# Opening up the Black Box: A Multi-Method Investigation of Expert-Novice Differences for a Software Diagnosis Task With Implications for DSS Research and Design

Richard Glass

A Thesis

in

The Faculty

of

Commerce and Administration

Presented in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montreal, Quebec, Canada

April 1991

ISBN 0-315-68737-1

# Abstract

Opening up the Black Box: A Multi-Method Investigation of Expert-Novice Differences for a Software Diagnosis Task With Implications for DSS Research and Design

Richard Glass, Ph.D.
Concordia University, 1991

Research concerning the effectiveness of DSS has provided inconclusive results. One factor contributing to the inconclusiveness of the results is the tendency in DSS research and design to treat the decision making process as a black box. This thesis is devoted to eliciting the underlying decision processes of experts and novices for the performance of software diagnosis tasks in order to better understand how humans solve problems and make decisions and to apply this understanding to the research and design of computer aided decision support systems.

An experiment is conducted in which four experts and four novices each performs two software diagnosis tasks of varying task familiarity. The subjects are professional software technical diagnosticians, the tasks are typical of tasks encountered in their work environment and the research site is the subjects' place of employment. A multi-method research approach is adopted that employs: (i) non probing, concurrent verbal protocol analysis, (ii) aided retroactive verbal protocol analysis with probing and (iii) causal mapping.

The results of the experiment indicate that: (i) experts have a higher quality representation of the diagnosis problems than novices, (ii) experts are superior to novices for the performance of the problem solving tasks, (iii) there is no consensus at the global level within skill groups or between skill groups for the number of information items used during the problem solving sessions but experts do use significantly more "key" information items that are important for making an accurate diagnosis, (iv) experts tend to agree more within group on the use of information items than novices and (v) both experts and novices use a hypothetico-deductive strategy to diagnose software problems but experts are more efficient and more effective at applying the strategy than novices. The implications of these results for better understanding human decision making processes and for the research and design of computer aided decision support systems is discussed.

# Acknowledgements

In an undertaking of the magnitude of a Ph.D. dissertation, a candidate is influenced and supported by a number of individuals. I would particularly like to acknowledge the commitment and support that I received from my committee members, Dr. Peter Wade (supervisor), Dr. Gary Johns, Dr. Jerry Tomberlin and Dr. Tony Marley. I not only have the utmost academic respect for the quality of their supervision but I also count them among my friends. Together they displayed a great deal of enthusiasm for my research efforts and admirable patience during my meandering.

Dr. Wade provided an endless stream of new and challenging ideas for me to consider in my research efforts. His creative approach to all problems pushed me to find my own creative solutions to problems that I encountered during my research. ability to get directly to the root of an issue prevented me, on numerous occasions, from wandering off into areas peripheral to my research topic. Dr. Wade's influence, and guidance may be observed throughout my dissertation. I will always remember with gratitude his personal counsel during anxious times.

I consider myself fortunate to have been able to work with Dr. Johns, both as a research assistant and a Ph.d candidate. His expertise in research methodology and his ability to communicate his knowledge in a manner which is both

educating and motivating was an important influence in my research. His pursuit of academic excellence in all his endeavours proved to be a source of constant motivation, propelling me to greater achievement in my work. Each time I left Dr. John's office, no matter how depressed or frustrated I was upon entering, I always left with renewed enthusiasm.

Given the qualitative nature of my research, determining which statistical tests would best describe my data proved to be a difficult task. Dr. Tomberlin's expertise in statistics proved to be invaluable in devising measures and in choosing statistical tests. I will always remember fondly his advice on more than one occasion when I questioned whether a certain approach or revision was necessary. He said "why don't you just do it." Invariably he proved to be right.

I feel enriched to have been able to work with Dr. Marley. I learned to listen carefully for his subtle and always polite comments on my research. His comments were instrumental in indicating to me whether my research was on track and in presenting me with new and exciting ways to look at my work. Dr. Marley was a constant source of support and motivation.

In addition to my committee members, I would like to thank Dr. Vimla Patel and Dr. Brent Gallupe. Dr. Patel provided me with encouragement, advice, and critical commentary on my work. She always made time to meet with me given the most demanding schedule I have ever witnessed. Her

interest in my work and in my career is much appreciated. Dr. Gallupe, in his role as external reviewer, provided a detailed and helpful critical commentary on my work.

The last set of acknowledgements are directed to my family. I believe that it is impossible to appreciate how much the support of family during the Ph.d. program can mean unless you have lived through the thesis experience. Above all, I owe any success that I have had in my academic accomplishments to my wife, Avivit. Her steadfast support, incredible patience (and well timed impatience) and general ability to put up with me during the Ph.D. program have given me the strength to achieve my goals. I don't believe that I will ever be able to adequately convey to Avivit, the gratitude, admiration and love that I have for her, but I intend to spend the rest of our lives trying. To my two and a half year old son Kyle, people are very amused when you pick up one of your toys and declare that you are "going - to work on your thesis on the computer." That statement tells it all. You have unknowingly served as an inspiration to us all. Lastly, I would like to thank my parents who have always considered the education of their children of highest importance and have sacrificed to provide me with the opportunity to pursue my education.

I dedicate this thesis to the memory of my mother, Audrey Glass, for her love and support of my pursuits throughout her life. I regret only that I did not adequately express my feelings to her when I had the opportunity.

# TABLE OF CONTENTS

CHAPTER III

## LIST OF TABLES

# LIST OF FIGURES

## INTRODUCTION

A key element in decision making is the ability to interpret and integrate information items (Payne, 1976). A central tenet of the Human Information Processing (HIP) paradigm and one adopted by the Management Information Systems (MIS) field is the recognition that humans are constrained in decision making because of their limited information processing abilities. As a consequence, researchers in decision making and MIS have placed a major emphasis on studying issues surrounding the design and implementation of computerized decision support systems (DSS)[1] that make use of the computer's large information processing capacity and speed.

Within the last decade, there has been a significant increase in the number of applications of DSS in organizations and in the level of related research activity. Most of the current research investigating DSS has been aimed at determining the optimal parameters and development process for implementing these systems (Sharda et. al., 1988). To determine optimal parameters for a DSS, researchers have focussed on identifying information items that are important in the decision making process.

Munro and Davis (1977) highlight the basic premise

[1] DSS is used in the very broadest sense to include all computer aided support for decision making and as such includes aids such executive information systems and knowledge based systems.

underlying this research:

> In fact, the value of the entire system, apart from its performance of clerical functions, is normally embodied in the user-perceived value of the information provided by the system. If the information provided by the MIS is perceived to be of low quality, then the MIS is held in low esteem.

For the most part, the research concerning information item requirements has taken the approach of considering dependent measures, such as decision accuracy, speed and confidence that represent the outcome of a particular decision making process and has investigated the influence on these variables of independent measures of information such as the value or structure of information presented (Todd and Benbasat, 1987). The objective of this line of research is to determine what information is most appropriate to present to the decision maker during a decision making session and how to best present it.

Todd and Benbasat (1987) point out that a common factor in this research is the treatment of the decision making process as a "black box" that is left unexplored. They claim that in order to make some a priori statements about the direction of the cause and effect relationships, researchers rely on the characteristics of the task, of the decision makers, and of the decision environment.

To date, this line of research has provided us with inconclusive results with respect to the effectiveness of DSS for aiding decision makers (Bronner and De Hoog, 1984; Huber, 1983; Sharda et. al., 1988). Certain studies have found that a DSS helps decision makers, other studies have indicated no effect at all, while others have actually suggested a deterioration in decision making outcomes (Kotteman and Remus, 1987).

Several contributing factors may have influenced the inconsistent results in the DSS research including: research design (one time studies, student subjects, unrealistic problems), differing tasks among research projects, quality of DSS development and implementation and supporting the wrong decision task with respect to influencing the outcome.

Not withstanding the other problems identified, a growing number of researchers have begun to suggest that a major factor in the failure of this body of literature to converge is the lack of consideration of the decision making process in the design of the research studies and in the development of the DSS.

The need to consider decision making process in the design of effective decision aids has been suggested for some time. Simon (1965) states:

> These aids to human thinking ... were devised without understanding the process they aided-the thought process itself. The prospect before us now is that we shall understand the process. We shall be able to diagnose with great accuracy the difficulties of a...decision maker...and

> we shall be able to help him modify his problem solving strategy.

Today, more than twenty-five years later, the need to study the decision process is still stressed as being basic to attaining a better understanding of decision making (Einhorn and Hogarth, 1981; Kleinmuntz, 1985; Little, 1986; Payne, 1976; Svenson, 1979) and to aid in the construction of DSS (Pitz and Sachs, 1984; Ramaprasad, 1987, Remus and Kotteman, 1987, Todd and Benbasat, 1987).

From the perspective of decision making process, the limitations of the research may be considered from two viewpoints. First, much of this research has been aimed at information requirement, that is providing information cues that are important for the decision process. The literature that considers the value of information, for example, attempts to identify information cues that have been subjectively assessed as being of high quality on several dimensions and then selects these for input into a system. The linear model research, on the other hand, attempts to identify information cues that are weighted significantly in multivariate regressions and to select these cues for input to the system.

In general, the information cues in these studies have been selected without an understanding of the decision making process and as such the systems do not give the decision maker a meaningful context for acquiring, storing and handling information. Bronner and De Hoog (1984) state:

> ... many so called decision support systems fail because too much attention is paid to providing information, instead of trying to elicit the underlying decision process in which the information must fit in order to be meaningful for the decision makers.

Second, this methodological approach tells us nothing about how the decision maker evaluates information. The acquisition and evaluation of information are closely related. A neglect of either may be detrimental to our understanding of the dynamics of decision making (Bronner and De Hoog, 1984; Einhorn and Hogarth, 1981).

An understanding of the decision making process could provide guidelines to: (i) determine what information is important to include in a DSS, (ii) design the interface or method of presentation of information to the decision maker, (iii) make the system cognitively compatible with the decision maker, (iv) emulate the decision strategies of expert decision makers and (v) train decision makers.

One discipline that has begun to investigate decision making processes in depth is cognitive science. In particular, cognitive scientists have studied differences in decision making processes between experts and novices.

The cognitive science paradigm provides an interesting theoretical foundation for studying decision making processes and the research methodology used by cognitive scientists may be adopted to help open up "the black box" and to determine how an understanding of the decision making process can be applied in the design of a DSS.

Johnson (1984) describes the synergistic effect of applying a cognitive science perspective to the design of DSS:

> The cognitive scientist studies the behavior of experts and non-expert individuals and his findings are a contribution to our understanding of the expert mind. The information scientist studies the behaviors of experts and non experts on the task to be supported and his findings provide both a source of algorithms for achieving high levels of proficiency, as well as a clear statement of what the behavior to be supported looks like at lower levels of task performance. Because the cognitive scientist ultimately seeks a performance model for the behavior in question, his findings add to an understanding of the phenomena ... and because the performance model for a task is written in terms of the human information processing system, the information scientist is able to design a support system that reflects the processing capabilities and needs of the user.

This research project applies the cognitive science paradigm to explore expert and novice decision making for decision tasks involving software problem diagnosis. The results of the research project help to unlock the "black box" to give us a better understanding of human decision making processes and to provide guidelines for the development of DSS. More specifically, the following research questions are addressed in this project:

1. Can we determine what information is important to the decision maker? How is this information organized in the decision maker's memory for processing?

2. Is there consensus among decision makers at a global level for a set of information items that is considered important in the decision making process, or does the information vary by individual? Does it differ between decision makers of varying skill levels?

3. How do decision makers use this information? That is, what strategies do they use to process the information? Does their strategy change for different kinds of problems in the domain?

4. Are there differences in the decision making process for tasks of varying complexity?

A key consideration in the study of decision making processes is the research methodology adopted to elicit and interpret the decision processes of the subjects. Todd and Benbasat (1987) have made a strong case for the use of process tracing, in particular non probing verbal protocol analysis, to study the decision making processes of decision makers. Verbal protocol analysis has been a popular research tool in the cognitive sciences and has had some application in DSS research (Biggs et. al., 1985; Elam and Mead, 1986; Kleinmuntz, 1985; Payne, 1976).

Verbal protocol analysis has been the recipient of a number of criticisms. While some of the more scathing criticisms (Nisbett and Wilson, 1977) have been shown to be limited to studies that have relied only on variations of protocol analysis such as retrospective analysis (Ericsson and Simon, 1980) other criticisms such as the limitation of protocol analysis to provide only partial information about the cognitive processes of an individual remain well founded (Payne, Braunstein and Carroll, 1978).

Rather than rely on one method for eliciting the decision making process of the decision maker, this research project has adopted a multimethod approach. A multimethod approach could better address the concerns pertaining to the reliability of the elicited decision making processes of decision makers (Einhorn et. al., 1979; Johnson et. al., 1982; Payne et. al., 1978; Todd and Benbasat, 1987; Wallsten, 1980).

Three methods for acquiring the knowledge of decision makers are employed in this research project: (i) concurrent verbal protocol analysis, (ii) aided retroactive verbal protocol analysis and (iii) causal mapping. Each method contributes to the richness of the data collected and allows for a comparison of results to support the validity of the conclusions.

In any research project studying decision making, the choice of the decision task is a critical element in the design of the project and in its influence on research outcomes. In this project, a software problem diagnosis task has been chosen. A diagnosis task may be characterized as a situation in which a decision maker must seek information cues and evaluate the cues to identify a specific state of affairs that underlies the information set.

Diagnosis is the first step in a problem solving process. It allows the decision maker to recognize a problem and to classify it into a problem type based on the attributes of the problem. The decision maker can then generate solution alternatives that are consistent with the problem attributes.

To date, most DSS research has concentrated on the evaluation and choice aspects of decision making (Kotteman and Remus, 1987). Mintzberg et. al. (1976) have claimed that:

> ... diagnosis is probably the single most important routine (in the overall decision making process), since it determines in large part, however implicitly, the subsequent course of action. Yet researchers have paid almost no attention to diagnosis, preferring instead to focus on the selection routine, which often appears to be just a trimming on the overall decision process.

After reviewing the DSS literature Kotteman and Remus (1987) suggest that the existing DSS appear mainly limited to quantitative modelling with interactive user interfaces. Formal modelling concentrates on evaluation and choice in the decision process and may be incongruent with the decision problem, with the truly appropriate decision process, or with general cognitive processes. They conclude that:

> ... the value of DSS for evaluation and choice in ill-structured situations might be minimal whereas the value of a DSS for diagnosis might be substantive.

Cognitive scientists have studied the diagnosis process in detail, particularly in the domain of medicine (Chi et.al., 1982; Elstein et. al., 1978; Johnson et. al., 1981; Johnson et. al., 1982; Kuipers and Kassirer, 1984; Patel and Frederikson, 1984; Patel and Groen, 1986). The application of the cognitive science research paradigm to a typical software diagnostic task may provide insights for establishing guidelines for the design of a DSS to support the decision maker in performing a diagnostic task.

In summary, the research in DSS has provided us with inconclusive results with respect to the effectiveness of DSS. One factor contributing to the inconclusiveness is the failure to consider decision making processes in the design of research and in the development of DSS. One discipline that has begun to investigate decision making process in depth is cognitive science.

This research study addresses, from the context of a decision process approach, major questions concerning the use of information during decision making that have been dominant in the MIS literature and for which no conclusive results have been generated. The project investigates differences between experts and novices for the performance of software problem diagnosis tasks utilizing a multimethod approach that is grounded in the theory and methods popular in the discipline of cognitive science. The implications of the results of the research project for the design and development of DSS are discussed.

Chapter I discusses the theoretical foundations of cognitive science and reviews the literature concerning expert-novice differences in problem solving. The implications of the cognitive science research for DSS is presented. Chapter II examines the three main research methods used in the project (concurrent verbal protocols, aided retroactive verbal protocols and causal mapping) and describes the research hypotheses. Chapter III presents the methodology used in testing these hypotheses and the research design employed in the project. Chapter IV presents the analysis of the data and the results obtained. Chapter V provides the interpretation of the results and discusses the implications for the design and development of DSS. It concludes with recommendations for future research to better understand how decision makers of varying skill level make decisions and to better understand how to design, develop and implement computer aided decision support systems.

# CHAPTER I

## FOUNDATIONS OF COGNITIVE SCIENCE AND EXPERT-NOVICE RESEARCH

### A - INTRODUCTION

Cognitive science has been acknowledged to have had its birth as an academic discipline in the mid 1950's with the emergence of the information processing paradigm (Simon, 1981). The information processing paradigm focussed on analyzing problem solving behaviour in detail and on providing theoretical interpretations that included specific assumptions about the component cognitive processes involved in problem solving (Greeno, 1978). In addition the information processing approach introduced a rigorous experimental methodology and began a tradition of experimentation that built upon a succession of research projects.

Drawing from its roots in information processing theory, the cognitive science approach seeks to understand reasoning in terms of the mental processes that individuals use in reasoning, the strategies into which these processes combine, the representations upon which the processes and strategies act, and the knowledge base that is mentally represented (Sternberg and Lasaga, 1984).

Within the information processing paradigm there are a number of sub approaches. At one extreme researchers have studied the speed of responses for simple tasks and simple forms of information processing. At the other extreme, researchers have studied complex forms of problem solving, emphasizing accuracy and strategies in information processing. In general, because of its focus on more difficult reasoning tasks, the research on problem solving has proven more interesting from the viewpoint of human reasoning and decision making.

## B - INFORMATION PROCESSING MODELS OF PROBLEM SOLVING

Significant advances in computer technology had been achieved during the second World War. After the war, researchers began to turn their attention to modelling human intelligence on the computer. By the mid 1950's actual computer programs that modelled human intelligence had appeared. Shortly after that models intended to serve as psychological theories of problem solving were developed. Newell, Shaw and Simon's (1960) General Problem Solver and Feigenbaum's (1961) EPAM were among the more prominent early theories.

In 1972, Newell and Simon published the book "Human Problem Solving" in which the first well developed information processing theory of problem solving was proposed. Their book had a profound effect on human reasoning research.

## B.1 - Newell and Simon's Theory

Problem solving is characterized by Newell and Simon as the interaction between a human information processing system, the problem solver's representation of the situation in terms of a problem space and a task environment. These three components establish the framework for the problem solving behaviour. More specifically four propositions capture the theory (Simon, 1978):

1. A few, and only a few, gross characteristics of the human information processing system are invariant over task and problem solver. The information processing system is an adaptive system, capable of molding its behavior, within wide limits, to the requirements of the task and capable of modifying its behavior substantially over time by learning. Therefore, the basic psychological characteristics of the human information processing system set broad bounds on possible behavior but do not determine the behavior in detail.

2. These invariant characteristics of the information processing system are sufficient, however, to determine that it will represent the task environment as a problem space and that the problem solving will take place in a problem space.

3. The structure of the task environment determines the possible structures of the problem space.

4. The structure of the problem space determines the possible programs (strategies) that can be used for problem solving.

The structure of the problem space and the strategies used by the problem solver must be compatible with the information processing system. Newell and Simon identify a few basic characteristics of the information processing system that shape the problem solver's efforts.

First, the system operates almost entirely serially. That is, during a problem solving session the problem solver progresses sequentially through a series of knowledge states, each state adding incrementally to the knowledge that an individual has available at that point in the process. Mental operations are used to transform one state into another. Second, the capacity, storage time, retrieval time and organization of short term and long term memory influence the knowledge that is available to an individual during a problem solving session. For example, the inputs and outputs of the information processing is stored in short term memory with a capacity of only a few familiar symbols or "chunks". Long term memory, on the other hand, has unlimited size and is organized associatively using symbols and structures of symbols. However, there are more severe time constraints on the storage and retrieval of chunks of information.

Newell and Simon suggest that the structure of the task constrains the problem solving process by providing information about the nature of the problem and what is required to solve the problem. Task structure is defined by four components: (i) information about the initial state of the problem, (ii) information about the goal state, (iii) information about legal operators, ie., the kinds of procedures that one can partake in to move from an initial state to a goal state and (iv) information about restrictions that may constrain the use of any particular operator.

