations?"

"What data items do you know about?"

"Do you have data by plant?"

"List data available."

"What types of data do you have?"

"Do you have data on transportation cost?"

"What data do you have regarding overhead expenses?"

"What data do you have regarding production cost?"

"What data do you have regarding product mix?"

Constructions Involving "information":

"Do you have any information on customer satisfaction?"

"Do you have any information on transportation cost?"

"Do you have any information about what these loans are for?"

"What information do you have on competition?"

"Do you have any information on production cost?"

Other Constructions:

"Do you have variable budgets?"

"Do you have a list of changes in sales force?"

"Do you have further breakdowns of overhead?"

"Do you have budgeted production costs?"

"Do you have production costs?"

"Do you have a list of production cost itemized per type of direct cost?"

The system responds to questions that ask whether a partiicular piece of data exists by providing its value if possible.

One of the subjects' questions asks for a property of data. It is similar to the questions about the properties of entities that are discussed in III.4

"Do overhead costs vary with volume?"

This requires a property called "variation"" to be associated with "overhead cost" containing a list of factors it varies with. In this case the list would be empty.

III.2 MODEL KNOWLEDGE

Some of the information that is requested requires the evaluation of models. Model names have their "a-k-o" property set to "model". The name matching routine described in III.5 attempts to match the noun group naming the model requested to model names known to the system. Linked to each model name is a list of inputs to a subroutine of the same name as the model. The inputs may be data or models. To produce a value for the model its data inputs are retrieved and its model inputs evaluated and fed into the subroutine. A model value that is requested with the adjective "budgeted" is calculated with budgeted figures for each of its inputs. The requests ask for the following models: "profit", "product mix", "turnover", "cost of goods sold", "contribution", and "contribution margin". Of these, "profit", "cost of goods sold", "contribution" and "contribution margin" are calculated in dollars. "Turnover" is a number and "product mix" is a set of percentages.

Figure III.2 lists all the models known to the system. The name matching routine attempts to match the noun group specifying the model to the model names known to the system. It is described in section III.5.

Each model name is associated with information indicating the cases wherein key information for retrieving its input data may be found. The key value assignment routine is described in III.6. If key values are missing a set of defaults are used. For "profit", "turnover", "cost of goods sold" and "contribution" if the customer, product, or plant specifications are missing the data is aggregated over them. "Product mix" is computed for all plants if the plant specification is omitted. Models have restrictions on key values they can be calulated for. "Profit" and "cost of goods sold" cannot be produced for all products since overhead cost is not allocated by product. Knowledge about key values over

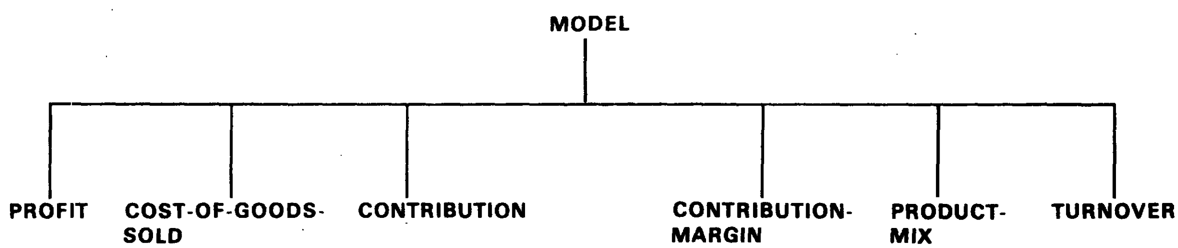


Figure III.2 The Models Known To The System which model values can or cannot be produced must also be known specifically to the system so that it can answer questions about it. The following two requests require such knowledge:

"What is the definition of profit for a product?"

"Can you compute a profit figure for a specific product at a specific plant?"

If the product specification is omitted in a request for "contribution margin" it is produced for all products. For all models, missing year specifications are assumed to be "1973".

In addition to model values, certain requests ask whether a model with given specification exists. There are three such questions among the subjects' requests.

"Do you have a forecasting model for demand?"

"Do you have a model to maximize contribution to the company subject to production and other constraints?"

"Do you have a model for measuring customer service?"

Each model has a definition associated with it. This is printed out if the question asks how the model (and in one case overhead cost) is "calculated", "computed" or "defined". If a model is requested with key specifications that are inappropriate, its definition is printed out.

III.3 KNOWLEDGE ABOUT FUNCTIONS OF DATA

The system also has the ability to compute named functions of data or model values. To respond to the requests it must be able to count and calculate percentages, increases, rates of increase, changes, differences, (accounting) variances, overruns, distributions and averages. It must also be able to compute a special function called "over budget" which is the excess of a named figure over the corresponding budgeted amount. The functions are referred to by these words in the subjects' requests except that "ratio" and "proportion" are treated as being equivalent to "percentage" and "deviation" as being equivalent to "difference". "Go up", "growth" and "dollar increase" are interpreted to mean the same as "increase" and "decrease" is implemented as "increase" with a change of sign at the output. Each function has knowledge associated with it about the number and nature of its arguments and the cases in the parsed sentence where the noun groups that name the data to serve as arguments will occur. The following paragraphs describe this knowledge for all the functions required to respond to the subjects' requests.

"Percentage" expects two numbers as arguments. Two ways in which their names are specified are in a pair of PGs starting with "of" and "to" and as the main noun group and the noun group in a PG starting with "of". In the latter case a "manner" PG starting with "as" and with the function as its main noun follows the main noun group. Examples are:

"What was the percentage of overhead costs to total sales for the last five years?"

"What are profit margins as a percentage of sales for each manufacturing installation?"

Another sentence pattern or convention for specifying inputs to "percentage" is a sentence that starts with "what percentage" followed by a PG starting with "of", a "be" verb and a noun group specifing the second argument:

"What percentage of overhead cost is operating cost?"

If percentages are required for a set of numbers then usually they are required to the total. This is exploited in a sentence pattern that specifies only the set of numbers:

"What percentage of each product is sold from each plant for the last five years?"

The system uses knowledge of this convention to sum the numbers and compute each percentage to that number.

"Increase" and "change" also require two numbers as arguments but they are always the same data item. Thus, only one noun group is used to specify them. Preposition groups specify the two sets of key values, typically as noun groups joined by an "and" in a PG starting with "between" or as two PGs starting with "from" and "to". One set of key specifications may be omitted, however, using the default that variation is required over the last complete time period.

Argument names can also occur as classifiers of the function name as in "salary increases", "percentage overhead growth", and "sales percent increase". "Rate of increase" and "percent (or percentage) increase" use the same conventions as "increase" for specifying data and the system can use the same knowledge for sentence analysis. A different function is used to operate on the data, however.

The two arguments required by "difference" normally come as noun groups joined by "and" in a PG starting with "between". They may also be specified as two PGs starting with "of" and "from". In some cases a single noun group is specified in a PG starting with "in" and two sets of key values in PGs starting with "to" and "from" or "between".

"Variance" is similar to "difference" except that the two data items are budgeted and actual values. Sometimes only one of them is specified and the system has to use knowledge about "variance" to infer the other. The noun group specifying the single input can appear as the adjectives and classifier of "variance" as in "overhead budget variance".

"Overrun", like "variance" takes an actual and a budgeted value as inputs. Both data names are rarely specified and the system uses knowledge to fill in the missing name. If the actual data is specified in a PG starting with "of" then the budgeted data, if specified, appears as "over budget". Arguments to "overrun" can also appear as classifiers as in "production cost overrun" and "percent overhead overrun".

"Over budget" expects only one argument and this occurs in a PG starting with "on" as in:

"Which plants were over budget on overhead by more than 5%?"

"Distribution" expects two arguments: a data name and a key. The first occurs in a PG starting with "of" and the second in a PG starting with "by".