An individual's representation of the task in memory is termed the problem space. The problem space influences the strategy that the individual will adopt to solve the problem. For example, given a board position in a simple game such as tic-tac-toe, a problem solver may be able to consider the results of applying each potential legal move to the position, backtracking to previous positions when reaching a dead end, until a goal state (win or forced draw) is found. Because of the limited size of the problem all the positions could be stored in memory and previous positions recalled. Thus a strategy of exhaustive search (try all possible moves) may be undertaken.

For large problems, with complex operators, backtracking to previous positions when a dead end is reached in the problem solving process may be impossible because of the excessive amount of memory that would be required to store the positions. For these types of problems a strategy would be required that limits the problem space to a more manageable size.

Greeno (1978) identifies three classes of tasks based on the psychological skills and knowledge needed to solve problems in the classes. These three classes are not exhaustive, and may appear in combinations for various tasks. Greeno's classes are:

1. Problems of inducing structure. Some elements are given and the main task is to identify the pattern of relations present among the elements.

2. Problems of transformation. There is an initial situation and a goal and a set of operations that produce changes in situations.

3. Problems of constructing arrangements. Some elements of the problem are presented and a problem solver is required to arrange them in a way that satisfies some criterion.

The relative ease of solving a problem depends on how well the problem solver captures the critical features of the problem task in his or her problem space. Each individual's problem space is representative of the total store of knowledge that he or she has to deal with a given problem. Therefore individuals faced with the same problem may take varying problem solving approaches. Some of the factors that can influence an individual's problem space are (Reed, 1982):

1. The task instructions that give a description of the problem and may contain helpful information.

2. Previous experience with the same task or a nearly identical one.

3. Previous experience with analogous tasks.

4. Plans stored in long term memory that generalize over a range of tasks.

5. Information accumulated while solving a problem.

The problem solver's choice of operators to use in transforming an initial problem state to a goal state is governed by his or her strategy. At the simplest level, a strategy may consist of an exhaustive search. The problem solver would apply all applicable operators to a state, choose one of the successor states and apply all applicable operators to that state and so on, backtracking when necessary, until a goal state is reached (these searches may take the general form of depth first or breadth first).

The difficulty with employing exhaustive searches is that the problem space quickly grows to exceed the capacity of the human information processing system. Consequently, it is suggested that problem solvers employ heuristic strategies that reduce the search space to solve problems. Heuristics are rules of thumb, or other simplifications that may be used to solve problems but do not guarantee a solution to the problem. Newell and Simon stress three such strategies, (i) means-end analysis, (ii) subgoaling and (iii) generate and test.

Means-end analysis refers to the process of comparing the current state of a problem to the goal state and discovering the differences between the two. An operator is chosen to apply to the problem that reduces the difference. If more than one operator exists then the one that reduces the difference the most is chosen first. For example, if a goal is set to minimize transportation costs for a large shipment of product and there are several alternative modes of transportation available then a means-end strategy might be to ship the maximum limit by the first cheapest alternative, then send as much as possible by the second cheapest method and so on until the shipment has been completed.

The main problem with this approach is that it may often be necessary to apply an operator (select a mode of transportation) that initially moves away from the solution but may later lead to a superior result. In the above example it may be better to truck the goods to a port (initially more expensive than rail transport) and then transfer the shipment to a boat for the balance of the distance.

Subgoaling is a strategy that can be used with means end analysis. It refers to the selection of intermediate states as the goal state and applying operators to transform the current state into the intermediate goal state. A new goal state is then chosen and the process repeated until the final goal state is reached. Selecting intermediate goals dramatically reduces the search space required to achieve a goal, making the problem more manageable. For example, in the transportation problem above, a subgoal may be chosen to find the cheapest method of transportation to a port, then the cheapest method of boat shipment to the destination city and finally the cheapest method to transport the goods from the dock to the customer.

A major limitation of this approach derives from the fact that the intermediate goal states that are chosen may not be appropriate for achieving the goal. In the transportation problem it may be cheaper to direct ship to the customer by truck rather than transfer the shipment among carriers three times.

Generate and test refers to the generation of a set of possible problem solutions and then testing them. It may be an appropriate strategy when the set of possible solutions is small or there is some criteria for choosing a subset of solutions. Typically, the selection and testing of one possible solution, even if unsuccessful, provides information for the testing of a subsequent solution. For example, in the transportation problem, a potential shipment alternative may be defined and evaluated to test whether it achieves the desired cost objective.

A major limitation of generate and test strategies is that for large problems it may not be practical to continue testing alternatives until a suitable one is found. A second limitation is that even if one achieves an objective, it may be still possible to improve upon it. In the transportation problem an alternative may be chosen that would cost nine thousand dollars therefore meeting the objective of not exceeding ten thousand dollars. It may be possible to achieve an even lower cost by exploring more alternatives. The choice of a satisfactory alternative rather than a proven optimal alternative has been referred to by Simon (1977) as satisficing.

In summary, Newell and Simon's theory of human problem solving is an attempt to describe a complex information processing model that would simulate a wide range of human mental activity (Posner and McLeod, 1982). In spite of proposing a general theory of problem solving, Newell and Simon, and many of the researchers who followed their approach in the early 1970's based their studies on naive subjects who performed well- structured tasks that required little prior knowledge of the task environment for their solution.

Well - structured tasks are tasks for which the initial task state, the goal state, the set of operators that can be used to transform the initial state into the goal state and the constraints on the operators are all known prior to attempting to solve the problem (Simon, 1978). Ill-structured tasks are tasks for which at least one or more of the above features of the task are not clearly defined or are unknown. Managerial decision tasks as well as many other real world problems fall into the ill - structured category.

Simon (1978) acknowledges this problem:

> The great bulk of research that has been done on problem solving has made use of task environments whose structure is well defined. It is reasonable to ask to what extent the mechanisms that have been discovered to govern problem solving in well-structured domains are also applicable and used in domains that are more loosely structured.

Simon (1978) answers this question by suggesting that the processes used to solve ill structured problems are the same as those used to solve structured problems. However, when studying more complicated problem tasks, the general strategies described by Newell and Simon did not, by themselves, sufficiently describe problem solving performance (Chi and Glaser, 1985). Therefore, because of the limited application of general problem solving strategies, research in the Newell-Simon approach to find general strategies across domains has been abandoned for the most part.

Nevertheless, Newell and Simon's theory has had a profound effect on research in human reasoning and on research in expert reasoning in particular. First, their theory significantly changed problem solving research by changing the focus of research away from the conditions under which solutions can be reached toward the component cognitive processes involved in problem solving (Chi and Glaser, 1985).

The focus on cognitive processes increased our understanding of how more structured problems are solved. It also contributed to theory and applications in the field of artificial intelligence (production rules for example).

Second, Newell and Simon's theory was an instrumental force in establishing the information system paradigm for research in human reasoning. The paradigm provided a framework in which to study the cognitive processes involved in more complex problem solving tasks. Researchers were able to compare the results of their studies in complex domains to the predicted results of Newell and Simon's theory and research. These studies supported Newell and Simon's notion of task importance and the human information processing system but could not explain the actual strategies used within the general problem solving theory. This led to a change of focus for understanding reasoning; away from modelling performance processes to concentrating on the nature of the problem solver's knowledge and his or her organization of that knowledge.

In the search for a more descriptive theory of human problem solving, cognitive scientists focussed their attention on in depth studies of the decision making process for narrow problem domains. The objective of these studies was to identify the type of information or knowledge that decision makers had acquired with respect to the domain and the organization of knowledge in the decision makers' long term memory.

A popular research methodology adopted by cognitive scientists was to investigate expert-novice differences in solving problems in a specific domain and the changes that occur as an individual progresses from novice to expert. The objective of studying experts and novices was to compare the knowledge acquired by each type of decision maker and their organization of the knowledge in memory in order to shed light on the decision making processes for proficient decision makers in the domain.

In the following sections the major results in expert-novice research will be discussed. The influence of Newell and Simon may be perceived throughout the discussion.

## C - EXPERT-NOVICE RESEARCH

The literature on expert-novice differences has provided a rich information base on decision making processes and on the nature of expertise in decision making. This literature suggests important considerations for the design of DSS, particularly in the case of information cue selection (search) and information cue combination (interpretation).

In this section, the cognitive science research in expert-novice differences will be reviewed and schemata theory will be discussed. Schemata theory provides the theoretical foundations upon which the cognitive science paradigm is based.

Do experts really solve problems differently than novices do? Johnson et.al (1981) claim:

> ..the prevailing view seems to be that experts and novices are similar in the type and frequency of concepts employed in generating a problem solution - what we shall term "form reasoning", but differ in the appropriate use of these concepts as well as the success of the eventual solution - what we shall term "substance" or "content of reasoning".

In this context experts are not seen to perform better than other people by virtue of their superior use of general problem solving methods (form reasoning), but rather, as a result of the vast amount of knowledge that they have accumulated with experience and the manner in which they organize their knowledge in memory (content reasoning).

In the next two sections research relating to form reasoning and content reasoning will be presented.

**C.1 Form Reasoning Research**

The research on form reasoning reviewed in this section supports the hypothesis that the type of problem solving methods employed by experts and novices are primarily the same and that the use of problem solving methods as a criterion to differentiate between expert-novice problem solving performance would not lead to fruitful results.

In an experiment by Chase and Simon (1973a), subjects of varying chess skill were asked to reproduce positions on a chess board. When meaningful positions were given such as familiar castled - king positions or a pawn chain, master chess players could recall approximately twice as much detail of the board positions than class A players could and four times more detail than beginners could. However, when presented with non familiar, random positions there was no difference in the performance of the subjects.

Chase and Simon concluded that experts have organized and stored in memory large numbers of patterns or chunks of chess information that they use to evaluate board positions. A chunk is basically any organization of information that has previously become familiar (Simon, 1979). A familiar pattern for an expert may contain the positions of several pieces whereas novices who are less familiar with the game and may have a chunk consisting of the position of only one piece. Given that short term memory has a capacity of approximately seven chunks (Miller, 1956), the chess masters could maintain much more information about the board when the positions were familiar than the novice. For randomly assigned, meaningless positions, the chunks of the experts and novices contained similar information.

McKeithen et.al. (1981) duplicated Chase and Simon's results in the domain of computer programming. Expert programmers performed much better than novices in recalling meaningful programs that were displayed briefly on a computer screen but performed equivalently in recalling scrambled programs with randomly assigned, meaningless code.

Elstein, Shulman and Spafka (1978) found that both experts and novices in a medical domain used the same strategy of hypothesis generation to solve a clinical diagnosis problem. They suggested that the decision makers simultaneously evaluated between two and four hypotheses during a problem solving session. If the decision maker rejected a hypothesis then the hypothesis would be replaced by another one, thus keeping the number of hypotheses under consideration relatively constant. The difference between experts and novices was suggested to be in the experts' greater underlying knowledge of the domain which allowed them to more accurately evaluate hypotheses and to consequently make better diagnoses.

Johnson et.al. (1981) found similar results in studying medical diagnosis problems. Students with very little knowledge and virtually no experience were able to generate a sequence of hypotheses that closely resembled the experts. The subjects did differ, however, in their choice of reasoning approaches employed to evaluate the hypotheses. The experts used more detailed evaluations and were more accurate in their diagnoses.

The results of these experiments suggest that the differences between expert and novice problem solving in a domain is not attributable to the experts' superior problem solving skills, but rather to the greater quantity and quality (organization) of the experts' knowledge in specific domains. Glaser (1984) states:

> Our interpretation is that the problem solving difficulty of novices can be attributed largely to the inadequacies of their knowledge bases and not to the limitations in their problem solving capabilities such as the inability to use problem solving heuristics.

In this case the understanding of expert reasoning lies in the study of content reasoning. Glaser (1984) further claims:

> Current studies of high levels of competence support the recommendation that a significant focus for understanding expert thinking and problem solving and its development is investigation of the characteristics and influence of organized knowledge structures that are acquired over long periods of time.

**C.2 Content Reasoning Research**

The research within the content reasoning perspective has focussed on the role of knowledge in expert reasoning. Two major questions have guided this research: (i) what kind of knowledge do experts have? and (ii) how is the knowledge organized in memory?

A major influence on the research methodology adopted in cognitive science has been the underlying assumptions held by researchers with respect to the nature of human memory. Generally speaking, the structural models that have been attributed to memory in the information processing approach are of a purely hypothetical nature. Estes (1978) comments:

> Structural concepts are introduced primarily in the course of attempts to provide characterizations of the state of information preserved by an individual at particular cross sections in time during the carrying out of a cognitive task. The goal is a characterization that is theoretically significant in the sense that it helps in the prediction of the further course of human information processing. Thus, the structural concepts must entail the analysis of stored information into constituents and the expression of relations between these constituents in some systematic way.

Nonetheless, these conjectural structures have proven powerful by providing a framework in which to intelligently discuss human reasoning, to predict performance and from a practical viewpoint, to design computer programs that solve complex problems.

The dominant structural model of human memory within cogniti'-' science is schemata theory. Schemata theory has been presented in a modified form as script theory (Abelson, 1976; Schank, 1980) and frame theory (Minsky, 1975; Winston, 1978).

### C.2.a Schemata theory

The foundation for schemata theory was proposed as early as the 1930's. Bartlett (1932) described the underlying notion of a schema:

> There is abundant psychological evidence that people use a large, well coordinated body of knowledge from previous experiences to interpret new situations in their everyday environment.

Schemata theory describes how an individual's body of knowledge is organized in memory. A Schema may be described as a recurring pattern or prototypical combination of data that is familiar and can be recognized. The schemata represent knowledge that we experience, interrelationships between objects, situations, events, and sequences of events that normally occur (Glaser, 1984). Rummelhart and Norman (1983) identify key features of schemata:

1. They contain variables; a schema of a concept has fixed parts, which are assumed always to be true for instances of the concept, and variable parts. Variables have two important properties: They have default values, values assumed to be true as long as the incoming

information does not specify it otherwise, and they are augmented with knowledge of the plausible range over which possible fillers of the variables might vary.

2. Schemata can embed one with another; a schema can consist of a configuration of subschemata.

3. Schemata represent knowledge on all levels of abstraction; there are schemata for natural basic concepts like apples and trees, as well as for abstract concepts like love and cognitive psychology.

4. Schemata are active processes, actively trying to evaluate incoming information and ascertaining the degree to which they are relevant to structuring the input.

It is hypothesized that an individual, faced with a situation (external data) compares the situation to the store of schemata in memory in order to determine if the situation is an instance of a known schema. If the situation is unfamiliar, that is there is no match, then the individual will attempt to construct a new schema by looking for commonalities in the external data and various different schemata in memory. In this manner the interpretation of an unfamiliar situation is attempted by reference to the familiar.

For example, suppose that an individual walks into a Japanese restaurant for the first time and immediately observes that people are seated on the floor without their shoes on. The observed behaviour may be totally unfamiliar and unexpected to the individual and may not match any instance of a known schema in his or her memory. The individual may then construct a new schema of a Japanese restaurant by noting the commonalities between the Japanese restaurant and other restaurants contained within existing schemata (they all have waiters, you are served, there are menus, you place an order, etc.) and between the people's behaviour and schemata relating to Japanese customs. The result of the comparisons will be a new schema that depicts a Japanese restaurant.

A schema is much like an internal theory of a particular state of the world. As such, individuals are constantly testing their schemata for validity. Glaser comments:

> A schema can be thought of as a theory or internal model that is used and tested as individuals instantiate the situations they face. As is the case for a scientific theory, a schema is compared with observations and if it fails to account for certain aspects of these observations, it can be either accepted temporarily, rejected, modified, or replaced. Like a theory, a schema is a source of prediction and enables individuals to make assumptions about events that generally occur in a particular situation, so that the knowledge they infer goes beyond the observations that are available in any one instance. Such prototypical structures play a central role in thinking and understanding, and in the context of specific networks of knowledge.

It is suggested that schemata are organized in hierarchies (Johnson et.al., 1981). The more general patterns are at the top and the more specific ones at the bottom. For example, in a chess game, activating specific schemata of chess positions (similar to the chunks described above) may in turn activate a more general higher level schema which provides an optional move (Chi and Glaser, 1985).

Within the schemata perspective an expert is considered to exhibit superior problem solving performance in a domain by being able to quickly match a given situation to existing schemata in memory. The schemata that are matched can activate other schemata in the hierarchy thus making all the relevant information for the problem available to the decision maker and providing the expert with solution alternatives. The schemata for novices in the domain are either nonexistent or poorly formed.

### C.2.b Schemata research

In Chase and Simon's (1973a) study discussed above, a conclusion drawn was that the superior performance of chess experts for meaningful board positions resulted from their having accumulated a large store of chess patterns or chunks in memory that allowed them to quickly recognize board positions. The organization of the chunks described by Chase and Simon are analogous to schemata.

In related experiments, Chase and Simon (1973b) found that experts were able to memorize the moves of both players in a chess game more quickly and more accurately than novices could. They concluded that experts have stereotyped patterns of moves organized in their memory.

Taken together with the first experiment, a possible interpretation of the two sets of experiments is that experts have schemata of board configurations in organized memory and that associated with these configurations are move sequences. Each move sequence has an expected performance outcome based on the experience of the expert. Consequently, the expert can quickly identify a board position and choose superior moves. A novice on the other hand, has not acquired the large store of chess schemata that the expert has acquired, nor the expected outcomes of move sequences.

The findings of Chase and Simon (1973a, 1973b) with respect to experts having superior memory capacity as exhibited by the formation of large pattern of chunks, has been supported by research in several other fields such as Go players, Gomoku players, bridge players, musicians, baseball fans, computer programmers and electronic technicians (Chi, Glaser and Rees, 1982).

The above interpretation touches on a key facet of the schemata approach, that is, the association between declarative and procedural knowledge in memory. Declarative information refers to the organization of factual knowledge and relationships that are relevant to the domain (evaluating the chess board position). Procedural knowledge is attached to specific chunks of declarative knowledge organized in memory and serves to translate the declarative knowledge into actions that achieve expected outcomes (selecting a sequence of moves that are perceived to lead to a favourable outcome).

Chi et. al. (1981) investigated the declarative and procedural association for problem solvers of various skill levels in the domain of physics. In a series of experiments, they found that both experts and novices categorized physics problems into potential solution categories in meaningful but different ways. Experts categorized the problems based on underlying physics principles such as the conservation of momentum, the conservation of energy and Newton's force law whereas novices categorized the problems based on surface knowledge such as literal objects (keywords) and concepts stated in the problem itself. They also found that the problem solvers adopted problem solving strategies that were consistent with their representation of the problem. They conclude:

> ...the early phase of problem solving (the qualitative analysis) involves the activation and confirmation of an appropriate principle-oriented knowledge structure, schema. The initial activation of this schema can occur as a data driven response to some fragmentary cue in the problem. Once activated, the schema itself specifies further (schema driven) tests for its appropriateness (Bobrow and Norman, 1975). When the schema is confirmed, that is, the expert

> has decided that a particular principle is appropriate, the knowledge contained in the schema provides the general form of the solution...

The role of declarative knowledge in schemata is thus to generate problem configurations, test their applicability and to conclude with an appropriate problem statement. Procedural knowledge in the schema provides potential solution methods.

Another result of their study was that advanced novices categorized problems by basic principles relying somewhat less on surface knowledge than novices but are still constrained by surface knowledge. The suggestion is that as one progresses from expert to novice, one acquires more refined knowledge about a domain allowing for a deeper, more basic understanding and reasoning about problems in the domain.

Given these results, an interesting question arises. Are the representations of problems in a domain similar for individuals at a similar level of expertise in the domain?

Johnson et.al. (1982) addressed this question in a study of medical diagnosis problems. The subjects were selected from three categories; expert (board certified), trainee (several years experience but still not certified), and student (medical students). Examining individual subject interpretations of specific cues and cue combinations in reaching an overall diagnostic judgment, they noted that for a relatively regular set of data the experts reached a common diagnostic judgment that was linked to the interpretation of a relatively small number of critical data cues. However, when faced with patient data that had discrepant or incongruent combinations of cues, the experts adopted different interpretations of the critical cues as well as divergent judgments. They conclude:

> These findings suggest that at a global level of reasoning where judgment outcomes and critical data cues are an issue, experts tend to agree., while at a more detailed level of reasoning involving the interpretation of individual data cues and specific reasoning steps, they diverge.