"Average" is somewhat different from other functions provided by the system in that it expects a set of numbers of the same kind as input. The only form in which the data set can be specified is as a noun group that has "average" as a classifier. It can also appear with "average increase" in the same manner. The word "average" is also used to indicate unit prices and costs as in "average price per product". In fact, since each product is produced and sold in different amounts a simple average over cost and price has little meaning. Thus, such questions are considered ambiguous and the user is asked whether he would like unit costs. (See also III.7.)

Some of the subjects' requests invoke simple arithmetic functions such as "subtract", "divide" and "multiply". "Times" is used as a synonym for "multiply". "Divide" and "multiply" appear either as a past participle after the first argument followed by the second argument or as verbs followed by the first argument with the second argument being contained in a PG starting with "by". "Subtract" also appears as a verb followed by a first argument with the second argument being contained in a PG starting with "from".

> "Display profit for each plant divided by plant sales."
>
> "Divide cost of sales by average inventory for each year."
>
> "Subtract 1972 sales by plant by product from 1973 sales by plant by product."

The system also understands the comparatives "higher" and "more" and can answer yes-no questions in which two data names, or a data name and a number, optionally followed by key value PGs occur on either side of the comparative. The five sentences in the subjects' requests that make use of comparatives are:

> "What components of the overhead costs go up more than 2% ?"
>
> "Which plants were over budget on overhead by more than 5% ?"
>
> "Which plants were over budget on fixed costs by more than 5 % ?"
>
> "Has product mix changed by more than 1 % in any plant whose profitability has decreased?"

"Was the actual overhead expense in plant 4 higher than the budgeted amount in 1973?"

III.4 ENTITY KNOWLEDGE

To respond to some of the subjects' requests the system must have knowledge about the properties of the the various entities referred to. These are the plants, the products, the customers, the years, each data item and model and the corporation and the system. This section specifies the knowledge required to answer the subjects' questions about the properties of entities.

The knowledge base stores values for named properties of each entity. Each of the above entities has an "a-k-o" property whose value indicates what it is. Generic entities (plant, product, customer, year, data and model) also have a property called "kinds" whose value is a list of all entities that are "a-k-o" it. To answer the question:

"What is plant 0?"

the system looks up its "a-k-o" property and finds that it is a plant. It also finds a piece of text that explains why plant 0 is special and different from the other plants. These two items of information make up the response to the question. To answer:

"How many plants are there?"

the "kinds" property for "plant" is checked and the elements in the value list are counted. Similarly,

"Who are my customers?"

"My" always refers to the corporation. So the system looks up the "kinds" property of "customer" and provides its value.

The "system" has a property named "calculate" with a value that is a list of the functions it can calculate. Thus, to answer the question

"Can you calculate percentages?"

the system looks up the "calculate" property for "system" ("you" invariably refers to the "system") finds "percentage" in its value list and returns the answer "yes".

The "system" also has the properties "do" and "know". The former contains as value the list: "answer questions", "calculate", "retrieve data" and "evaluate models". The latter contains a list of all the highest level data items known to the system. The "have" property has as value all the top level data items and models and the word "information". The question

"Do you have any model at all?"

is answered by looking up the "have" property and checking if its value list has any items that are "a-k-o" models. Similarly,

"List all data items you know about?"

is answered by looking up the value list of "know" and selecting the items that have the "a-k-o" property value "data". The question:

"What types of data do you have?"

is answered in the same way after "types of data" is analyzed and found to select the "data" value of the "a-k-o" property of all elements in the "have" property of "system". Similarly:

"List data available."

can be answered after realizing that as there is no indication of who the data is available with it must be the "system" by default. The system must also realize that if it "has" data it is "available". The following two questions can be answered in the same manner since the system knows that "perform" is equivalent to "do".

"Do you perform mathematical calculations?"

"List the functions you can perform."

Each data item and "information" have a property "duration" for which the value list is "year-1969 year-1970 ... year-1973". This is combined with special knowledge about the phrase "go back" to answer:

"How far back does your information go?"