Interestingly, the problem solving behaviour of novices was found to be more congruent. Perhaps their lack of detailed knowledge of the field and the influence of recent medical studies served to limit the means that they had at their disposal for dealing with the problems. The novices and trainees, however, consistently performed below the standard of the experts.

The results of Johnson et. al. (1982) are consistent with prior research on experts. Einhorn (1974) studied the cues that experts use in a medical diagnosis task. He hypothesized that experts would agree on which cues were important for making a diagnostic judgment and that they would also agree in the relative weights of the cues for a given problem. There was support for the hypothesis that experts agreed on which cues were most relevant for the task, but there was no support for the similarity of weights. In fact, "they weight the cues quite differently in coming to their global judgment."

If experts weight cues differently and reach different diagnostic judgments, then it is suggested that at some level the schemata that they have organized in memory may differ. What leads to this difference? A study by Kuipers and Kassirer (1984) may be interpreted to shed some light on this question. Kuipers and Kassirer investigated the reasoning of medical specialists for a medical diagnosis problem. They hypothesized that the physician has a cognitive causal model of the patient, incorporating the expert's knowledge of anatomy and physiology, the body's pathological behaviour in a diseased state and the particular idiosyncracies that characterize a particular patient.

After many years of experience an expert physician has compiled his or her knowledge so that a long chain of inferences between the patient's physical state and a final diagnosis is likely to be reduced to a single association. The associations reflect the schemata organized in memory.

An expert may select only a few factors from a set of all potentially relevant factors for a problem. For example, if a patient is high on factors one, two and three, then the other factors for consideration would be ignored. This smaller set of factors that the expert relies on are referred to by Kuipers and Kassirer (1984) as "smaller models". The smaller models allow the expert to quickly reach a diagnoses without having to perform an extended search through memory.

..o make up for the detail omitted in the smaller model, the expert must have many different small models to deal with the problem. The particular model chosen by the physician to deal with a problem task is dependent on the physician's comparison of the problem to schemata organized in memory.

Each small model may vary in the combination of factors upon which the causal model is based or in the weighting of factors. The content of the models depends on the training and experiences of the physician. Thus, two experts in the same field may view a domain similarly from a global perspective (what factors are important in the general domain) but may vary in their use or weighting of factors (cues) for any specific task within the domain.

The notion of compiled knowledge is a recurring theme in cognitive science research. The compiled knowledge may be of two forms: (i) a chain of associations and (ii) sequence of production rules.

Compiled knowledge that consists of a chain of associations reflect the hierarchical structure of schemata. An expert faced with a problem in a domain would extract information (cues) from the situation that would activate a schema or set of schemata in memory. Other schemata would be automatically activated in a predetermined sequence that the expert has developed with experience. A novice in the domain may not have developed a schemata structure that could adequately deal with the problem. Therefore, the novice may be unable to solve the problem or may take considerably longer because of the need to construct new schemata in memory. Even in the case where the novice has a set of associations to deal with the problem, the schemata may not contain the same amount of detail nor the quality of knowledge that an expert's associations contain and consequently may result in a lesser quality of output.

Compiled knowledge from the production rule approach suggests that an individual will form a system of rules to deal with situations. The rules would be of the general form: if CONDITION is satisfied then perform ACTION. An expert would develop rule systems that are maintained in memory that enable the expert to deal effectively with a wider range of problems in a domain than a novice.

Schemata and rules may be combined in memory. Attached to the schemata may be a sequence of rules that are activated (fired) when a specific schema is instantiated. The schemata may be interpreted then to provide an understanding of a problem (declarative knowledge) and the attached rules to provide a sequence of cognitive actions (procedural knowledge) that would suggest a resolution for the problem.

**C.2.c Summary of schemata theory**

Schemata theory provides a coherent explanation of how individuals organize and store knowledge in memory into familiar combinations. Schemata serve as a basis with which an individual can compare external data and recognize the data as instances of familiar situations or interpret the data to create new schemata of unfamiliar situations. A schema contains both declarative knowledge that is used to characterize an appropriate representation of a situation and procedural knowledge that provides solutions or actions to deal with the situation.

Experts are considered to exhibit superior performance in a domain because of the large store of domain specific schemata that they have acquired with training and experience and the high quality of knowledge that is contained in their schemata. Experts appear to have similar global schemata that identify key factors or cues in a domain but tend to differ in their schemata at the level of specific domain problem tasks. The differences stem from the unique personal experiential history of each expert.

Anderson (1983) has criticized schemata theory on two grounds. First, he claims that the theory blurs the procedural declarative distinction and leaves unexplained all the contrasts between declarative and procedural knowledge. Declarative knowledge is flexible and can be accessed in many different ways. Schemata, Anderson claims, are inherently flexible and are more typical of declarative knowledge. Procedural knowledge tends to be rigid but efficient. Once evoked, procedural knowledge follows a direction to conclusion. The extreme difference between the two types of knowledge merits a separation in a theoretical model.

Anderson's second criticism centers on the fact that units of knowledge in schemata tend to be too large thus forcing the system into modes of behaviour that are too limiting. A theory of memory should be organized around smaller units allowing for richer and more dynamic combinations.

Anderson's criticism of the declarative - procedural blurring is well taken. Schemata theory implies that because of the linking of declarative and procedural knowledge in memory, the relationship between the two types of knowledge is symmetric. Information from either type of knowledge can influence the choice of schemata to represent a situation. For example, given a situation that partially matches the declarative knowledge in a schema and procedures or actions that are being performed in the situation that do match the procedural knowledge of the same schema, schemata theory would suggest that the procedural knowledge will be used to fill in the missing declarative knowledge and thus complete the schema.

Logically, however, the relationship between declarative and procedural knowledge would appear asymmetric, declarative knowledge being necessary prior to the choice of procedures. For example, from the fact that the light is green, one wants to infer that one can walk. One does not want to infer that the light is green from the fact that one is walking, that is, if one values their life (Anderson, 1983). While a schemata theorist would not suggest that an individual would make the fallacious assumption that the light is green, it is not quite clear how the individual would make the discrimination.

Anderson's second criticism however is basically one of interpretation. While schemata theorists posit large schemata hierarchies that are used to make sense of the external world, there have been several attempts to break down schemata into smaller constituent parts that provide for dynamic and flexible combinations of information (for example script theory).

## D. IMPLICATIONS OF SCHEMA THEORY AND EXPERT-NOVICE RESEARCH

The importance of developing DSS for supporting diagnosis in decision making has been suggested previously. Schemata theory would lend support to the importance of diagnosis in decision making. According to schemata theory, before an action could be taken to resolve a problem, stimuli from the environment would have to be associated with a schema structure in memory so that the problem could be recognized and pertinent information made available to the decision maker. Attached to the schema would be procedural knowledge that would allow the decision maker to commence a pattern of action to resolve the problem.

The initial instantiation of schema structures corresponds to the traditional notion of diagnosis and must precede evaluation of task solution alternatives and choice. In fact, the instantiation of the schema structures dictate the decision resolution activities that may be considered by the decision maker. Therefore, decision aids to support the decision maker in diagnosing or understanding a problem in the initial stages of the decision making process may have a greater payback than aiding the selection and choice phases.

A second implication derives from the importance of cue selection and cue combination in decision making. For routine decisions where the decision maker has had substantial experience, the role of a DSS would primarily be to provide the decision maker with the information cues that are critical to the decision task in an efficient and effective manner. This may involve providing important information cues that may not be consciously solicited by the decision maker (intelligent checklist) or that are not readily available to the decision maker. It may involve helping the decision maker to distinguish cues from a background of noise or to provide the cues in a predetermined effective order.

Once provided with the critical information cues, the decision maker will be able to instantiate the schemata in memory that are required to effectively deal with the problem at hand.

The fact that a decision maker has had substantial experience with a particular decision task is not a guarantee that he or she will make accurate diagnoses. Where it has been identified that this may in fact be the case, a DSS could be designed using normative principles or the domain representation of an expert decision maker to improve upon the decision maker's performance. For example, the DSS could modify the cue selection process of the decision maker in order to influence the instantiation of existing schemata contained in the decision maker's memory or to aid the decision maker form new schemata to deal with the problem.

For non routine decision tasks, where the decision maker has not had substantial experience with the decision task, decision makers will experience problems in both determining which cues are relevant for the decision task and how to combine the cues together to gain an understanding of the problem.

It should be noted that a decision may be non routine for a decision maker because he or she is a novice in the domain and therefore inexperienced. Alternatively, the decision maker may be an expert in the domain and the specific problem task confronting the expert is one for which he or she has not had significant experience. In the latter case it would be expected that the expert would perform at a higher level than the novice by virtue of their vast experience in the domain.

The role of DSS for unfamiliar tasks would be more than to provide the decision maker with the critical information cues in the domain. It would also involve imposing a structure on the information cues presented to the decision maker in order to facilitate the combination of existing schemata in memory or to create new schemata in memory that would allow the decision maker to gain an understanding of the problem and to formulate an accurate diagnosis

In these cases, a critical factor in the success of the DSS would be the interface between the DSS and the decision maker. The interface should be designed to provide the user with access to the system in a manner which is cognitively compatible with the user's representation of the problem domain. This would suggest that an effective interface would be flexible to adapt to a user's varying needs or to be compatible with multiple users.

## E. SUMMARY

Chapter I presented a discussion of the foundations of cognitive science with particular emphasis on the study of human problem solving within the discipline. Schemata theory was introduced and research in expert-novice differences reviewed. The implications of schemata theory and expert-novice research for DSS research and development was discussed.

Section A of Chapter I introduced the field of cognitive science indicating that its roots derive from human information processing theory. Section B discussed human information processing theory focussing on Newell and Simon's general theory of human problem solving. Newell and Simon attempted to identify certain generic strategies that could account for problem solving behaviour. The generic strategies found support in studies of relatively structured, less complex problems such as cryptarithmetic but failed to account for more complex decision tasks. The failure of Newell and Simon's theory for complex problems led researchers to abandon the search for generic decision strategies and to focus their attention on in depth studies of decision making process for narrow problem domains.

Section C reviewed the expert-novice research in problem solving and discussed schemata theory. Experts were seen to be superior to novices in performing problem solving tasks in a domain by virtue of their greater quantity and quality (organization) of knowledge that they have acquired with experience in the domain. Schemata theory provides a theoretical foundation to explain how the knowledge acquired in a domain is organized in memory.

Section D discussed the implications of schemata theory and expert-novice research for the research and development of DSS. It was suggested that decision aids for diagnosing problems may have a greater payback than decision aids for the selection and evaluation phases of decision making. The importance of cue selection and cue combination was stressed as a key area for research and for consideration in the development of DSS. The design of the DSS interface to be cognitively compatible with the user is important in the design and development of a DSS.

# CHAPTER II

## MULTI-METHOD APPROACH AND RESEARCH HYPOTHESES

It is important that an overview of the methodology be presented prior to the discussion of the hypotheses in order to provide a context for the understanding of the terminology used in stating the hypotheses. The following section describes the theoretical basis for the methodology used. Chapter III presents the details of the implementation of the methodologies.

Within the Cognitive Science perspective, the key to understanding the decision making process is to make explicit the decision maker's representations of a problem domain. This project incorporates three techniques for eliciting the decision maker's knowledge: (i) concurrent verbal protocol analysis, (ii) aided retroactive verbal protocol analysis and, (iii) causal mapping. The first two techniques are part of a broad class of techniques known as process tracing.

### A - PROCESS TRACING METHODOLOGY

The activities that are related to the recording of the decision maker's problem solving processes during an actual problem solving session have been referred to as process tracing. The goal of process tracing is to make explicit the train of thought (cognitive processes) that allows an individual to come to a final decision or solution to a problem (Waldron, 1985). There are two basic approaches to process tracing:

1. Non Verbal. Some non verbal behaviour of the decision maker is recorded during his or her performance of the problem task.

2. Verbal. The decision maker is requested to provide a verbal report of his or her thought processes during performance of the task.

The main goal in recording the non verbal behaviour of decision makers is typically directed towards identifying the patterns in the decision maker's use of information during task performance. A decision maker is hypothesized to input information from the environment and to process this information to provide a solution to the problem (output). Identifying the nature of the information, the order in which it is sought, the frequency of use of information and the duration of information usage may provide valuable insights into the cognitive processes by which the decision maker solves a decision task in a domain.

Several nonverbal methods have been used to study information usage in decision making. Some of the more popular methods are (Payne, Braunstein and Carroll, 1978):

1. Recording eye movement. The task is designed so that the decision maker is presented with visual information and the eye movements of the individual are recorded during the task performance.

2. Explicit information search. The decision makers are presented with decision tasks for which they are required to search for information about the various alternatives. The information is concealed from the decision maker until the time that he or she actually seeks the information. The information may be organized on information boards, slide projectors, computer systems or any other reasonable method of presentation.

The second basic approach to process tracing involves giving the decision maker a task to perform and asking the decision maker to verbalize about his or her thought processes that occur during the problem solving activity. The verbalizations of the decision maker are referred to as protocols and the activities that are related to eliciting and analyzing the verbal reports are termed protocol analysis.

## A.1 Verbal Protocol Analysis

The use of verbal data to provide information about an individual's mental processes was popularized in the 19th century. Referred to as introspection, the technique was abandoned as a research methodology in the 20th century after being subjected to intense criticism by the behaviourist school. Today, as a result of the disrepute of introspection, there is a tendency to classify all verbal data research techniques as being limited to the role of hypothesis generation and to consider them worthless as a method of verification (Ericsson and Simon, 1980).

With the recent movement away from behaviourism and towards mentalism and information processing models of thought, researchers have turned again to verbal data research techniques (Johnson, Zualkernan and Garber, 1987). The contemporary verbal data methodologies are however, distinctly different from introspection and should not be subject to the same criticisms that plagued introspection.

Payne, Braunstein and Carroll (1978) distinguish between introspection and verbal data techniques in process tracing:

> 1. Subjects under process tracing are naive about the theoretical constructs of interest to the researcher. In contrast, highly trained subjects (sometimes the researcher himself) were used to generate introspective data.
>
> 2. In collecting verbal protocol data the subject is not asked to theorize about his behaviour. Instead the subject is asked "only to report the information and intentions that are within his current sphere of consciousness.
>
> 3. Verbal data may be collected during the actual performance of the task rather than through later questionnaires or interviews. This emphasis is related to the concern with obtaining measures of behaviour

> over time. In contrast, the collection of verbal reports after the response has been the traditional method of obtaining verbal data in psychology.

There are two basic considerations in the design of verbal process tracing: (i) the timing of the verbalization and (ii) the degree of probing of the expert by the researcher. The timing may be either concurrent or retrospective. In concurrent verbalization the decision maker is requested to verbalize or "think aloud" while performing a specific task. In retrospective verbalization, the decision maker is requested to recall the thought processes that took place after the performance of the task has been completed.

Retrospective verbalization has been severely criticized because individuals are not able to have access to their higher order mental operations and can not observe nor verbally report what they have done and why they did it (Nisbett and Wilson, 1977). Consequently they are subject to memory failures, and to the tendency to reconstruct stories that may be plausible or justifiable but not necessarily representative of what really happened (Svenson, 1989).

Concurrent verbalization is hypothesized to avoid the above limitation because of the fact that the verbalization takes place during the task activities and the individual's thoughts are resident in short term memory. Thoughts in short term memory are conscious. They do not require accesses to long term memory for expression and therefore may be verbalized. Ericsson and Simon state:

> When the subjects articulate information directly that is already available to them, the model predicts that thinking aloud will not change the course and structure of the cognitive processes. Nor will verbalization under these conditions slow down these processes.

In the case of concurrent verbal protocols the degree of probing of the expert may range from no probing at all (the expert is simply asked to think aloud) to a relatively structured probing (the researcher asks specific planned questions while the expert performs the task). The advantage of having no probing occur is that the research technique is less obtrusive. Structured probing, on the other hand, requires that the subject's task performance be interrupted and that the subject respond to the researcher's probes. The subject's train of thought may therefore be broken and the risk that the subject will frame his or her responses to address what is thought to be a response that the researcher desires will be increased.

The reliance on concurrent verbal protocols for eliciting the decision maker's representation of a domain is dependent on the researcher's ability to interpret the protocols accurately. For rather simplistic tasks such as cryptarithmetic and word puzzles this condition may be attained. For complex, less structured tasks, the ability of the researcher to interpret the protocols may be taxed. This is particularly true when the researcher is not an expert in the domain.

In this situation a form of retrospective verbalization may be used where after the decision maker performs the task, a memory aid may be presented to the decision maker to help him or her recreate the cognitive processes that were employed during the problem solving session (Waldron, 1985). Waldron (1985) suggests that videotaping or audiotaping the verbal protocols of decision makers while they perform a decision making task and then playing the tapes back for the decision makers can be a valuable memory aid for the decision maker. In addition it provides the researcher with the opportunity to probe the problem solver by asking questions to determine why certain actions were taken during the problem solving sessions.

In the case of aided retroactive protocol analysis, the use of probing may provide the researcher with richer data. Interruptions pose less of a problem as it is possible to stop the analysis or to backtrack to a previous decision point and recommence the analysis. In cases where the problem situations are complex and depend on a rich knowledge base, reliance on concurrent verbal protocols may not provide adequate data for analysis. This may explain why a recent trend in the research on expert - novice problem solving towards empirical methods based on answers to probes rather than pure "thinking-aloud" protocols (Patel and Groen, 1986).

In general, it has been recommended that verbal process tracing should be concurrent and non probing (Ericsson and Simon, 1980, Payne, Braunstein and Caroll, 1978, Todd and Benbasat, 1985; Svenson, 1989). However, the combination of concurrent and aided retroactive protocols may provide the researcher with the opportunity to gain richer data and with the potential to compare results of each technique to gain more reliable data and more valid results. Ultimately, the choice of verbal techniques should be contingent on the task domain and the goal of the research.

In this research project both concurrent and retroactive verbal protocol analysis are performed to elicit the decision making processes of decision makers for a software problem diagnosis task. The decision task requires the decision maker to operate a computer terminal during the problem solving session. The terminal screen and "think-aloud" verbal protocols are recorded on video tape. Subsequently the videotapes are played back for the decision maker and the aided retroactive verbal protocols are audiotaped for analysis by the researcher.

The "pure" concurrent verbal protocols are used to test hypotheses about the information item use and performance of the problem solvers. To test hypotheses about information item use it is imperative that the information items be concrete items used by the decision maker during a problem solving session and be selected as a result of the decision maker's decision making process and not as a result of confounding influences. Thus, the researcher should not influence the decision making process through probing or other obtrusive means.

The concurrent verbal protocols are used in conjunction with aided retroactive protocols to assess the decision making strategies that the decision makers use during the problem solving session. The purpose of the strategy analysis is more in the vein of exploratory research and as such the protocols should generate as complete a description of the decision making process as possible. Considering the complexity and richness of the domain it would be difficult to extract an in depth description of decision strategy from the concurrent protocols alone.

## B - CAUSAL MAPPING

Causal mapping is a specific technique for representing a person's causal assertions about a limited domain. The purpose of the technique is to portray relationships in a domain in a form which is amenable to study and analysis (Eden, Jones and Simms, 1983). The technique is based on George Kelly's Personal Construct Theory (Kelly, 1955), a theory that has recently been the recipient of renewed attention among cognitive theorists (Rorer and Widiger, 1983). Kelly suggested that individuals act as "personal scientists" forming hypotheses about the world, testing them against their experience base and revising them as a result of event outcomes to better fit their perceived reality of the world. In this manner, an individual evolves a system of constructs that categorize experience and classifies the environment. The construct system provides a template or frame of reference by which a person can discriminate between aspects of his or her world in order to understand and manipulate events in that world.

A construct within Kelly's theory corresponds closely to the cognitive scientists use of schemata. In the context of Kelly's theory, an expert would have developed, with experience, a well formed construct system to represent his or her domain and to provide guidance for dealing with domain problems. The expert would deal with problems in the domain by first comparing the situation with his or her construct system looking for similarities or marked dissimilarities. Attached to the results of this matching procedure would be assessments of the likely consequences of decision actions for the problem at hand.

The complexity and ill-structuredness of a problem domain would make it unlikely that an expert would make a single deductive leap from a decision action to an outcome. Instead, there would be several intermediate steps in the assessment where the effect of one event would in turn have other effects, and so on, creating an intricate causal network.

It is the structure of the expert's concepts or constructs and their organization into causal "schemata" that allow for a coherent interpretation of experience and form the basis of the specialized heuristic knowledge acquired by an expert in a domain (Tversky and Kahneman, 1980). Thus, in order to capture an expert's heuristic knowledge it is necessary to make explicit the expert's construct system and the linkages between these constructs. Making this knowledge explicit provides the initial structure that is basic to the development of a DSS.

Novices, on the other hand, would not have developed as "rich" a construct system as that of experts. Therefore the causal networks or schemata that represent the domain in the mind of the novice would have an inferior ability to diagnose problems in the domain. The inferior representation of novices would be reflected in their causal maps of the problem domain.

Causal assertions are defined relationships between variables. They are not intended to capture all the intricacies of a personal belief system nor do they represent non causal beliefs. Rather, they are intended to represent the simplifications or heuristic rules that an individual invokes to deal with the complex reality of the problem domain.

In order to elicit the subject's causal representation of the domain, the researcher conducted open ended interviews with each of the subjects after they had completed the two experimental diagnosis tasks administered by the researcher and the aided retroactive protocol sessions. The sessions were tape recorded and transcripts of the dialogue were prepared. Using the transcripts, the researcher then constructed a causal map of the subjects' representation for each the two diagnosis problem tasks.

## C - SUMMARY OF MULTI-METHOD APPROACH

This project employs three techniques for eliciting the decision maker's knowledge: (i) concurrent verbal protocol analysis, (ii) aided retroactive verbal protocol analysis and (iii) causal mapping.

The "pure" concurrent verbal protocols are used to test hypotheses about the information item use and performance of the problem solvers. Causal mapping is used to elicit the decision maker's causal representation of the domain in order to assess the quality of the decision maker's knowledge of a domain and to compare with the results of the "pure" concurrent verbal protocol analysis to test for convergent validity. The concurrent verbal protocols are used in conjunction with aided retroactive protocols to assess the decision making strategies that the decision makers use during the problem solving session. The purpose of the strategy analysis is more in the vein of exploratory research and as such the combination of protocol techniques should generate a richer and more complete description of the decision making process than reliance on one technique alone.

## D - RESEARCH HYPOTHESES

The objective of this research project is to identify the information items that experts and novices[2] consider while solving problem diagnosis tasks and the strategies that they use to process the information. Differences within skill groups and between skill groups are examined for two software diagnosis tasks of varying decision uncertainty.

An underlying assumption of this research is that experts are better than novices at solving problems in a complex, rich domain. Identifying similarities within skill groups and differences between skill groups should provide us with a better understanding of decision making processes of experts and novices. This knowledge may be applied to better understand what makes a decision maker skillful in a domain and to suggest key factors for the development of computerized decision aids for decision makers of all levels of skill in

---

[2]In this study all novices have completed a training course in the domain of expertise. The distinction between eperts and novices is described in Chapter III, section D.

the domain.

The research is based in schemata theory which provides a theoretical grounding for the differences in decision making skill between experts and novices. The following section provides an overview of the research hypotheses that are tested. After the overview, each set of hypotheses is explained in detail.

## D.1 Overview of the Research Hypotheses

Schemata theory explains how an individual organizes information in memory. Schemata represent knowledge that a person experiences, interrelationships between objects, situations, events and sequences that normally occur. Experts maintain a higher quality representation of a specific domain than novices. The quality of a representation relates to the amount of knowledge that the expert has acquired with regard to the domain and the manner in which the expert organizes that knowledge in memory (schemata). The first set of hypotheses, H1a through H1e test whether experts do in fact have a higher quality representation of a problem domain than novices.

The expert's higher quality representation of a domain allows them to diagnose difficult problems more accurately and in less time than novices whose schemata may be incompletely formed or non existent. Hypotheses 2.a through 2.g test the accuracy of decision makers and the time that they take to reach a solution by skill level for two problem solving tasks with varying degree of task uncertainty.

Experience in a domain should allow decision makers to develop a well-formed set of schemata to interpret problems within that domain. The schemata would enable decision makers to attend to pertinent information cues in the environment in order to solve domain problems. In turn the schemata that are triggered by information cues should provide the decision maker with problem solving procedures that should lead the decision maker to seek more information cues in the environment. The schemata structures of experts should be more extensive than those of novices and should contain more developed procedural attachments. The experts should therefore have a better awareness than novices of the key information items needed to solve any specific problem in the domain. This should lead to a greater consensus among experts than among novices for information items used during a problem solving episode. Hypotheses 3.a through 3.d test the level of consensus for information item use among decision makers. Hypotheses 3.e through 3.h test whether experts do consider more of the key information items identified as being the most important for solving the assigned problem tasks than novices.

The research reviewed above concerning expert-novice differences suggested that experts and novices do not differ in the strategies that they employ to solve problems but do differ in the quantity and quality of their representation of the domain. The last hypothesis H.4 tests whether in fact experts and novices use the same strategies to solve diagnosis problems.

The following sections present each hypothesis in detail.

**D.2 Quality of Representation**

It has been suggested that experts maintain in memory a higher quality representation of a problem domain than novices. The quality of a representation relates to the amount of knowledge that a decision maker has acquired with regard to the domain and the manner in which the expert organizes the knowledge in memory (schemata).

The causal map of each individual decision maker depicts key concepts (knowledge) that he or she perceives as being relevant to the domain and the causal relationships among the concepts (organization). It would therefore be expected that the causal maps of experts would display a higher quality representation of the domain than the causal maps of novices.

A causal map may be transcribed into an acyclic hierarchical tree representation of a decision maker's causal assertions concerning a specific domain. The hierarchical tree representations may therefore be compared across decision makers to assess the relative quality of their representation for the domain.

Three measures of the properties of the hierarchical trees may be considered to test for the quality of the decision makers' representation of the domain: (i) the depth of the tree: the greater the number of levels in the tree, the greater are the number of stages that the decision maker may consider to reach a decision and the more detailed the causal relationships in the map, (ii) the breadth of the tree: the greater the number of subtrees emanating from the root of the tree, the greater the number of causal relations directly leading into the goal node and therefore the broader the representation of the domain, (iii) the size of the tree: the greater the number of concepts depicted in the causal map, the greater the number of causal relations represented by the map. The rationale and operationalization of these measures will be discussed in more detail in Chapter III.

The following three hypotheses concern the quality of representation of the decision makers with respect to their causal representations of the domain as depicted on the causal maps.

HYPOTHESIS 1.a: The depth of the hierarchical tree representations of the causal maps of experts will be greater than the depth of the hierarchical tree representations of novices.

HYPOTHESIS 1.b: The breadth of the hierarchical tree representations of the causal maps of experts will be greater than the breadth of the hierarchical tree representations of novices.

HYPOTHESIS 1.c: The size of the hierarchical tree representations of the causal maps of experts will be greater than the size of the hierarchical tree representations of novices.

It was also suggested that during a problem solving session, experts refer to underlying principles of the domain more often than novices. An underlying principle of the domain may be considered to be a reference to a basic principle that would have explanatory power linking together perceived causal relationships. Knowledge of the basic principles of the domain has often been referred to as deep knowledge (Harmon and King, 1984). Novices, on the other hand, would depend more on heuristic relationships among concepts in their representation of the domain (surface knowledge). These relationships would be based more on pattern recognition than on interpretation of the problem.

For example, Chi et. al. (1982) found that experts in the domain of physics would consider the principles of mechanics, conservation of energy and Newton's force laws when solving problems in the domain. Novices, by contrast, would pick out key words in the problem statement and associate these words with perceived variables of interest and equations for solving the problems. The novices did not reference the underlying theoretical principles of the domain to solve the physics problems. Chi et. al. (1982) attribute the experts reference to theoretical principles as an indication that the experts have a qualitatively superior knowledge of the domain that is available to solve problems in the domain.

Basic principles are defined in this study as references to underlying causal relationships in the domain. To support deep knowledge the problem solver must have developed in memory explicitly defined relationships between facts and/ or concepts relating to the domain (Baldwin and Kasper, 1986). These explicitly defined relationships should be captured in the causal tree representations described above. It would therefore be expected that experts and novices would display differences in the use of deep and surface knowledge that parallel the differences in the causal tree representation of the task problem. In this manner, the results of the tests for the hypotheses relating to the causal tree analysis and to the use of deep and surface knowledge should converge. To the extent that they do converge, the results would increase the degree of confidence that the quality of the problem solvers' representations are being reflected by the measures applied in the study.

The following hypotheses relate to the expert's greater use of underlying principles to represent a problem domain.

HYPOTHESIS 1.d: Experts will make reference to basic principles in their causal map representation of a domain more ofter than novices.

HYPOTHESIS 1.e: Novices will employ greater use of pattern recognition during problem solving sessions than experts.

**D.3 Decision Making Outcome Performance**

The experts' superior representation of the domain permits them to outperform novices for more difficult diagnosis tasks in the domain. A difficult diagnosis task is defined as a task for which: (i) the solution is not trivial in the sense that the decision maker is required to select and assess multiple variables prior to making a decision and, (ii) the diagnosis task has been assessed by experts to require substantial knowledge of the domain.

The following hypothesis describes the relationship between decision accuracy and level of expertise. In testing this hypothesis the results of both decision tasks will be combined to give a global comparison of decision accuracy.

HYPOTHESIS 2.a: Experts will be more accurate in diagnosing difficult tasks in the domain than novices.

While experts may exhibit better performance on problem solving tasks in a domain than novices it is not necessarily the case that experts will perform at the same level for all types of tasks. In fact, experts would be expected to perform at a higher level for problems with which they are more familiar. For problems that are less familiar or less routine, it would be expected that overall performance of the experts would decrease.

In this research project the subjects are presented with two tasks to perform. The first task is in a class of problems that are frequently experienced by decision makers in the domain. However, the specific diagnosis problem that the subjects are requested to solve is a type of problem within the class that is encountered infrequently. The lack of familiarity with the diagnosis problem is what makes the task non trivial (section C.2 of Chapter III describes the tasks in detail).

Experts, by virtue of their vast experience in the domain will have developed an in depth representation of the class of problems for first task and will have likely encountered variations of the specific problem a few times in the past. Novices, on the other hand will have experienced the class of problems less often than experts and may have had no exposure to the specific problem at all. It would be expected that the greater experience of the expert would result in the experts having much greater success in solving problems of this type than the novices. Hypothesis 2.b tests this proposition.

HYPOTHESIS 2.b: Experts will be more accurate in diagnosing difficult tasks that belong to a class of problems that is frequently found in the domain than novices.

The second diagnosis task presented to the problem solvers falls into the class of problems that are very infrequently experienced by decision makers in the domain. Consequently, it is likely that an expert may have encountered a problem of this type only a few times in the past and a novice not at all.

The schemata to deal with this type of problem would be less well formed than for the first task. Therefore, both expert and novice decision maker will likely need to create new schemata or to recombine existing schemata in memory in order to reach a diagnosis. The lack of experience with the specific problem may result in less consistency among experts for diagnosing this task. This would imply that some experts may be inaccurate in their diagnosis for the type of problem for which expertise is most needed. It is expected, however, that the expert's superior knowledge of the domain in general should still allow him or her to diagnose the problem more accurately than novices.

The following two hypotheses test the above propositions.

HYPOTHESIS 2.c: Experts will be more accurate in diagnosing difficult tasks that belong to a class of problems that is infrequently found in the domain than novices.

HYPOTHESIS 2.d: Experts will be more accurate in diagnosing difficult tasks that belong to a class of problems that is frequently found in the domain than they will be for diagnosing difficult tasks that belong to a class of problems that is infrequently found in the domain.

The concept of the frequency with which problem tasks appear naturally in the domain may be considered to be a form of task uncertainty which parallels the notion of routine and non routine tasks. The greater the task uncertainty, the greater the need for the decision maker to process information in order to come to a decision and the more variability that is introduced into the decision process. The variability in decision process may be expected to lead to both greater variation in the information items used during a problem solving session and in the accuracy of the problem solvers.

There has been some evidence in the literature for hypotheses H2a through H2d. Lamberti and Wallace (1990) found that for software diagnosis problems both experts and novices performed a relatively routine diagnosis task more accurately than they did for a less routine problem task. The lower skilled problem solvers however, experienced a greater deterioration in accuracy performing the less familiar task than the higher skilled problem solvers.

Another facet of performance is the time required by a decision maker to reach a solution to the problem task. Experts are expected to diagnose a problem in a shorter period of time than novices by virtue of the superior quality of their knowledge of the domain. The schemata of experts will guide the selection and interpretation of information available to make a diagnosis allowing the expert to more quickly isolate the pertinent information and to process that information in memory and to ignore irrelevant information items.

The results of studies comparing the time performance of experts and novices for problem solving tasks have indicated that for a wide range of problem solving tasks experts have been documented to be faster than novices to reach a conclusion (Hershey et. al., 1990; Johnson, 1985; Lamberti and Wallace, 1990; Larkin et. al., 1980; Simon and Simon, 1978). Chi, Feltovitch and Glaser (1981) however, found that experts took more time than novices to categorize physics problems according to their similarities.

The following hypotheses compare the time required by experts and novices to arrive at a diagnosis for the problem tasks.

<u>HYPOTHESIS 2.e</u>: Experts will take less time to diagnose difficult tasks that belong to a class of problems frequently found in the domain than novices.

<u>HYPOTHESIS 2.f</u>: Experts will take less time to diagnose difficult tasks that belong to a class of problems infrequently found in the domain than novices.

Lamberti and Wallace (1990) found that experts performed more routine tasks faster than less routine tasks. Given the expert's substantial experience with the more familiar class of problems in the domain, it would be expected that they would take less time to diagnose the more frequent task than they would to diagnose the infrequent task presented to them in the study. Hypothesis H2.g tests this proposition.

HYPOTHESIS 2.g: Experts will take less time to diagnose difficult tasks that belong to a class of problems frequently found in the domain than they will take for diagnosing difficult tasks that belong to a class of problems that is infrequently found in the domain.

## D.4 Consensus for Information Cue Usage

Studies considering the number of information cues used during problem solving sessions and the consensus for information cue usage between and within skill levels of problem solvers have provided inconsistent results. Experts have been found to use more information items than novices (Ford et., al., 1989; Jacoby et. al., 1984), the same number of information items as novices (Grant and Marsden, 1988) and less information items than novices (Hershey et. al., 1990; Johnson, 1985; Kuipers and Kassirer, 1984). With respect to information item consensus, experts tend to display more consensus than novices for information item use (Johnson, et., al., 1982) or equally low consensus (Grant and Marsden, 1988). For information use within group, experts have been shown to display consensus (Einhorn, 1974, Johnson et., al., 1982) or fail to display within group consensus (Shields, 1983).

The variation in results concerning the quantity of information used and information cue consensus may be attributed to two factors: first the nature of the information cues being recorded and second the nature of the problem solving task.

It was suggested in Chapter I that for medical diagnosis tasks that are routinely encountered by expert physicians in a domain, the experts should exhibit consensus for the information cues that they use (Johnson et. al., 1982). The software diagnosis task that is performed by the subjects in this study closely resembles medical diagnosis tasks that require the problem solver to diagnose which of a number of possible conditions best describes a patient from a set of patient information cues.[3] The following hypothesis tests for information item consensus for expert problem solvers performing familiar tasks.

<u>HYPOTHESIS 3.a</u>: There will be a consensus among experts for the information cues selected to diagnose difficult decision tasks that belong to a class of problems that are frequently found in the domain.

For software diagnosis tasks that are rarely encountered by experts, the experts would not have existing schemata structures in memory that would guide information selection. New or modified schemata structures would be required by the decision maker. It would therefore be expected that the experts would exhibit less consensus for information cue selection during the performance of a decision task that belongs to a class of problems that are infrequently found in the domain. Hypothesis 3.b tests this proposition.

---

[3]The software diagnosis tasks and the information cues employed in this study are discussed in detail in Chapter III.

HYPOTHESIS 3.b: There will be less consensus among experts for the information cues selected to diagnose difficult decision tasks that belong to a class of problems that are less frequently found in the domain than for tasks that belong to problems that are more frequently found in the domain.

Novices, by virtue of their lack of experience in the domain, will exhibit less consensus for the information cues selected during the performance of a difficult decision task than experts would. This would be the case for decision tasks that belong to both frequently and infrequently found classes of problems in the domain.

Even though experts would show less consensus within their skill group for less routine decision tasks, their knowledge of the domain would guide their overall information cue selection enabling them to exhibit a greater degree of consensus in information cue usage than novices.

It should be noted that the opposite result may be experienced in certain cases. For example, if the novices have had a rigorous training program, then the training program may instill a pattern of responses for cue selection that may be used whenever the decision maker experiences a high degree of uncertainty regarding how to proceed. In this case, the novices may exhibit a higher degree of consensus than experts who may also find these problems unfamiliar, but with an anticipated lower level of accuracy than the experts.

HYPOTHESIS 3.c: Novices will exhibit less consensus than experts for the information cues selected during the performance of difficult decision tasks that belong to a class of problems that are frequently found in the domain.

HYPOTHESIS 3.d: Novices will exhibit less consensus than experts for the information cues selected during the performance of difficult decision tasks that belong to a class of problems that are infrequently found in the domain.

It has been proposed that experts will exhibit greater consensus for information item use for problem solving tasks than novices and that their performance for solving the problems would be superior to the performance of the novices. It therefore seems logical that the information items used by the experts are those that are key to solving the problems.

The use of key information items is related to the type of problem solving tasks that are being performed. For example, it may be argued that for certain tasks, experience may permit an expert to organize schemata at a high enough level to enable them to store a basic solution to the problem in memory. They may therefore be able to go right to the solution bypassing many of the information items that may be part of the key information set.

For the software diagnosis tasks employed in this study, it is necessary for the problem solver to systematically acquire information items that correspond to an ordered sequence of problem solving steps. It would be extremely difficult for a problem solver to eliminate the basic information search sequences required to diagnose the problem. Therefore there would be an expectation that expert problem solvers by virtue of their experience in the domain, would consider more key items during a problem solving session than novices.

The following hypotheses test whether or not the experts do in fact use more key information items than novices and to what extent they use what may be termed the relevant set of information items for solving a problem task in the domain.

HYPOTHESIS 3.e: Experts will consider a greater number of the key information items for solving a problem task than novices during the performance of a problem solving task.

HYPOTHESIS 3.f: Experts and novices will consider a greater number of the key information items for solving a problem task that belongs to a class of problems that are found frequently in the domain than they will for a problem task that belongs to a class of problems that is found infrequently in the domain.

The ability to recognize and consider key information items in the software diagnosis problem solving process will enable the decision maker to more accurately solve the problem task. The greater use of key information items would in part explain the superior performance of experts compared to novices in the domain. Hypothesis 3.g tests this proposition.

HYPOTHESIS 3.g: The decision makers who use the greatest number of key information items during the problem solving session will have the most accurate problem solutions.

## D.5 Decision Making Strategy

It was suggested that experts display decision making performance that is superior to that of novices because of the quality and organization of information that exists in their memory rather than their use of superior decision making strategies. Nonetheless, decision making strategies are important because they dictate the order in which information is sought for diagnosis and the method with which it processed in memory. For example, a decision maker may adopt a backward reasoning strategy in which he or she may first generate a hypothesis and seek information that would support or reject the hypothesis, continuing until a hypothesis is accepted. Another strategy might be to adopt a forward reasoning strategy in which the decision maker may seek information cues and dependent on the values of the cues combine them to provide sufficient evidence to choose a goal state that satisfies the problem task. Another possibility is that the decision maker may use a hypothetico-deductive reasoning process which employs a combination of backward and forward reasoning.

The strategy adopted by a decision maker would be an important consideration for the design of a DSS, in the sense that it would suggest patterns of information presentation to best match the decision maker's cognitive processing of information. In addition, if certain strategies were shown to be more effective for certain problem tasks, this information could be used in training and in the design of DSS.