The answer is "We have information for" followed by the value list. In the same way,

"Which years do you have cost figures for?"

can be answered by associating "year" with "duration" and looking up this property for the name "cost".

III.5 NAME MATCHING

The system contains a complex set of mechanisms for matching names of requested information to data and model names known to the system. If a match cannot be found the system replies that it does not have information about the noun group.

At the simplest level there is a equivalence and a noun idiom list and a routine that translates incoming words, or groups of words, to words known to the system. The following equivalences are required to respond to the requests received from the subjects:

For Data:

expense	for	cost
variable cost	for	production cost
manufacturing cost	for	production cost
direct manufacturing cost	for	production cost
data on operations	for	production cost
labor cost for finished products	for	labor cost
cost of raw material	for	material cost
unit cost	for	unit production cost
overhead	for	overhead cost
OH	for	overhead cost
fixed cost	for	overhead cost

non-manufacturing cost	for	overhead cost
price	for	list price
guideline price	for	list price
unit price	for	list price
outstanding loans	for	loans
amount borrowed	for	loans
quantities produced	for	production
inventory level	for	inventory
selling price	for	quotation price
average selling price	for	quotation price
revenue	for	sales
sales revenue	for	sales
volume	for	sales
sales volume	for	sales
company sales	for	sales
revenue for company	for	sales
freight cost	for	transportation cost
distribution cost	for	transportation cost

For Models:

net profit	for	profit
net income	for	profit
profit before tax	for	profit
profitability	for	profit
company wide profitability	for	profit

company wide average profitability	for	profit
cost of sales	for	cost of goods sold
margin	for	contribution margin
profit margin	for	contribution margin
gross margin	for	contribution margin
profit contribution	for	contribution
contribution to the company	for	contribution
relative percentages sold	for	product mix
product mix by percent	for	product mix

In General:

planned	for	budgeted
expected	for	budgeted
info	for	information

After the equivalence substitution, if the terminal noun in the noun group has an "a-k-o" property that is "data" or "model", a name matching routine tries to match the noun group and, if necessary, the information contained in an immediately following PG that starts with "of" or "for" with an existing data or model name. It looks for an exact match between the adjectives and nouns of the noun group and the data name. The adjectives "actual" and "total" are ignored and the noun in the PG is treated as a classifier.

If the terminal noun group has an "a-k-o" value that is not "data" or "model" then the name matching routine looks for special noun group constructions. To analyze the subjects' requests knowledge about the following special noun constructions must be included in the knowledge base. These specify the data or model name in the adjectives, classifiers and noun of an immediately following PG starting with "of" or "from" of a general noun such as "figure".

Constructions Involving "figure":

cost figures

overhead figures

contribution figure

revenue figures

profit figures

production cost figure

gross sales figures

gross profit figures

dollar figures for overhead expenses

comparative figures for management salary

comparative figures for interest cost

comparative figures for depreciation

comparative figures for sales revenue

comparative figures for overhead expenses

Constructions Involving "breakdown":

These ask for the components of a named cost.

breakdown of items in your overhead

breakdown of overhead cost

breakdown of actual overhead cost

breakdown of budgeted overhead cost

breakdown of overhead expenses

breakdown of direct costs

breakdown of budgeted direct costs

breakdown of difference between list and actual quoted price

further breakdowns of overhead

Other Constructions:

actual value of freight cost

actual value of distribution cost

records on sales

data on product mix

data on transportation cost

Finally, the requests include a few constructions that use a relative clause to specify the information required or contain redundant PGs that must be ignored. These are listed below along with their interpretation.

quantities that were produced	for	production
actual prices charged our customers	for	quotation price
overall profits on operations	for	profit
variable costs for		

manufacturing operations	for	production cost
production cost for one unit	for	unit production cost
contribution per unit sold	for	contribution margin

The noun groups used to specify data or models in the subjects' requests have been analyzed exhaustively and occur either in their own name or in one of the forms listed above.

III.6 KEY VALUE ASSIGNMENT

The problem of analyzing key information contained in various parts of the sentence and assigning the appropriate key values is similar to the problem of matching noun groups naming the information requested. The system must analyze information contained in the parse of the sentence to assign key values for plant, product, customer and year. Key values are typically specified in prepositional groups. The preposition is, however, not very useful in indicating the type of key variable to be specified. Plant specifications, for example, may be preceded by "at", "from" and "in" and the more general "per", "of" and "from". The nature of the noun group has to be analyzed before it can be associated with a particular key with confidence.

Key specifications can also occur in adjectives of the main noun group, relative clauses and participle constructions. The constructions used to specify plant, product, customer and year values in the subjects' requests are listed below:

Plant Specifications:

by plant

by plants

for each plant

by each plant

for each branch

from each plant

at each plant

from plant to plant

per location

at plant 2

in plant 2

for plant 2

for plants 2 and 4

for plants 1 2 3 and 4

for plants 1 2 3 4

for plants 1 2 and 3 and 4

for each plant separately

of the independent plants

plant sales

plant 0 production cost

Product Specifications:

by product

by product (5 products)

for each product

of each product

on each product

per product

for each manufacturing installation

of each manufacturing installation

for each type of product

for each type of battery

by battery types

by battery type

for battery type 1

for the various products

over all products

for all products

for any product

of the five batteries

for product 1 through product 5

for products 1 through 5

for products 1,2,3,4,5

for product 1 (one)

of product 4 (four)

of products one, two and five

of products 2,3,4

broken down by product

of products produced by plant 1

concerning plant 0

related to each product

associated with each product

which product of the five

which product or products

Customer Specifications:

per major customer

for the 5 major customers

Year specifications:

by year

for '71 (71, 1971)

for the year 1972 (for the years 1972 and 1973)

for each year

for each year studied

in (for) each of the last (past, previous)
2 (two) years

for the last 2 years, 1972 and 1973

in the previous year

in the most recent 2 years

in (for) 1972 and 1973

for both 1972 and 1973

in 1972 and (&) in 1973

for 1972 vs 1973

for 1969 through 1973

1973 sales

from 1972 to 1973

in 1973 over 1972

1971 1972 1973 1974 (71 72 73 74)

at the end of 1972

Key specifications may also occur in relative clauses. The only other examples that occur in the subjects' requests are: "all products produced by plant 3", "every piece of information you have concerning product cost", and "the various costs you know about".

III.7 DEDUCTION RULES AND KNOWLEDGE ABOUT SPECIAL CONCEPTS

Knowledge about specific concepts that are used in questions and deduction rules that are required to respond to questions whose answers cannot be retrieved directly from the data base is also required to respond to some of the subjects' requests. These are listed below along with the knowledge required to analyze them and respond to them. To answer the question:

"What is the percentage of repeat customers in 1973?"

the system needs to know that repeat customers can be obtained as the intersection of the customer lists for the last two years. Similarly, to analyze and respond to:

"Did one plant assume more production of batteries from other plants in 1973?"

the system must realize that "assume more production" refers to a shift in manufacturing ratios between plants. To answer:

"Do you have a list of changes in sales force for each branch?"

the system must know that changes in sales force are different from changes in other data such as cost. Another request is:

"What was the most profitable product in 1973?

This is a way of asking for the product with the highest profit and the system must be able to interpret "most profitable" correctly and use the profit model. Similarly, special knowledge is required to interpret:

"Have any plants been supplying batteries to other than normal customers ie outside of their normal sales district?"

To interpret:

"List actual and budgeted unit costs for product 1 for 65 to 73."

The system must know that "65" can be a year and that in this usage it does refer to an year. After this it is not difficult to generate a message as a result of the key assignment process that the system only has data from year-1969 to year-1973.