To the extent that novices receive training in the domain by experts and are in daily contact with the experts during the performance of their decision tasks, the strategies that experts and novices use to make diagnoses for problem tasks should be similar (Elstein, Shulman and Spafka, 1978; Johnson et.al., 1981; Glaser, 1984). Recently the similarity of strategies between experts and novices has been challenged. Patel and Groen (1986) have suggested that experts use a process of forward reasoning based on highly elaborated representation of the problem and that novices tend to use processes that involve backward reasoning.

Hypothesis 4 tests whether or not the decision strategies used by expert and novice decision makers are similar.

HYPOTHESIS 4: Experts and novices will exhibit the same decision making strategy for solving problem tasks in the domain.

## E - SUMMARY

In this chapter an overview of the research techniques used to elicit the problem solvers' representation of the domain was presented. Three techniques are used, (i) concurrent verbal protocol analysis, (ii) aided retroactive verbal protocol analysis and, (iii) causal mapping.

The research hypotheses tested were described. The hypotheses are classified into four groups. The first group of hypotheses deals with a comparison of the quality of representation of the problem domain between and within expert and novice skill groups. The second set of hypotheses compares the decision making performance of the problem solvers between and within expert and novice skill groups. The third set of hypotheses considers the degree of consensus among problem solvers that exists for information item use between and within expert and novice skill groups. The fourth set of hypotheses considers whether or not expert and novice problem solvers use the same strategies.

The next chapter describes the methodology employed for this research project.

# CHAPTER III

## RESEARCH METHODOLOGY

### A - INTRODUCTION

The purpose of this study is to identify and compare the information items that experts and novices consider while solving software diagnosis tasks and the strategies that they adopt for processing the information. Underlying the investigation is the notion that experts perform better than novices for tasks requiring domain expertise. It was suggested that the superiority of the experts' performance is not a consequence of their using superior problem solving strategies but rather a consequence of the experts having acquired more highly developed schemata of the domain by virtue of their vast experience in solving domain problems.

This study first tests the hypothesis that experts do in fact have a greater quantity of information and higher quality representation of the domain than novices. It then tests whether or not experts outperform novices for diagnosing problems in the domain. The consensus among and between experts and novices for information items is then assessed and a test is performed to identify whether experts consider a greater number of "key" information items during a problem solving session than novices. The final set of hypotheses test whether or not experts and novices use the same strategies to process information during problem solving.

A major contribution of this research project is the multi-method approach adopted to test hypotheses pertaining to the objectives stated above. Using this approach permits the collection of richer data than one method alone could provide and permits the researcher to test for convergence of methods. The test for convergence of methods is particularly important given the small sample size and the nature of the data (protocols). This chapter describes in detail the methodology used.

The research approach consists of a controlled experiment. The major advantage of an experiment is to provide the researcher with the ability to manipulate the independent variables (expertise, problem task) in a controlled setting. Given that the objectives of the research are to compare subjects' use of information items and strategies for solving problem tasks, it is imperative that the problem tasks administered to the subjects be identical in content and in setting.

Exercising control over the administration of problem tasks does not preclude having the tasks patterned after real life tasks that the subjects perform on a day to day basis. In fact a major strength of the research design in this study is that it uses real subjects performing real tasks, that is, subjects whose occupation requires them to solve problems of identical form to those being manipulated in the experiment. In addition, the experiment was conducted at the subjects' place of work, in a setting that closely resembled their natural decision environment.

One limitation of much of the research conducted in problem solving is that the tasks used are fabricated for the laboratory and that the subjects are often drawn from an artificial population (students for example). This is particularly true of research in MIS and has led MIS researchers to call for the use of more realistic tasks in the experimental design (Jarvenpaa, Dickson and DeSanctis, 1985). This study addresses these issues of validity.

## B - RESEARCH SITE

The experiment was conducted in a large branch office of a multinational manufacturer and distributor of computer products located in a major Canadian urban center. The subjects for the experiment were all employees of the company working in the branch office's technical support center. One responsibility of the technical support center, and the major responsibility of all the subjects in the experiment, is to field calls from clients who have experienced a system failure on their mini or mainframe computers and to diagnose the source or cause of the failure.

A major cause of system failures can be attributed to software problems and the decision tasks that are employed in this study relate to diagnosing failures due to such software problems.

When a client experiences a system failure the client determines whether the problem is software or hardware related. If the problem is deemed to be software related, the client performs various standard software diagnostic checks that have been recommended by the host company. If the client can not identify the specific component that is failing, a call is placed to a technical support representative who is responsible for servicing the specific operating system installed at the client site. The technical support representative will assist in diagnosing the failing component and in determining the cause of the failure. Once the problem has been diagnosed, the information will be communicated to the client who will take remedial action or the problem will be passed on to another department in the company for follow up.

To assist in the diagnostic task, the technical representative has available several dumps of computer memory that have been automatically provided by the system at the time of failure. The contents of the dumps are organized by the operating system of the client's computer into several titled reports. The nature of the reports are dependent on the type of failure and the software installation at the client's site.

The dump reports are made available to the technical support representatives by the client and form the basis for diagnosing the problem. The technical support representatives may receive hard copy print outs of the dump reports or they may access the reports via an online interactive data management system that has been developed by the host company. In the current study all technical support representatives were accustomed to using the online system and only the online facility was employed during the experiment.

When a technical support representative receives a client request to diagnose a problem, the first step is to determine from the dump what type of problem it is. There are five basic categories of problems. The first category is a hardware problem. A hardware problem occurs only 5% of the time[4] because if the client has identified that the problem is hardware related the client would have forwarded the problem to a hardware specialist and not to the software technical support representative. In the case of a hardware problem the problem is transferred to a hardware specialist. The second category of problems is related to a user error. A user error occurs 21% of the time and is caused by the client incorrectly using a given program. The technical support representative

---

[4] The statistics on the type of problems and their frequencies were supplied by the host organization.

will refer the client to the program application source for help.

The third category of problems turns out not to be a problem at all but a misunderstanding by the client which may be cleared up relatively quickly. This non-problem occurs 30% of the time.

If the software technical support representative cannot attribute the problem to one of the above categories he or she will conclude that it is a software problem. In this case the technical support representative will check to see if it is a known software problem which has previously been identified and for which a diagnosis has already been made. The fourth category of problems consists of problems of this type for which the diagnosis is documented.

To determine if the problem is a known software problem a description of the problem is entered into a problem diagnostic database that is connected world wide by satellite. The data base contains problem descriptions for problems that other company employees worldwide have had to deal with in the past and the action that was taken to solve the problem. If the problem is found then the diagnosis of the problem is given. The technical support representative will find the solution to his or her client's problem in the data base 40% of the time.

That leaves only 4% of the time that the problem will be software related and will not be found in the database. When the problem is not in the database, the technical support representative begins a problem diagnostic routine to find the source of the problem. It is in this last category that the two experimental problem tasks are found.

## C - TASK DESCRIPTION

The choice of problem tasks for the experiment was critical for achieving the objectives of the study. It was necessary for a problem task to meet rigid criteria in order to be considered for the study yet at the same time the task had to emulate very closely a real life decision task. A number of tasks were considered in different subject domains. The tasks selected for the study were chosen because of their compliance with the criteria listed below and the ability to implement the manipulation of task familiarity without complication.

### C.1 Criteria for task selection

In order to be considered for the experiment a task had to meet all of the criteria listed below.

1. The task requires expertise based on specialized knowledge of the domain.
2. The task could be performed in a period of time not exceeding 60 minutes.

3. The task could be performed by one individual without consulting other persons during the problem solving session.

4. The task is a diagnosis task requiring the decision maker to select relevant information cues from the decision environment and to combine them to diagnose the source of the problem.

5. There are no existing algorithms or routinized formal support procedures that enforce a specific decision making process.

6. Experts and novices exist within the decision domain and may be readily distinguished. An important consideration for the project is that the novices be well versed in the domain but lack the experience to be considered experts. For example, a medical intern may be considered to be a novice in a domain even though he or she has had significant medical training, whereas a specialist who has practiced in the domain for several years may qualify as an expert.

7. A measure of decision accuracy must be available to enable the performance of the decision makers to be assessed.

To identify suitable tasks for the experiment the researcher met with a software diagnostic expert in the host organization and described the objectives of the research project and the criteria for task selection. The expert communicated the task requirements to a structional manager at the national training center located in another city. The manager was a software diagnostic expert himself and had extensive experience in designing test problem cases to administer to students in the training programs. After interacting with the researcher and the expert several times, the instructional manager identified four problem tasks that he thought would be suitable for the study. One task was an actual task experienced by a technical support representative who had been transferred to another department. The other three tasks were designed by the instructor to emulate realistic client software problems.

One of the tasks that was designed by the instructor was eliminated immediately because it was felt that the task could not be completed in sixty minutes. The task that was previously experienced by the technical support representative was eliminated because there were missing data in the dumps that were deemed to be necessary for the diagnosis and that could not be reconstructed. The final two tasks met the criteria established by the researcher.

To ensure that the tasks were realistic, they were developed by the instructor to reflect real problems encountered in the work environment. The instructor created the problem situations by purposefully introducing an error into the operating system software. The errors introduced were unique in the sense that they were selected to provide output that could not be considered to be consistent with more than one possible diagnosis. That is, the dumps of the system at the time of the error could not, to the best knowledge of the instructor and the expert, be correctly interpreted to be caused by more than one specific sequence of events.

This characteristic of the task is an important factor in the design of the study for two reasons. First, it is important to be able to devise an accurate measure of decision performance that is comparable across subjects. Second, if more than one valid alternative is available to the decision makers then a new factor is introduced into the study of the decision making process, namely the process for discriminating between potentially valid solutions paths. The necessity to consider this additional process would change the nature of the study and add a new level of complexity to the analysis. In addition, the small sample size would make the comparison across multiple alternatives difficult to operationalize.

The instructor defined what he would consider to be a correct diagnosis and this definition became the measure for task performance accuracy for the experiment (section G.3 describes the measurement for decision accuracy).

To ensure that the two tasks were suitable for the purposes of the experiment and that the instructor's solution was in fact unique and accurate, the expert solved the diagnostics task himself and gave the tasks to a second expert for an opinion.

The two tasks that were chosen were both classified as system abends. An abend (abnormal end) occurs when a task being performed by the computer system terminates prior to its normal completion because of an error condition. Both abend tasks were designed to be "difficult" to diagnose. A difficult diagnosis task was defined as a task for which: (i) the solution is not trivial in the sense that the decision maker (expert or novice) is required to select and process multiple information cues prior to making a decision and (ii) the problem has been assessed by experts to require substantial knowledge of the domain for accurate diagnosis.

It should be noted that non triviality does not necessarily correspond to task complexity. For example, a task that is considered to be very complex because of the number of steps required to solve the problem and the vast number of interrelationships among variables may in fact be trivial for a problem solver if he or she has performed that task several times before.

If a trivial task was chosen for the experts or the novices then the performance of the task by the subjects may have become a highly automated or programmed process that would be rapid and less accessible to studies of decision process and verbal protocols (Kuipers and Kassirer, 1984; Svenson, 1989). For this reason, the tasks had to be chosen carefully so that neither experts nor novices would be highly familiar with the specific abend problem tasks. The two diagnosis problems that were chosen were both unfamiliar, infrequently occurring abends.

What differentiated the two tasks from each other was the degree to which the class of problems that each of the two abends belonged to was typically encountered by the technical service representatives in the normal course of performing their jobs. The first problem task belonged to a class referred to as an "OC4" abend. An OC4 abend is the most frequently encountered type of abend problem. The second problem task belonged to a class of problems referred to as a 30A abend. The 30A abend is an infrequently encountered abend.

In summary, the specific OC4 task that was chosen for the experiment was a relatively unfamiliar abend that was in a class of abends routinely encountered by the technical support representatives. An OC4 problem may be encountered several times during a week. A problem of the nature of the specific OC4 task chosen likely would not be encountered more than two or three times during a year.

The specific 30A task that was chosen for the experiment was an unfamiliar abend that was in a class of abends rarely encountered by the technical support representatives. In a typical year a representative may expect to experience a 30A problem three or four times during a year and a problem of the nature of the specific 30A task that was chosen more than likely, not at all.[5]

---

[5] Statistics on the frequency of the two tasks selected for the study were not gathered by the host organization. The frequencies mentioned were derived from a consensus of two experts and the instructional expert.

The degree to which software diagnosis problem tasks frequently or infrequently occur in the domain may be considered to reflect a form of task uncertainty. The construct of task uncertainty however, encompasses considerably more scope than the familiarity of a problem task. For this reason the frequency of a software problem occurring in the domain will be considered to reflect "task uncertainty" in only a limited sense.

## C.2 Problem task characteristics

This section describes the two software diagnosis problem solving tasks in detail. The descriptions include the general class of problems, the characteristics of the specific abends chosen for the study and the elements of the problem that are critical for solving the tasks. Before describing the two abend tasks chosen for the study a discussion of the software diagnosis task in general is presented to provide a context for the experimental tasks.

Reiter (1987) describes a diagnosis problem in simple terms:

> If we have available an observation of the system's actual behaviour and if this observation conflicts with (ie. is logically inconsistent with) the way the system is meant to behave, then we have a diagnostic problem. The problem is to determine those system components which, when assumed to be functioning abnormally, will explain the

discrepancy between the observed and correct system behaviour.

The notion of a system's behaviour conflicting with the way the system is meant to behave may be applied broadly to include systems as diverse as the human body (medical diagnosis) to automobile engines (mechanical diagnosis). The major differences in diagnosis tasks for different systems stems from the characteristics of the systems and the implications of these characteristics for the methodology required to identify the components which are functioning abnormally and causing the incorrect system behaviour.

For the software diagnosis tasks[6] employed in the study, the problem solvers are presented at the onset with the knowledge that a system abend has occurred and that a partial dump of the contents of the computer memory has been automatically formatted into a number of reports that are available to the problem solvers for consultation. In addition, the general class of the problem (OC4 or 30A in this case) is provided in the headings of the dump.

The class of abend provided to the problem solvers may be considered to be a general statement at the highest level of

---

[6]The experimental software diagnosis tasks have been patterned after "real" software problems that may naturally occur in the subjects' work environment. Therefore, the description of the tasks given here would apply to software problems in the domain.

abstraction of the symptoms[7] of the abend. For example, an OC4 problem means that a program interruption has occurred for which there has been no routine specified. Associated with the class of problem are various descriptors that may be found in the dump that provide lower level descriptions of the symptoms of the abend. For example, an interruption code may be sought in the dump that provides more problem specific symptoms. In the case of the OC4 problem, the interruption code may take on one of three values: 4, 10 or 11. In the experimental task the interruption code is 11, signifying that a page translation exception fault has occurred.

A major step in the diagnosis of the software problems is to establish what the symptoms of the abend are at a level of specificity that enables the problem solver to hypothesize a cause of the observed symptoms. Establishing the symptoms of the problem may be considered to be a generic diagnosis activity. For example, in the domain of medical diagnosis, a physician may be presented with a patient who complains of a sore arm (highest level of abstraction). The physician may then feel the arm and check the swelling (second level symptom). The physician may then have an xray taken of the arm and look for a fracture (a third level symptom) and so on

---

[7]A symptom is defined to be any state of events that may be determined to be an artifact of the problem that is taking place. That is its existence may be seen to be dependent on the specific problem. For example, swelling of the arm or fever may be considered to be symptoms related to a patient's medical problem. An invalid branch from one memory location to another may be a symptom of the abnormal abend.

until enough information is gathered to diagnose the cause of the symptoms.

The degree of specificity of the symptoms gathered is dependent on the needs of the problem solver. For example, the physician may be able to diagnose the problem as being a fractured wrist without having an xray taken or alternatively may have an xray performed as well as several additional tests. Establishing the symptoms of the problem at a sufficient level of specificity to diagnose a problem may require considerable expertise in the domain. For example, the physician must know what information is necessary and which tests to perform to accurately diagnose the problem.

The software diagnosis task is similar to the medical task in that the problem solver must determine which symptoms related to the abend to look for during the problem solving process. It may not be obvious which symptoms to look for nor may it be obvious where the symptoms may be found in the dump reports. Causal knowledge of the class of problem task may provide important guidelines for the problem solvers' search for symptoms. For example, knowledge that a fracture can cause swelling and pain may lead the physician to request an xray of the arm or conversely, coming across the fact that there is a fracture may lead the physician to check if there is also swelling in the arm. In this manner, compared to novices, experts in the domain may be better able to diagnose a problem by virtue of their higher quality causal representation of the problem task permitting them to search and find symptoms relevant for the problem.

There is a significant difference between medical problem diagnosis and software problem diagnosis. In the case of the medical diagnosis, the physician is typically involved in an interactive process of history taking from the patient (Patel and Groen, 1986) which allows him or her at any point in the diagnosis process to seek data that is not present at the onset of the problem solving session. In the case of software diagnosis, the pertinent data required to solve non documented problems is generally contained in the dump reports. If information that is sought by the problem solvers is not in the dump, then the fact that it is not present may provide meaningful information about the problem. In any case, the problem solver will most likely need to do without the information.

It is interesting to note that in many of the research studies of expert-novice differences for the performance of diagnosis tasks, the researcher, with the intention of restricting the study to diagnosis performance, will provide the problem solvers with all the necessary or desired data for solving the problem (for example, see Patel and Groen, 1986). In the present study, the subjects were provided with all the information necessary to make a decision. This restriction is less artificial for the case of the software diagnosis task than it may be for other domains such as medicine because of the infrequent requests of the software problem solvers for additional information.

For the software diagnosis problems, it is necessary to first identify where the abend that caused the abnormal termination actually took place before beginning a search for symptoms. This knowledge is required to provide guidance to structure the search for symptoms (it provides the addresses in memory to begin the initial search for symptoms) and to provide a context for interpreting symptoms. Determining where this abend occurred may be non trivial. For example, in the 30A problem there are two abends in the dump. The problem solver must be aware that there are two abends and that the first abend is a user generated normal termination and that the second abend is an abnormal termination that occurs while the system is trying to execute the user's normal termination.

Once the location of the abnormal termination is identified, the problem solver can then begin to establish the symptoms of the problem. To accomplish this the problem solver must determine where information relating to symptoms may be found in the dump reports. Knowing where to look is often a difficult task for a problem solver.

Finding a symptom may occur in one of three ways: first, the problem solver may seek out a specific symptom in the dump that he or she anticipates will be present, second, the problem solver may go to an address in memory or to a specific part of the dump reports with the expectation that this is a likely place to find an unspecified symptom and third, the problem solver may come across a symptom by chance. In the last two cases the problem solver must be able to recognize that a symptom is present.

To direct the search for symptoms and to interpret their implications for diagnosing the problem, it is suggested that the problem solver uses his or her knowledge representation of causal factors (schemata) relating to the class of software problem task. Once an adequate number of symptoms have been identified, the problem solver may with some confidence state the cause of the software abend.

The last component of the software diagnosis task is to verify that the proposed cause of the abnormal abend is in fact correct. Given the large number of potential causes of system abends this step plays an important role in confirming the diagnosis. The information to confirm a diagnosis is typically found within the dump. For example, the OC4 abend is caused by an overlay in a register that causes the system to branch to an address, F7E000, that is incorrectly offset by one byte. The correct branch should be to 00F7E0. To verify that this is in fact the correct diagnosis the problem solver could look up the location 00F7E0 in memory to determine if the contents of that address contains a valid instruction and to verify that the sequence of code displayed to either side of the address 00F7E0 makes sense.

This method of confirmation of a diagnosis is quite different from several other domains. For example, a medical diagnosis may be confirmed in a number of ways. First a symptom may confirm a diagnosis. If a physician determines upon examining a patient's swollen arm that a fracture has occurred then he or she may confirm the diagnosis by taking an xray of the arm. In the case of the OC4 problem the confirmation of the diagnosis takes into consideration information that is not a symptom of the problem nor is it related directly to the problem. A second way in which a physician may confirm a diagnosis is by treating the patient for the diagnosed problem and watching to see if the patient improves. This option is generally not available for the software problems because the technical service representative's job is not to remedy a problem but only to diagnose its cause and to pass it on to an appropriate person for remedy. It would be an exceptional case where the technical service representative would pass on a problem to another individual without being confident that the problem is accurately defined. A third way for a physician to verify a diagnosis is to simply wait and see how the symptoms progress over time without treatment. The software abend is a static event and does not evolve over time. Therefore this alternative is not feasible.