A few yes-no questions ask for two sets of data to be compared with a view to testing if a sub-problem exists. The discriminating function that decides whether the sub-problem exists may be quite vague, however. Such cases can be recognized by the use of the verbs "change" and "compare" and the adjectives "same" and "major". These notions are too vague to be operationalized by special knowledge and the system can only present the data and ask the subject to reach his own conclusions. In the case of "same", however, the system checks to see if the data is identical or synonymous i.e. "exactly the same". If it is, the system responds with a "yes". If it is not, however, it does not try and operationalize "almost the same" and presents the data to the subject.

To generate an answer to:

"Were prices raised in 1973 over 1972?"

the system requires the knowledge that "raise" is equivalent in meaning to "increase" if the data item is a kind of "price". A more difficult problem is raised by the requests:

"At plant 2 which product accounted for the largest percentage of total sales in dollars?"

"In 1973 what percentage of the direct manufacturing cost was accounted for by operating cost?"

The system must recognize that "accounted for" asks for the share of a particular key or component in the whole.

To answer the question:

"Was any equipment purchased for long

term depreciation?"

requires knowledge about the motivation of managers and the actions they can take to improve profitability or merely to make the figures look "good". The subject is asking whether equipment has been purchased towards the end of the year to depress profits, decrease taxes and improve cash flow. This sort of question moves into the area of inquiring about motivation rather than merely working with the data available.

III.8 AMBIGUOUS AND INCOMPLETE REQUESTS

Some of the questions asked by the subjects were ambiguous or omitted information that was essential to the generation of a response. The system has the knowledge to detect two kinds of ambiguities. First, if the second argument to a percentage function is omitted as in:

"What was the percentage of profit last year?"

The system must either know, as a default, that if profit is asked for as a percentage it is usually as a percentage of sales. Alternately it should respond by saying that it does not know what it should calculate profit as a percentage of. This uses the knowledge that percentage has two arguments and that the sentence specifies only one.

Second, requests for "cost" and "budget" are considered ambiguous since the names are not specific enough:

"What were budgeted costs for each product?"

"Give me the budget for each product and
the overrun if any?"

In each of these cases a number of budgets can be provided and a more specific request is required. Sometimes it is possible to use special knowledge to narrow down which cost or budget is really required. If the "cost" or "budget" is required by plant then it can either be overhead cost or production cost. The system can ask a question to clarify which of these the subject really wants. Similarly, "the cost for each product" may be production cost or unit production cost and the system can ask which is required.

The word "average" is used ambiguously in some requests. The "average manufacturing cost per product" may mean "unit cost for each product" or the average of the manufacturing costs for each product. As discussed earlier, the system suspects that the former is meant but asks a question for clarification.

III.9 QUESTIONS THAT CANNOT BE ANSWERED

Some of the requests obtained from the subjects cannot be adequately analyzed and responded to by the knowledge described above. These requests fall into a number of categories each with its special problems.

The four most common classes of requests that cannot be handled are declarative sentences providing information, model definitions and the use of defined models, questions asking for the reasons or motivations behind facts and actions and sentences that are "bad English" and cannot be parsed. The remaining requests that cannot be answered are listed below.

Some "what-if" questions cannot be answered because they require models that the system does not have.

> "What would have 1973 profits been compared to 1972 if the product mix had not changed?"
>
> "Disregarding plant 0 totally, what is the difference in profit between 1973 and 1972?"
>
> "Suppose the sales in 1973 had remained unchanged would the profit picture have altered if the selling price of product 1 had been increased to allow a margin of $ 5.5?"

Other types of questions cannot be answered because the system does not have the data or cannot provide the facilities. The request:

> "What was the number of units of product 2 produced at plant 2 in 1973 times the unit price of product 2?"

cannot be complied with because t e system does not have production figures in numbers of bat eries. Similarly, the system does not have the facilities or information to respond to:

"Are we facing inflation?"

"Will our customers pay more for the product?"

"Do you have a count of the number of sales requests and the number of requests filled?"

"In the future, please express numbers of over 100000 in terms of units of millions, and numbers over 100 but less than 100000 in units of thousands."

"Please retain the results of specifications until I change them."

"Remember to retain the specifications of previous requests."

Finally, the following questions cannot be answered because they are too difficult i.e. because the system does not possess the specialized knowledge required to analyze and respond to them.

"For each plant give the ratio of 1973 to 1972 figures for each type of production cost and overhead?"

"By what percent did the overhead expenses in 1973 increase over those n 1972?"

These are very special constructions and it does not seem important enough to include the knowledge necessary to analyze them in the system. Consider the pair of requests:

"Why were quotation prices lower than list prices in 1973?"

"Have they been this way for the past years too?"

The first question is quickly recognized as inquiring about causality and is, therefore, rejected. Since it is not analyzed the system cannot associate "this way" correctly, even if it had the special knowledge to do so. In:

> "Were there any changes in product mix
> in terms of sales dollars?"

the final "in terms of sales dollars" is intended to differentiate between product mix in terms of the number of batteries sold and in terms of dollar sales. The system does not have the knowledge to interpret this, however. Similarly, special knowledge is needed to interpret:

> "How much was the additional revenue received
> from the 20% sales increase and where was it spent?"
>
> "Give me details of how the additional sales
> revenue in 1973 was spent."
>
> "What are we doing with the $13 million loan?"

It is not difficult to incorporate the knowledge into the system that associates "spending revenue" and "doing something with a loan" with "sources and uses of funds" but there seems to little point in doing so since the data is not available. Special knowledge is also required to interpret

> "In as much as allocating costs is a tough job I would
> like to have the total costs related to each product
> I mean I would like the cost of each product
> broken down on a direct and indirect basis."

"Please display the overhead budget variance
in percent and absolute $ for plants 2 & 4.

"Do you have budgeted production cost on a
per unit basis?"

"For what year was that figure?"

"Even though plants are not operated as profit centers
could you tell me the profits from each plant
for the years 1972 and 1973?"

The system does not have the knowledge to analyze the redundant initial clause.

"List the data for the last 5 years
for each product by unit cost?"

The data required appears in a final PG that looks like a key value specification.

"What is the difference between plant 1 and plant 2
plant 3 and plant 4?"

"Has product mix changed in any plant whose
profitability has fallen off?"

"Has product mix changed by more than 1% in any
plant whose profitability has decreased?"

"The ratio of products costing $6.25 and $5.00
from each plant during 72 and 73?"

"Are all increases from freight carriers passed
on to the customer?"

"Do you have a count of the number of sales requests and the number of requests filled?"

"Which product of the five had the largest percentage variance?"

In each of the above cases it is not difficult to specify the additional knowledge that would be required to interpret it and respond to it. The issue is, however, whether these forms occur often enough or are important enough to justify the additional knowledge. We have judged that they are not.

BIOGRAPHICAL NOTE

Ashok Malhotra was born on November 14 1943 in Delhi, India. He graduated from the Modern High School, New Delhi in 1959. In August of that year he entered the University of Delhi and three years later received a B.Sc. (Honours) degree in Physics. In September 1962 he entered M.I.T. and received S.B. and S.M. degrees in Electrical Engineering in June 1964 and February 1966 respectively. In February 1967 he received the S.M. degree in Industrial Management from the then called School of Industrial Management. During his stay at M.I.T. he was elected to Tau Beta Pi, Eta kappa Nu and Sigma Xi.

In April 1967 Ashok Malhotra joined the Tata Consultancy Services in Bombay, India, a computer-based management consulting organization that had just been started by the Tata conglomerate. He worked there until July 1971 as part of its senior management and was responsible for the design and implementation of a number of management information systems and for several Operations Research and E.D.P. feasibility studies.

September 1971 found him back at M.I.T. as a fellow at the Sloan School of Management, workirg towards a doctorate in management. In January 1972 he married Patricia Coelho of Konkanadi, a district of Mangalore, India. Since June 1972 he has been employed by Honeywell Information Systems.