In summary, four steps in the diagnosis of the software problems have been identified: the first step is to determine where the abend is actually taking place, the second step is to determine the actual type of abend at a high level of abstraction to focus the search for symptoms (for example, for the OC4 determine that the interruption code is 11 and invalid branch takes place), the third step is to diagnose a cause for the abend and the fourth step is to verify the diagnosis.

It should be noted that the order of these steps may vary by problem solver. For example, rather than proceed through the steps in the prescribed order, the problem solver may generate a diagnosis prior to performing steps 1 or 2. This may be the result of past experience, perhaps by using prior probabilities for the causes of OC4 problems, or may simply be a hypothesis that is generated intuitively early in the problem solving process and is based on the consideration of few information items.

If a diagnosis is made without completing step 1 or 2, the problem solver will find that he or she is unable to verify the diagnosis. It is necessary to know where the abend occurred and the type of abend to pursue information to confirm a diagnosis. It may be possible to generate a diagnosis that is correct and not verify it. However, without having performed step 1 or 2, it may not be possible to provide enough detail of the problem to meaningfully pass it on to another person for problem resolution. In any case, there would be no real context with which to justify or support a diagnosis.[8]

The need to perform an information search before being able to verify a diagnosis is quite different from many other domains. For example, in automobile mechanic diagnosis the problem solver may diagnose from the initial symptom "the car won't start" that the battery is dead. He or she may then simply replace the battery to verify the hypothesis.

**C.2.a OC4 task characteristics**

An OC4 abend is an instance of the general class of abends known as OCx abends. An OCx type abend is a general category of error which is characterized by a program interruption but for which no routine has been specified. The last digit "x" of the OCx is a hexadecimal number that indicates the cause of the program interruption, in this case an OC4.

For the selected OC4 problem a slip dump is provided in addition to the standard dump generated by the dumping services for an abend. A slip dump is generated by a slip command and is intended to intercept or to trap specified system events at the time of an abnormal termination. The slip

---

[8]The individual problem solver's approach to software diagnosis is described in more detail in section D of Chapter III under the heading of strategy for diagnosing software problems.

command is entered by a system programmer, either at the console or via the input stream. It is therefore a dump that originates with the client.

At the time the system is experiencing an error or interrupt, the system will compare the slip dump qualifiers with the actual system conditions. If there is a match, then the system will execute the slip dump providing additional dump information to that which is normally provided by the dumping services during an abnormal termination.

The problem solver will be made aware of the nature of the abend (OC4) and that a slip dump has been generated immediately by the title of the dump file which is presented as an "OC4 SLIP DUMP."

In the case of an OC4 problem, the interruption is described by one of three reason codes: 4, 10 or 11. In the problem task at hand the reason code is 11, specifying a condition known as a PIC 11, page translation exception.

A PIC 11 means that a page fault has occurred. That in itself may not be an error, but when a PIC 11 is accompanied by an OC4, it means that the system encountered an error while a virtual page is trying to be brought into storage.

There are generally two causes for an OC4 - PIC 11. The first cause occurs when an error is committed in attempting to access storage that is not authorized for access. The error arises from specifying the wrong address, that is the wrong location to go to find the storage that is being sought. The second error occurs when the system is pointing to storage that has been referred to by a register that has invalid contents. In other words, the address that is being sought is correct, but that location does not hold the expected values. The first type of error is the more common and a variation of this type of error was selected for the problem task.

In general, OC4's are so common that most instances of an OC4 problem are found in the organizational diagnostic data base. What makes this particular OC4 task difficult is the fact that it is an unfamiliar type of OC4 abend that is not found in the diagnostic data base of the company. Therefore the technical support representative must diagnose the problem from scratch.

In the OC4 problem chosen for the study, the source of the problem is a bad branch instruction. An attempt is made to branch to a location that is invalid causing a system abend. The branch instruction that is failing is referred to as a branch and link register 14-15 (BALR 14-15) which is a branch from register 14 to register 15. Register 14 contains an instruction that when executed directs the system to a location in memory to fetch the contents of this address and place it in register 15.

Tracing backwards through the instructions that lead up to the BALR 14-15, to verify what value should have been placed in register 15, it may be determined that register 15 was loaded from a location specified by the contents of register 2 plus an offset of 355. The offset of 355 should be recognized as being a strange value as it is extremely rare to load an odd offset which would result in an address that was not on a full word boundary. It would be much more likely that the offset would be 354 or 356.

By going to look at the address specified by register 2 plus 355 in storage it should become apparent to the problem solver that the load value requested should have been 00F7E0 which is on the full word boundary rather than F7E000. Thus the source of the problem is a bad branch instruction that is offset one byte from where it should have been.

The likely cause of this problem is that somehow there was an overlay of storage, modifying the offset value in the branch instruction and creating the error termination. To confirm the diagnosis the address F7E0 should be found in storage to verify that it in fact has valid code at that location.

To diagnose the problem the problem solver would be expected to experience the following insights during the problem solving process. First he or she would have to realize that the problem involves a BALR 14-15. Second, it should be noticed that the load instruction is on an odd value and that it would be more appropriate to load the instruction on a full word boundary. This particular insight requires some subtle reasoning by the problem solver. Third, it should be noted that address F7E0 is on a full word boundary and that it should probably replace F7E000. To complete the diagnosis the validity of the contents of F7E0 should be verified to support the conclusions above.

## C.3 30A Task Characteristics

The 30A problems are a general class of problems relating to an error that occurs during the execution of a freemain macro instruction. The system was trying to free storage in a particular location and an abnormal termination resulted. Associated with a 30A problem is a reason code. The reason code distinguishes among several different kinds of freemain errors that can occur. In the case of the 30A problem selected for the study the reason code is 18, signifying that a private area subpool was not found.

This particular 30A is complicated by the fact that the system 30A abend was issued by a previous user abend. That is there are two abends in the dump. The first user abend is a normal termination requested by the user's application program that is running at the time. While trying to execute the user's termination an error occurred which issued an abnormal abend 30A. In fact the first users abend is perfectly normal and has nothing to do with causing the 30A abend.

This is an important factor for the problem solver to comprehend. If the problem solver does not realize that this abend is not implicated in the 30A abend and that a second abend that followed the first abend is where the diagnostic effort should be placed then considerable time and effort may be wasted. The clues required to recognize this situation are subtle and therefore contribute to the high degree of difficulty for this problem task.

Once the problem solver focusses on the 30A abend he or she must determine the reason code in order to find out what kind of 30A they are dealing with (either a getmain or freemain problem) and then verify that it is in fact a freemain problem by applying a sign test (positive or negative) to the value in the register where the abend took place.

Once it is verified that the abend is a freemain problem the problem solver can look up the register that contains the address of the register to be freed. The address turns out to be 31 bit addressable (in high storage) but a SVCA (the abend instruction) is used for 24 bit addressability. This is a key observation in the problem solving process.

At the time that the problem solver is concentrating on the SVCA he or she should see several SVC78's preceding the SVCA and notice the similarities in the addresses that appear in the registers associated with the SVC78 and the SVCA. This should alert the problem solver to the real source of the 30A abend, that is the fact that there might have been an overlay here and in fact the wrong SVC may have been issued. A verification in the manual will show that an SVC78 can be used to freemain storage that is 31 bit addressable. A further verification can be made by searching for the address that issues the SVC and check whether there is in fact an SVCA or an SVC78 there. This information is stored in a report called an AMBLIST. The problem solver must ask for that report as it is not automatically included in the dump. If this check is made, the problem solver will in fact find an SVC78 at the issuing location confirming that an overlay had taken place.

## D - SUBJECTS

The subjects consisted of technical support representatives who were all employed in the same branch office and whose responsibilities at the time of the experiment included diagnosing client software problems for systems running under a specific proprietary operating system. The subjects were selected from a list of technical support representatives who had volunteered to participate in a study of problem solving. They were told that they would be required to devote a few hours of their time during normal work hours to perform typical diagnostic tasks and to discuss the tasks with a researcher. Aside from this information the subjects were not given any additional information about the purpose nor details of the study.

While using real subjects to perform real tasks in a real setting has the potential to increase the validity of the experimental process, it also introduces the potential to confound the results due to a fear on behalf of the subjects that the results of the study will be used in some way by their employer to measure their job performance. Motivated by fear, the subjects may be inclined to say and do what they perceive would be expected from them by their superiors rather than what they would normally do in the experimental situation. Fear could also induce stress which could interfere with performance of the problem tasks.

To help allay these fears, the subjects were assured when they were asked to volunteer and again prior to participation that the research was in no way being sponsored by their employer and that all results would be kept in strict confidence. During the study the subjects were randomly assigned alphanumeric identification numbers and told that all references to individuals would be made by code and that their names would not be linked to any of the results of the study.

Four experts and four novices were selected to participate in the study. Expertise was defined as the possession and organization of domain specific knowledge and procedural skill that can be accessed and efficiently applied during problem solving (Vitalari and Schenk, 1989).

The host organization has a departmental classification system that identifies five formal levels of expertise among technical support representatives beginning with a student software technician at stage one and ending with a principal software technician at stage five. The novice subjects were selected from the third level of expertise and the expert subjects were chosen from the fifth level.

In order to reach the third level, the technical support representative must complete a number of courses including an advanced training course in the specific operating system used in this project. A technical support representative will typically reach this stage after having completed 2 years of service in the technical support department. To reach the fifth level the technical support representatives must complete a minimum of five years of service diagnosing software problems after having completed the advanced course.

Subjects were chosen from the third and fifth levels to help ensure that there would be a significant difference between their level of expertise. Subjects below the third level did not have enough training nor experience to meaningfully interpret the decision tasks employed in the study and any tasks developed that would be meaningful for this group would be trivial for the experts.

In addition to being in level five, the experts selected had to be confirmed by management as having a perceived expertise that made them superior performers in the department.

Another requirement of the subjects was that they had to be familiar with the interactive online data management system that was being used to help diagnose software problems. Not all technical support representatives use the online system, some preferring to work with hard copies of the dumps.

If a significant difference of skill in using the online system existed among the subjects then this might affect their ability to solve the diagnostic tasks or the speed in which they could perform the task. To control for this factor, only technical support representatives who were actively using the online system to diagnose software problems at the time that the experiment took place were considered for participation in the study.

## E - PRETEST

The pretest activities took place in a branch office of the organization that was located in a different city than the research site. The technical support representatives at the pretest location had the same responsibilities and performed the identical tasks as the subjects.

A different city was selected for the pretest site to minimize the opportunity for confounding the results of the experiment. First, it was felt that the presence and visibility of the researcher at the test ıte for a prolonged period of time prior to the actual experiment might have had an effect on the subjects if they had been selected from the same site. The researcher was observed by the technical support representatives to be conferring with management on many occasions over a period of several months. The closeness of the contact of the researcher with management might have led the subjects to suspect the objectivity and motivation of the researcher. Secondly, even though all participants in the pretest agreed to keep any information about the study confidential, it was felt that the opportunity for leakage of information about the experiment to potential subjects would be minimized by changing research sites.

The researcher worked closely with an expert at the test site to gain an understanding of how the technical support department functioned and to verify that the decision tasks that had been selected for the study were suitable. After having ascertained that the tasks were in fact suitable for the study, the tasks were given to test subjects to perform in order to debug the procedures, to test the performance of the decision makers and to practice protocol techniques. Sessions with the test subjects were open ended allowing for two way communication with the researcher.

Upon completion of the pretest a pilot study was conducted at the test site under experimental conditions using four subjects (two experts and two novices) to ensure the adequacy of the experimental design.

## F - EXPERIMENTAL DESIGN

### F.1 Subjects

The sample consisted of four experts and four novices selected from a list of volunteers who met the criteria described previously. Prior to the researcher arriving on site to conduct the experiment a manager of the technical service department, in consultation with other department managers, constructed a list of employees in the department who were classified by the organization as level 3 or level five and who had experience in the operating system associated with the experimental tasks. The managers verified the list to ensure that their perceptions of the employee's skill matched the organization's classification.

The department manager approached each employee on the list individually and explained to them the need for participants in a research study and requested that they consider volunteering for the study. A list of volunteers was then compiled and the manager randomly selected and approached experts and novices from the volunteer list until four experts and four novices had committed to participating in the experiment during the two weeks that the researcher had designated.

To obtain descriptive information on the subjects a questionnaire was developed. After all the subjects had completed their participation in the study a structured interview session was conducted with each subject to complete the descriptive questionnaire. An interview was selected rather than requesting the subjects to fill out the questionnaire on their own for two reasons. First, depending on the answers to the questions the content and sequence of the questions varied. It was felt that it would be easier for the interviewer to determine the correct sequence of questions than it would be for the subject, who would have to interpret a set of instructions explaining the question sequencing. Second, certain questions required a follow up probe for specific responses to the questions posed. A copy of the questionnaire my be found in appendix I.

The questionnaire is divided into four sections. The first section records the subjects' age and sex. The second section poses questions concerning the subjects' formal work experience. Formal work experience may include work experience with the host company, with the technical support department or in a similar position with another company. The degree of formal work experience may be interpreted to be an indication of the subjects' general level of experience in the domain.

The third section considers the education that the subjects have received. There are two classes of education, formal education and company sponsored courses. Formal education refers to the technical college, commercial college, university or other formal education that the subjects have completed. Company sponsored education refers to only those courses or programs offered by the company that are specifically related to the operating system for which the two experimental tasks represent.

The last set of questions assesses the degree of involvement or concentration of the subjects' experience in diagnosing software problems fo~ the operating system in question. This information is pertinent to accurately compare subjects. For example a person working five days a week, full time with the operating system for two or three years may in fact have gained greater experience in the job than a person working only one or two days a week at diagnosing software problems for a much longer term.

The results of the questionnaire for the first and second classes of questions indicate that experts tended to be: (i) male - all four experts were males compared to two males and two females for novices; (ii) to be older (m=43,range=39,46) than novices (m=25,range=24,26); (iii) to have worked longer for the company (m=17,range=14,24) than novices (m=3.35,range=2,4.5) and to have worked longer for the department (m=7.25,range=5,9) than novices (m=2.4,range=2,3).

Novices tended to have a higher level of formal education (three university graduates for novices and none for experts), but tended to have taken less company courses concerning the operating system (m=2,range=2,2) than experts (m=3.75,range=3,4)

Experts have more years (m=9,range=5,13) of experience diagnosing software problems than novices (m=2,range=1,3) but currently tend to spend about the same amount of time (10-20 hrs/wk) during the week diagnosing problems. Table III.1 summarizes the descriptive data for the subjects.

The descriptive data supports what prior expectations would dictate: experts tend to be older, to have worked longer for the company and to have more job experience and training than novices. An interesting observation is what appears to be a movement towards hiring university graduates for technical support service representative positions and to employing women for these jobs. This finding lends credence to the propositions often found in the practitioner literature that the technical business community is demanding ever more increasing qualifications for entry level positions and that large organizations are recognizing the potential contribution of women for technical jobs.

## G -MEASUREMENT

The following sections describe the measurements employed in the study for the variables of interest. A general description of protocol analysis is presented first as it has implications for several subsequent measures that are dependent on the protocols.

**TABLE III.1**

DESCRIPTIVE STATISTICS FOR SUBJECTS BY SKILL GROUP

## G.1 Analysis of Verbal Protocols

The methodology used to analyze the verbal protocols is described in this section. The actual experiences during the implementation of the methodology is described in Chapter IV. The description of the analysis of verbal protocols is presented in two parts: first, the conceptual foundation for the analysis is discussed and second the mechanics for implementing the analysis is presented.

### G.1.a Conceptual foundation

To a great extent the usefulness of the results of this research project depend on the quality of the analysis of the verbal and aided retrospective protocols generated during and after the problem solving session. For this reason, it is important that an analysis framework is devised that is grounded in the theoretical foundations of the research and that allows for the interpretation of the data in a manner that addresses the research questions proposed by the study.

A solid theoretical basis is required because verbal protocols tend to disclose much of what the subject is actually doing in terms of information processing behaviour, rather than disclosing why or how it is being processed. Svenson (1989) states:

> Most verbal protocols actually contain a lot about the information processed and very little or nothing about how it is processed. Therefore it is important to have a good theory or model which makes it possible to understand the data in a process-oriented language. Very rarely do verbal protocols themselves provide the theory for their understanding.

The theoretical foundation for this study is based in schema theory. Schema theory suggests that when faced with a problem a problem solver will seek information from the problem environment that will allow him or her to activate schemata in memory or to develop new schemata. The activated or new schemata will in turn direct the problem solver's search for other information in the decision environment until enough information has been acquired to reach a conclusion.

The research hypotheses are directed towards two aspects of the above process. The first aspect concerns the information items that are considered by the problem solvers during a problem solving session. The second aspect studies the information processing strategies that the problem solvers use to guide the acquisition of the information items and that lead to a problem solution.

These two classes of hypotheses may be considered to study two distinct dimensions of decision making. The first dimension, information processing, is concerned with those processes that are involved in the organization, integration and use of information. The second dimension, information evaluation is concerned with those processes that lead directly to a decision (Schweiger, Anderson and Locke, 1985).

Two methods of analysis are employed to study these two dimensions. The first is a scoring method that permits the researcher to identify and to compare the information items that are considered by the problem solvers. In order to score the verbal protocols a method for external coding must be adopted since the verbal protocols by themselves are not quantitative measures and are not amenable to scoring (Schweiger, Anderson and Locke, 1985).

The second method is a more qualitative approach which involves studying the protocols for evidence of a pattern of behaviour that may characterize a decision strategy. The underlying theory of problem solving directs the researcher's efforts in finding patterns of behaviour. The coding of protocols is a means of describing the decision processes that are observed in the problem solver's behaviour and is therefore instrumental in assisting the researcher to find patterns in that behaviour.

The method used in coding the protocols is basic to achieving the research objectives. Several different coding schemes have been used in the literature. The actual coding scheme that is adopted should be a function of the theoretical foundations of the study, the research objectives, and the characteristics of the problem task.

A coding scheme used by Ericsson (1975) and reported in Ericsson and Simon (1984) was adopted for this study and was modified to better meet the objectives of the research. Ericsson's model coded protocols into four general classes.

1. Intentions, which is information representing goals or future states of the subject.
2. Cognitions, which is information based on attention to selected aspects of the current situation.
3. Planning, which is information representing intermediate constructions to explore sequences of possibilities mentally.
4. Evaluations, which are explicit comparisons of alternatives.

The coding scheme adopted for this study uses four classifications for protocols: (i) Intentions, (ii) Cognitions, (iii) Hypotheses and (iv) Conclusions. "Intentions" and "cognitions" are similar to Ericsson's use of the same terms. "Hypotheses" are similar to planning but are expanded to have more of a causal explanatory context than Ericsson intended. "Conclusions" differ from Ericsson's evaluations in that they are information concepts that the problem solver derives from the processing of previous information and that become stored as facts in his or her representation of the problem state.

An "intention" is defined as the identification of an information item by the problem solver that he or she intends to search for during the problem solving session. "Intentions" are therefore important indicators of the information that the problem solver's schemata are directing the problem solver to search for. A "cognition" is defined as an information item that the problem solver selects for consideration from the current situation. That is, it is a part of the current problem task. As such, "cognitions" are taken to be facts that exist in the information given by the problem. "Cognitions" correspond to information items that are sought and selected by the decision maker from the decision environment.

A "hypothesis" is defined as a statement that suggests a cause or reason for a perceived state for which the problem solver has not yet been able to verify. "Hypotheses" are often generated on the basis of the problem solver becoming aware of cognitions (facts presented in the problem environment) or of conclusions (facts that are deduced and which are not present in the problem environment). A "hypothesis" may be generated as an intermediary step in the problem solving process or as a final causal diagnosis of the problem. For example, during the problem solving session for the OC4 problem an intermediary "hypothesis" may be generated that suggests that an invalid branch has been executed causing the abnormal abend. The "hypothesis" of the invalid branch is in fact correct but at a level of specificity that is not detailed enough to present a final diagnosis. Later in the problem solving process a second and final "hypothesis" may be generated that suggests that the value of the offset for the branch instruction is off by one byte and should be 354 instead of 355. This "hypothesis" is also correct, but provides enough detail of the problem to conclude the diagnosis process.

A "conclusion" is a statement of fact that the problem solver has reached from the consideration of previous information items. It differs from a cognition in that its content is not directly available as an information item that exists in the problem environment. One form of a "conclusion" is the verification of a hypothesis. It may in fact be the same statement as a "hypothesis" but this time it is stated as a belief rather than as a possible causal statement. For example, once the offset "hypothesis" described above has been verified by the problem solver, it no longer has a hypothesized existence but rather its existence is now being stated as a fact. "Conclusions" need not be hypothesis driven but may simply be deductions made from information items that have previously been considered by the problem solver.

A "conclusion" has been described as a belief. Therefore it may not be an accurate assertion about the environment. What it is however, is the problem solver's perception of a true state in the environment and as such it influences the problem solving process. "Conclusions" may be repudiated or changed if the problem solver finds contradictory evidence during the problem solving process.

## G.1.b Mechanics of protocol analysis

The first step in the analysis of the protocols is to make a complete transcript of the verbal reports. After the transcripts are complete it is possible to eliminate verbalizations that are not of value in the analysis. In eliminating this data strict guidelines must be followed to ensure that important information is not deleted as well.

In order to keep the analysis task manageable, not all information need be retained in the transcripts. Information that is related to straight descriptions of mechanical activities as they are being performed are presumed to hold no value in understanding the problem solving processes and in fact it may not be safe to extrapolate from this kind of verbalization to conclusions about problem solving (Ericsson and Simon, 1984).

In this study the subjects perform a task using a computer terminal as a tool to access information. Many of the verbalizations are related to the mechanics of using the computer system. A hypothetical example of these kind of verbalizations might be: " Now lets see. Press F8 then PAGEDOWN to get to - no F9 will get me to where I want to be." Verbalizations of this type were eliminated.

Once the non essential verbalizations have been eliminated and the final transcripts of the verbal reports are prepared it is necessary to make a decision on how to present the verbalizations for analysis. There are a number of alternatives available. The protocols could be given in entirety, they can be aggregated according to some theme (for example, episodes) or they can be broken down into small phrases or segments. In this study the protocols were broken down into segments that correspond roughly to what might be considered a single task assertion or reference by the problem solver (Ericsson and Simon, 1984). The segments may be considered to loosely represent "thought units" in the sense described by Newell and Simon (1972), but may contain more than "one single idea."

Kuipers and Kassirer (1987) suggest that how the transcript is broken down (size and content of segments) and recorded is not critical, but rather the format should ease the burden of later analysis. This comment applies to the current study. What is important is that the protocols be accurately classified and not their segmentation on a piece of paper.

After the transcripts were prepared, the segments were coded according to the four classes described above. The actual coding of the segments into classes is a difficult process. In order to increase confidence in the accuracy of the coding, random segments of the protocols were given to a second rater for coding and interrater reliabilities were calculated for the second rater and the primary researcher.

The actual transcribing, coding and reliability calculations will be described in detail in Chapter IV.

## G.2 Quality of Representation of the Problem Domain

Hypotheses 1.a through 1.e concern the quality of the experts' and novices' cognitive representations of the problem domain. The quality of these representations is defined as the amount of knowledge that a decision maker has acquired with regard to the domain and the comprehensiveness of the schema structure organizing the knowledge in memory. Two sets of measures of quality were employed. The first set of measures is derived from the subjects' causal map representation of the domain. The second set of measures distinguishes between the subjects' use of deep and surface knowledge to diagnose a software problem.

### G.2.a Causal tree measures of representation quality

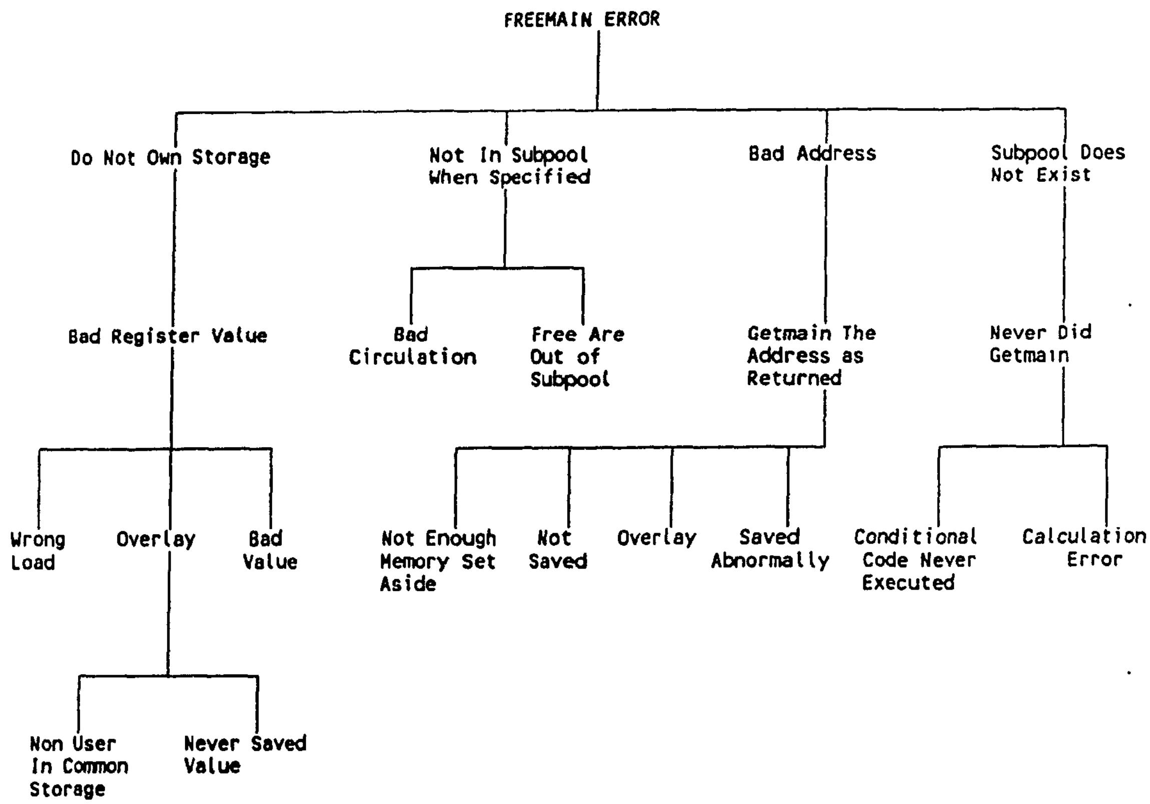
In order to compare the representation of experts and novices at the conceptual level open ended interviews were conducted to elicit the subjects' perceived causal relationships in the domain. The causal map of each individual decision maker represents the causal assertions of the decision maker for the domain. The causal maps of the subjects can be transcribed into an acyclic hierarchical tree representation of the causal relations. An acyclic hierarchical tree is a directed tree with the following properties: (i) there is exactly one vertex, called the root which has no entering edge. The root relates to the goal state in the causal map, (ii) every vertex except the root (concepts on the causal map) has exactly one entering edge and (iii) there is a path from the root to each vertex.

Figure III.1 gives an example of a causal map elicited from an expert for the class of software diagnosis problems described in this study as being "difficult-infrequent" (30A problem). The root of the tree presents a statement of the problem class being considered, namely an abnormal termination that results from a freemain error. "Do not own storage" and "bad address" are examples of vertices that represent factors that may have a causal influence on the freemain error condition.

In general, more abstract classificatory concepts tend to be higher up in the tree structure and more specific concepts are lower in the structure. This corresponds to the discussion mentioned earlier concerning the software diagnosis task and the search for symptoms. Each level down in the hierarchical tree structure may be considered to represent a symptom (from a higher level) at an increasingly specific and detailed level of analysis. For example in Figure III.1 "do not own storage"

**FIGURE III.1**

AN EXAMPLE OF AN EXPERT'S CAUSAL
MAP FOR THE UNFAMILIAR 30A TASK



would be a more abstract level symptom of the problem. Each level down, "bad register value", "overlay" and "never saved value" provide increasingly more specific symptomatic information about "do not own storage." In this context, the diagnosis task may be considered in simplistic terms as identifying the node in the tree that offers an adequate level of specificity necessary to meaningfully diagnose the problem.

Given that the causal tree is generated by the subject and not by the researcher, the causal tree may be observed to reflect the problem solver's causal scheme or representation of a class of software problems and as such may provide structure for interpreting the problem task and for guiding the problem solvers diagnosis process.

A node or concept in the tree should not be confused with the notion of an information item. An information item has been defined as an information cue that may be found in the information set provided to the problem solver at the onset of the problem solving session. In the context of the software diagnosis tasks in the study, this would mean that the information items exist in the dumps of memory that are available to the problem solver. Information items may provide cues that allow the problem solver to either confirm or reject the existence of a symptom of the problem described by a node in the causal tree.

For the OC4 problem for example, a node in a causal tree may contain the symptom referred to as an "invalid branch."

The problem solver may become aware of the following facts that are available in the dump: (i) register 15 at the time of error is at fault, (ii) register 15 contains an address that can not be found in memory (iii) just prior to the address in memory containing the contents of the register 15 is a branch and link instruction. The problem solver may now conclude from these information items that an "invalid branch" has occurred. It should be noted that the level of specificity of this node is not adequate to give a satisfactory diagnosis of the problem. The problem solver may now engage in an information search that attempts to describe the problem at a lower level of symptomatic detail.

With respect to the coding scheme applied to the verbal protocols, a node in a tree corresponds to either a "conclusion" or an "hypothesis" and an information item corresponds to a "cognition." In the OC4 example above, the node "invalid branch" is a "conclusion" deduced from a series of information items or "cognitions." An "hypothesis" may also represent a node in the tree. For example, a problem solver may generate the hypothesis that an invalid branch has occurred based on the high prior probability of occurrence of branch and link instructions. He or she may then search for the same information items that have been described earlier to verify the hypothesis. Once these items have been found, the problem solver may then conclude that an invalid branch has occurred. The main difference between an "hypothesis" and a "conclusion" in this example is the search strategy employed to diagnose the problem. In the first case forward reasoning is invoked to reach a conclusion. In the second case backward reasoning was applied to instantiate a hypothesis. The problem solvers' problem solving strategies will be discussed in more detail in section D of Chapter IV.

Three measures for the quality of the problem solvers' representations of problem task are derived from the hierarchical causal tree representation: (i) a measure of the depth of the tree, (ii) a measure of the breadth of the tree and, (iii) a measure of the size of the tree.

The depth of the tree is a measure of the problem solvers' ability to represent a symptom of the abend by a causal chain of events at increasingly more specific and detailed levels of description. To the extent that the necessary causal relations exist in the tree to diagnose a problem at an adequate level of specificity, the problem solver should be able to perform the problem solving task more effectively than a problem solver whose causal representation of the domain is less well developed. It is suggested that experts have a deeper and better developed causal representation of the class of problem tasks than novices.

The standard measure of the depth of the tree is the tree's height which consists of one plus the number of the highest level in which it has nodes (the root is level 0), or equivalently the longest path from the root to a leaf in the tree. A leaf of a tree is a vertex for which there is no directed edge exiting the vertex. For example in Figure III.1, the root of the tree is "freemain error", "never saved value" is a leaf, the height of the tree is 5, which is the path from the root to a leaf at level 4 (eg. "never saved value").

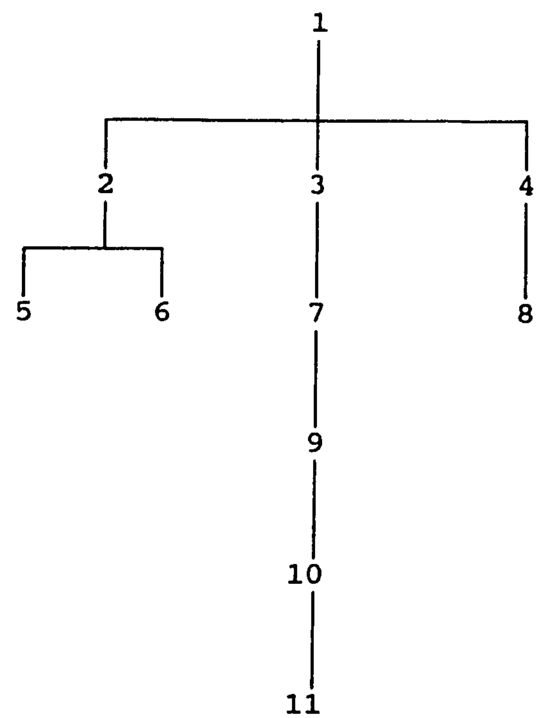
The height of the hierarchical tree representation of the causal map is not by itself sufficient to measure of the quality of the representation. For example, in Figure III.2 the tree for subject A has a height of 6 whereas the tree for subject B has a height of 5. The representation in tree B has a lesser height but the tree indicates a higher quality representation by virtue of the greater number of vertices at level 3 and 4.

To rectify this problem, the measure of depth for the tree has been modified to be calculated as a weighted measure that considers the number of nodes at each level of the tree multiplied by the level of the tree. This weighting scheme was adopted because it was felt for example, that vertices at level 5 in a tree displayed a longer more detailed sequence of causal relations than vertices at level 4 and should therefore receive a stronger measure of quality. A sparse tree such as the tree for subject A in Figure III.2 should not be considered to display a greater quality representation with respect to depth than subject B. Multiplying the number of vertices at each level by the actual level that they were at would ensure that a tree such as that depicted for subject B

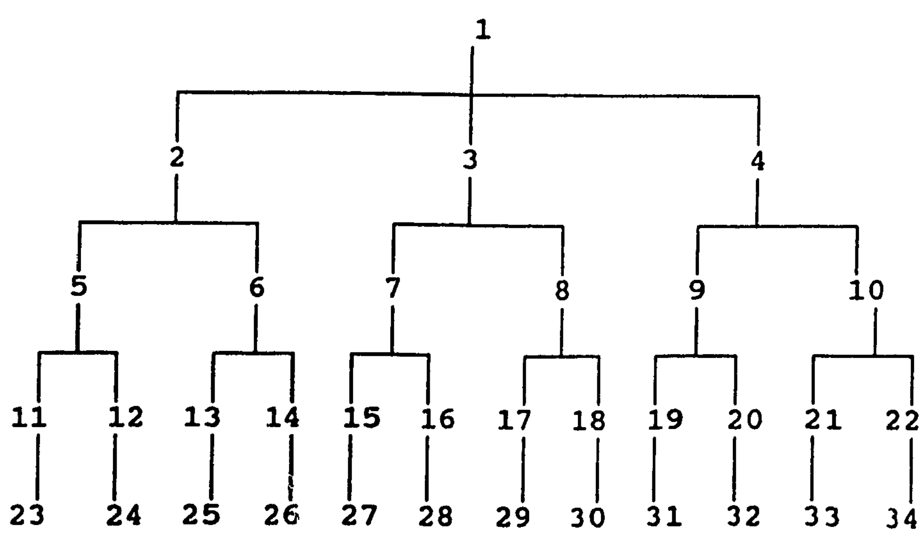
**FIGURE III.2**

A COMPARISON OF TWO CAUSAL TREES

TREE FOR SUBJECT A




TREE FOR SUBJECT B



in Figure III.2 would not receive a lower depth measure than the tree of subject A.

The calculations for the depth of the two trees in Figure III.2 is given below. In this measurement approach tree B is found to have a greater quality representation of the domain as measured by the depth of the two causal trees.

```
DEPTH TREE A = 3 vertices times level 1 =  3
             + 4 vertices times level 2 =  8
             + 1 vertex   times level 3 =  3
             + 1 vertex   times level 4 =  4
             + 1 vertex   times level 5 =  5
                                          -----
                          total           23

DEPTH TREE B = 3 vertices times level 1 =  3
             + 6 vertices times level 2 = 12
             +12 vertices times level 3 = 36
             +12 vertices times level 4 = 48
                                          -----
                          total           99
```

A standard measure for the breadth of the tree is the degree of the root. The degree of a vertex is the number of subtrees of that vertex. In Figure III.2 the degree of tree A is 3 which is the same for tree B. The degree of the root will be used in this research project as a measure of breadth. The breadth of the tree measures the ability of the problem solver to identify classes of symptoms at the most abstract level of analysis that may be the cause of the problem. The ability to generate abstract classes permits the problem solver to represent the problem in a more structured and directed way. This may result in the problem solver being able to direct the problem solving search along narrower paths thus increasing the effectiveness of problem solving performance. Experts would be expected to display broader representations than novices.

The size of the tree will be measured as the total number of vertices in the tree. In Figure III.2 the size of tree A is 11 and the size of tree B is 34. The number of nodes in the tree is a direct measure of the number of causal relationships described in a tree. An expert would be expected to have a richer, larger tree than novices.

**G.2.b Deep versus surface knowledge representations**

The second measure of quality of representation considers the number of references to the underlying principles of the domain, that is to deep knowledge. Several representations of deep knowledge are conceivable: mathematical and simulation models of a complex process, physical laws that govern a situation, functional models of how a device works, detailed causal networks and collection of the rules of the form "If in (situation), (action) is taken, (situation) will follow (Chandrasekaran and Mittal, 1983).

In this study an underlying principle or "deep knowledge" is defined as a reference to a causal network of relationships in the domain. The concurrent verbal protocols of the subjects that are elicited while they perform the problem tasks are analyzed to identify the subjects' references to underlying principles of the domain (scoring of the verbal protocols is discussed in Chapter IV, section A). The total number of references is compared between and within subject skill groups for each task as well as for the two tasks combined.

There is a direct association between the causal tree representations of the problem solvers and their use of deep knowledge during their performance of the problem solving tasks. Both the causal tree and the references to deep knowledge reflect the problem solver's representation of cause-effect relationships in the domain. It is therefore expected that the results of the two methods of measuring the quality of the subjects' representation of the class of problems will converge.

The causal maps of the subjects' representation was elicited from unstructured interviews with the subjects and the references to deep knowledge were derived from the verbal protocols of the subjects that were recorded during the subjects' performance of the problem solving tasks. Given the diverse source of the two measures, convergence of the results could provide strong evidence for the validity of the measures.

## G.3 Decision Accuracy and Speed

Hypotheses 2.a through 2.d consider the accuracy of the subjects' diagnoses. The measurement of decision accuracy was based on: (i) discussions with a manager who was an instructor at the corporate training center and an expert in software diagnosis for the class of tasks introduced in the experiment and (ii) on a detailed analysis of the diagnosis problem tasks. As described previously in section C.2 of Chapter II, four steps in the diagnosis of the software problems have been identified: the first step is to determine where the abend is actually taking place, the second step is to determine the actual type of abend error that occurred, the third step is to diagnose a cause for the abend error and the fourth step is to verify that the diagnosis is in fact valid.

For each of the four subtasks that the subject accurately diagnoses, the subject scores .25 points[9]. If the subject accurately identifies all four subtasks then he or she receives a perfect score of 1.0. Note that in order to receive the .25 points for each subtask, the identification must be based on analysis and not for example, on random suggestions

[9]The measure of accuracy was based on a discussion of the organizationally sanctioned methodology for software problem diagnosis taught by the instructional department. Each step in the problem solving process was identified as being an important element for diagnosing a problem by the instructional expert. A problem solver who had accurately completed more of the steps than a second problem solver, even though not having completely diagnosed the problem (score of 1.0), would be considered by the instructor to have displayed superior diagnosis performance.

as to the type of error that occurred or on the random generation of hypotheses.

The assignment of equal weighting for the four components of an accurate solution was based on the observation that all four components are important for the accurate diagnosis of a problem. While the assumption of equal weighting was rather arbitrarily defined, it provided the advantage of being able to get an objective assessment of accuracy with relative ease from the protocols recorded during the problem solving sessions. An alternative method that was abandoned because of the complexity of the task and the inherent potential to introduce bias in scoring was to have at minimum one expert review each problem solving session on tape and score the performance of the problem solver.

Hypotheses 2.e through 2.f consider the time taken by the subjects for the problem solving tasks. Each subject was given a maximum of 60 minutes to diagnose each problem task. Decision speed was measured by the total time (minutes) taken by a subject for one of the following conditions: (i) time to complete the diagnosis task (ii) time up to the voluntary termination of the session before presenting a diagnosis or (iii) the session reaches the set 60 minute time limit.

## G.4 Consensus for Information Item Use

Hypotheses 3.a through 3.d relate to the information item use of the subjects. Information item use is defined as the consideration by the subject during the problem solving sessions of information items that exist in the memory dumps that have been issued by the system at the time of the system failure. The information items correspond to what has been designated as "cognitions" in the scoring of the protocols (section G.1).

Three different measures of consensus are used in this study. The first measure of consensus identifies the total number of information items that each subject considers during a problem solving session. The total number of information items measure is a simple count of the items used and does not include any information about the content of the items.

The second measure of consensus considers the degree of agreement within skill level group for each single item and calculates a frequency of agreement table. That is, for each item the number of experts and the number of novices who use this item is recorded. The frequency of agreement table would thus present for the four expert subjects and the 4 novice subjects the total number of times: 4 experts agreed on an item, 4 novices agreed on an item, 3 experts agreed on an item, 3 novices agreed on an item and so on for all possibilities.

The third measure of consensus considers the subjects' use of "key" information items that have been designated as being important for the diagnosis of each problem task. The methodology for determining which items are "key" is discussed in section C.3 of Chapter IV.

In order to organize the information item data to facilitate the calculations of the above measures an information item use matrix was constructed. The information item matrix presents an exhaustive list of the information items used by all subjects and enables a visual comparison of the use of information items within and between subjects groups. Each row in the matrix represents a specific information item and each column represents a subjects or subject class total. The last column represents the information items that have been identified as being key items.

In section G.1 of Chapter III, an information item was described as corresponding to a "cognition" in the coded protocols of the subjects' problem solving sessions. To construct the information item matrix for each problem solving task, the "cognitions" of one subject were identified and the information items corresponding to those "cognitions" were recorded in the first column of the information item matrix. A "1' was then placed in the column corresponding to the subject for each row that an information item appeared. The "cognitions" of a second subject were then identified. If the information item corresponding to the "cognition" already appeared in the table then a "1" was placed in the column designated for subject corresponding to that information item. If the information item had not previously been recorded in the table then it was added to the bottom of the list of information item. This procedure was repeated for each of the subjects and for each problem solving task. A hypothetical example of the information item matrix is presented below.

| | EXPERT | | | | | NOVICE | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | E1 | E2 | E3 | E4 | TOT | N1 | N2 | N3 | N4 | TOT | K* | TOTAL |
| ITEM1 | 0 | 1 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| ITEM2 | 1 | 0 | 1 | 1 | 3 | 1 | 1 | 1 | 1 | 4 | 1 | 7 |
| ITEM3 | 1 | 1 | 1 | 0 | 3 | 1 | 0 | 1 | 0 | 2 | 0 | 5 |

* K represents key items

## G.5 Decision Making Strategy

Hypothesis 4 considers the decision making strategies employed by the subjects during the problem solving sessions. Decision making strategy refers to the general strategy used by the subjects to control information search and interpretation. In assessing decision strategy the concurrent and aided retroactive verbal protocols of the subjects performing the problem tasks were assessed qualitatively.

In general, practitioners of protocol analysis have suggested that a subject should be required only to express his or her thoughts that are going on at the time that the problem solving behaviour is actually taking place. Newell and Simon (1972) comment on both verbal and retroactive protocol analysis:

> The protocol is a record of the subject's ongoing behaviour, and an utterance at time t is taken to indicate knowledge or operation at time t. Retrospective accounts leave much more opportunity for the subject to mix current knowledge with past knowledge, making reliable inference from the protocol difficult. Nor, in the thinking-aloud protocol, is the subject asked to theorize about his own behavior - only to report the information and intentions that are within his current sphere of conscious awareness. All theorizing about the causes and consequences of the subject's knowledge state is carried out and validated by the experimenters not by the subject.

In this study think-aloud concurrent verbal protocols are recorded in the strict sense of Newell and Simon's recommendations. The subjects are requested only to verbalize the thoughts that they have during the problem solving sessions. The problem that arises is that the think-aloud protocols are not sufficient in themselves for the researcher to reconstruct the strategies that the problem solvers are using during the problem solving sessions.

There are a number of reasons why this situation arises. First, the nature of the task is such that the interpretation of the reasoning behind the protocols is difficult to determine. The task is highly complex, having many potential solution paths for the problem solver to explore in a highly unstructured problem environment. Therefore, given a sequence of protocols it may be difficult to determine whether the sequence was for example, intended to combine information items to generate a problem solving hypothesis or was in fact intended to collect evidence in support of a hypothesis.

This is particularly true of the given problem task. The nature of the task requires the problem solver to consult pages of dump data presented in numerical format. Many of the verbalizations of the problem solvers would be a sequence of memory addresses or register contents that would be reported as hexadecimal values. The hexadecimal values reported would be the same values considered for many different solution paths and consistent with more than one strategy. Given this situation, it may be difficult for the researcher to interpret the protocols without some additional aid.

A second problem of interpretation is that the researcher, while having a background in computer science, is not an expert in this particular problem domain. Given the complexity of the domain, it may at times be difficult for the experimenter to achieve insight into the strategies adopted by the problem solvers.

It is interesting to note that being an expert in the domain may not be an advantage for interpreting the protocols, but may rather introduce subjective bias to the analysis. During the pretest, an expert was given a partial list of think-aloud protocols for one decision session and asked to comment on what he thought the decision maker was doing at that time. After the expert stated his opinion, the researcher asked him how he came to that conclusion. He responded that given the addresses that the problem solver was looking at he must be searching for information on X because that is what he would be doing if he was looking at those same items.

In order to obtain additional data, the subjects were shown video taped replays of their problem solving sessions and requested to explain what they were doing. The researcher probed the subjects with directed questions when it was unclear, from the think-aloud protocols and the subjects retroactive explanations, what was transpiring.

The combination of verbal, think-aloud protocols, videotape of the computer terminal screen and retroactive protocols provided rich and complete data for analysis of the decision making strategy. More liberal data gathering techniques are recommended for exploratory research of this nature (Svenson, 1989), and in the case of complex, unstructured decision environments probing techniques may lead to data that is more amenable to analysis (Patel and Groen, 1986). Aided retroactive verbal analysis, particularly employing video taping and playback of problem solving sessions for the decision makers is a strong technique for research under these conditions (Waldron, 1985).

## H - METHOD

The research project generates three sets of data from eight subjects, four experts and four novices: (i) concurrent verbal protocols elicited during the performance of two software diagnosis problems, (ii) retroactive verbal protocols elicited during the review of recorded video tapes of the two software diagnosis problems and (iii) causal maps of the subjects' conceptual representation of the two software diagnosis problems elicited during open ended interview sessions.

During the experiment, an attempt was made to prevent the experimenter from knowing the skill level of the subjects. The skill level became apparent during the protocol sessions however, from comments that the subjects made that referred to their experience and by the age and demeanor of the subjects.

Knowledge of the skill level of the subjects presented no problems for the concurrent protocol sessions as the experimenter was basically a non participant observer. The potential for bias occurred during the coding of the protocols. The experimenter attempted to treat all subjects equally during the coding of the protocols but to help establish whether the coding was in fact done impartially, a second rater was used to code segments of the protocols and interrater reliabilities were calculated. The second rater did not know the purpose or objectives of the study nor did he know that the subjects for the experiment were grouped by skill level (see section C.3 of Chapter IV for a more detailed description of the interrater reliability tests and for the test results).

## H.1 Software Diagnosis Task Sessions

### H.1.a Experimental setting

The concurrent verbal protocol sessions for the two diagnosis problem solving tasks were completed during five consecutive days (one work week) at the subjects' place of work. A closed room was set aside for the sessions that was in the same general office space as the subjects' personal workstations. The room was locked during the problem solving sessions so that the subjects could not be disturbed and the windows were completely covered by drapes so that it was not possible to see in or see out of the room.

Along one wall of the room a table was set up and a computer terminal was placed on the table. The table provided adequate room for the terminal and considerable space for books, notes, writing paper etc. The terminal was the standard terminal used by the subjects in their job and was connected to the host system in the identical manner as their own terminals. A chair was placed in front of the terminal for the subjects to use during the problem solving sessions. Behind the chair on a fourty-five degree angle to the chair a camcorder was mounted on a tripod and the lens was directed towards the terminal screen. To the side of the camcorder, well behind the subject, a chair was placed for the experimenter to be seated during the problem solving sessions. It was imperative that during the problem solving sessions the experimenter be as unobtrusive as possible. It was n possible to conceal from the subjects that someone was observing and listening to them during the problem session however reasonable precautions were taken by the experimenter such as choosing a location that prevented non verbal communication from taking place, refraining from coughing or moving and from creating any other potentially interfering event.

### H.1.b Problem solving sessions

One week prior to the problem solving sessions the subjects who had been chosen for the experiment (see section D of Chapter III) were requested to sign up for two separate one hour sessions by entering their names on a schedule circulated by a department manager. First thing, on the morning of the first day of the problem solving sessions, the experimenter met briefly with each subject to verify the schedule. Schedule changes were found to be necessary because of unexpected business commitments of some of the subjects. The changes were made without difficulty. The experimenter took this opportunity to introduce himself to each subject, to thank them for their participation and to begin to develop a rapport with the subjects. After finalizing the schedule, the experimenter had time to verify that the equipment in the room had been set up properly and to prepare for the first session..

The procedure for the first problem solving session was identical for each subject. First the subject entered the room and was seated. The experimenter thanked the subject for his or her participation in the experiment and reinforced the fact that all results would be completely confidential. A code number was assigned to the subject and he or she was told that all reference to the participants in the study would be by code number to assure confidentiality. The subjects were told that they were participating in a study of problem solving behaviour and that they would be required to solve two software diagnosis problems and then to meet afterwards with the experimenter to discuss the problem solving sessions. The subjects were then told that in order for the experimenter to be able to better analyze the problem solving sessions, the experimenter would be videotaping the session. It was explained that it would be important for the experimenter to be able to refer back to the data on the terminal screen used during the session to help in the analysis.

The subjects were given a brief explanation of the verbal protocol analysis technique and were instructed in how to produce think-aloud verbal protocols. They were then asked to practice on a simple problem requiring them to perform a multiplication task (two digit number by two digit number). The subjects were given the multiplication task and were asked to think out loud while solving the problem. The subjects were allowed to ask questions while performing the multiplication task and received verbal encouragement from the experimenter. After having completed the first task, the subjects were given a second task, an anagram requiring them to unscramble seven letters to spell a commonly used word.

The practice sessions allowed the researcher to determine beforehand if a subject had a problem verbalizing his or her thoughts and if necessary, to provide additional practice for the subject to improve his or her verbalizations (Ericsson and Simon, 1980). In the extreme case a subject who could not express himself or herself verbally could be eliminated from the experiment. In this study all subjects were able to verbalize well after performing the two practice tasks.

After the practice session was terminated the subjects were given brief instructions depending on which of the two tasks they were to perform and were asked to commence the problem solving task. For both tasks they were told that the specific problem that they were to diagnose was an abnormal termination (abend) problem that was not documented in the company data base and that the memory dumps that they would be provided with were the actual dumps generated by the client system at the time of the abend. The subjects were told that additional dump information may or may not be available to them and that if they felt that they needed additional data during the problem solving session that they should ask for it by name. If a subject requested additional information, the experimenter would either give the material to the subject if it was available or respond that the information was not available. There was one additional report available for the 30A problem and no reports available for the OC4 problem.

To make the problem solving task as realistic as possible, the subjects were free to consult any manuals or documents that they might normally refer to during the performance of a diagnostic task. They were directed to get any materials, manuals or other documents that they felt they might normally refer to while diagnosing a software problem. They were then asked to commence the problem solving task using the terminal provided. The subjects were given the job control language statements to access the dump for the specific problem on the terminal. Once the correct screen had been accessed the video taping of the session commenced. During a problem solving session the subjects were allowed to leave the room to get other materials if needed. The tape was stopped upon their departure and recording was resumed when they sat down to continue the problem solving session.

A counterbalancing technique was employed, varying the order of the two tasks presented to the subjects. The counterbalancing technique controls for the effect of experimental bias that may occur as a result of the subjects' experience with the first problem solving session influencing their performance for diagnosing the second task (see the Introduction to Chapter IV for a detailed explanation of the counterbalancing technique).

During the performance of the task, the experimenter remained silent unless asked a specific question or to prompt the subjects if they remained silent for a long period of time. In the case of a prompt, the experimenter simply said "what are you thinking about now."

After completing the first task the subjects were requested to keep all information about the experiment confidential and not to discuss the experiment with anyone else, even if that person was not participating in the experiment. They were thanked for their participation and the time for their participation in the second session was confirmed.

When the subjects returned for the second problem solving session, they were given an opportunity to get any additional material that they wanted. They were then given the instructions for the problem task and requested to begin. The procedure during the second session was identical to the first session.

## H.2 Aided Retroactive Protocol Sessions

After all the subjects had concluded the two problem solving sessions they were asked to sign up for two one hour sessions during the following week. They were told that these sessions would be follow up sessions to discuss the two problem tasks that they had performed.

The aided retroactive protocols sessions were conducted as soon after the problem solving sessions as possible to limit the effects of forgetting, to minimize the opportunity for the subjects to discuss the problem solving sessions together, and to conclude the experiment while interest and motivation among the subjects was still high.

To start the retroactive protocol session the subjects returned to the same room as was used for their problem solving session. The subjects were asked to review video tape recordings of their problem solving sessions of the previous week in the same order that they performed the tasks and to describe the thoughts that they were having during the problem solving sessions. The experimenter made a point of asking the subjects not to explain what they were doing but rather to relate what they were thinking at the time.

The researcher verbally probed the subjects to vocalize during prolonged periods of silence or to expand upon their comments when the subjects' descriptions of their thought processes were vague or incomplete. A subject may have been asked why they were doing some activity in order for him or her to generate a causal description of the behaviour exhibited in the video tape. The subjects' verbalizations during the video tape review sessions were recorded on a cassette recorder for later analysis.

At the end of the video review session for each of the two problem solving tasks the subjects were engaged in an open-ended discussion to elicit their global level causal representations of the problem tasks. The subjects were first asked to define what the problem meant to them. For example, they were asked to describe what an "OC4 PIC 11" means to them. They were then asked to describe what might cause an OC4 PIC 11. Based on the subjects' responses the experimenter probed the subjects to clarify their comments or to expand upon their causal explanations.

It was decided to conduct the causal map interviews at the end of the experiment rather than at the beginning for two reasons. First, it was felt that giving the subject an opportunity to think about the problem domain in detail prior to the concurrent verbal protocol sessions may have influenced the spontaneity of the responses and therefore the quality of the data. If the causal map session was conducted immediately after the problem solving session and before the aided retroactive protocol session then it was felt that requiring the subjects to think in a structured manner about the first task may have a similar negative effect on the concurrent protocols verbalized in the second task. Second, the purpose of eliciting the causal maps was to make a comparison between experts and novices. It was felt that the subjects' previous experiences in the experiment would certainly influence the results of the causal explanations of the subjects but the complexity of representations between subjects would remain relatively the same. In fact, if there were influences in the results they may be to increase the complexity displayed by the novices more so than the experts rendering the comparison between them more conservative.

## I. SUMMARY

Chapter three introduced the research methodology employed in this study. A major contribution of this study is the use of a multi-method approach to test the hypotheses proposed.

The research method adopted was a controlled experiment that allowed the researcher to manipulate the independent variables, expertise and task difficulty in a controlled setting. Subjects of high and low skill level in software diagnosis were selected to perform two software diagnosis tasks that were similar in design to tasks that they typically perform in their natural work environment. One experimental task was chosen from a class of problems that frequently occurred in the real work environment and the other task was chosen from an infrequently occurring class of tasks.

Three sets of data are generated from the subjects: (i) concurrent verbal protocols elicited during the performance of two software diagnosis problems, (ii) retroactive verbal protocols elicited during the review of recorded video tapes of the subjects performing the two software diagnosis problems and (iii) causal maps of the subjects' conceptual representation of the two software diagnosis problems elicited during open ended interview sessions.

# CHAPTER IV

## DATA ANALYSIS

Chapter IV presents the results of the analyses performed on the data. The chapter is divided into five sections: section A presents the results for the assessment of the quality of the expert and novice representations of the problem domain, Section B reports on the tests comparing the decision accuracy and performance of the two skill groups for the two diagnosis tasks, section C describes the results of the tests for convergence in the use of information items within and between skill groups and reports on interrater reliabilities for the protocol analysis and section D presents the results of comparing the decision making strategies adopted by experts and novices and section E summarizes the findings reported in Chapter IV.

The basic research design employed in this study is a factorial design in which all combinations of two factors at each of two levels are tested in order to study their independent and interactive effects on a number of dependent variables (Kerlinger, 1973). The specific factorial model adopted is a two-factor experiment with repeated measures on one factor, that is each experimental unit is observed under all levels of the one factor (Winer, 1971).

The non repeated, independent categorical variable is skill level which groups the subjects into either an expert skill level group or a novice skill level group. For the specific dependent variable being tested (for example, decision accuracy), measures are taken for each of the subjects for both problem task types. That is, each individual subject is required to perform the "difficult-frequent" OC4 task and the "difficult-infrequent" 30A task and a measure of the dependent variable is taken for the performance of each task.

Conceptually, the two problem diagnosis tasks may be considered to be two repeated measure trials that along with the skill level factor allows for the test of the null hypotheses that there are no differences in performance (dependent variable) between skill levels (between group), between task types (within group) or in the interaction of skill level and task type.

Figure IV.1 presents the general two factor 2X2 repeated measure model used in the analysis. The subjects in the expert and novice categories are observed under the treatment combinations of task1 and task2. Thus the subjects are observed under all levels of the task factor in the experiment but only under one level of the skill factor (Winer, 1971).

The repeated measures design was chosen for two reasons. The first reason was to control for subject heterogeneity. The manipulations conducted in this experiment are intended to

**FIGURE IV.1**

MODEL OF TWO FACTOR ANALYSIS OF VARIANCE
WITH REPEATED MEASURES ON ONE FACTOR

| | TASK 1 | TASK 2 |
|---|---|---|
| EXPERT | EXPERTS 1-4 | EXPERTS 1-4 |
| NOVICE | NOVICES 1-4 | NOVICES 1-4 |

study the differences between experts and novices over two different tasks. To compare the performance of the two groups over the two tasks it is essential that the relative differences in the subjects' skill levels be maintained as closely as possible for the two tasks. In other words, the unique characteristics of the individual elements remain constant under the different treatments (Winer, 1971).

The repeated measures design provides for subjects of equal skill level under both task treatments, therefore reducing experimental error and helping to support the conclusion that any differences observed among the two task conditions is the result of the effects of the treatment alone.

The second reason that this approach was adopted reflects the limited availability of subjects to participate in the experiment. The use of repeated measures on one factor reduces the number of subjects required in a 2X2 repeated design by half. A consequence of repeating the tasks over the same subjects however, is that the precision of the estimators may be reduced (Winer, 1971).

The repeated measures test does have other drawbacks. The most serious drawback is that the performance of the first task may in some way systematically effect performance on the second task. To avoid the confounding of systematic "practice" effects with the treatment effects a counterbalancing technique was applied. The counterbalancing technique consisted of varying the order of presentation of the two tasks to the subjects. Two experts and two novices received the "difficult-frequent" task first and the other two experts and two novices received the "difficult-infrequent" task first. In the case where there are no carry over effects the repeated measure design may be considered to be similar to what has been called a split-plot design outside the behavioural sciences.

For all statistical tests performed in the study a .10 level of significance was adopted as a cutoff point to assess the significance of the results. The .10 cutoff was chosen because the objectives of this study are exploratory in nature and a restrictive level of significance would tend to suppress potentially interesting relationships.

## A - QUALITY OF REPRESENTATION

Five hypotheses compare the quality of expert and novice representations of the problem domain. To test for the first three hypotheses, hierarchical tree mappings of the subjects' causal assertions are used. Subjects are compared on the depth, breadth and size of their causal trees.

It should be noted that the causal maps are derived from single session interviews with the subjects and as such are not intended to be complete causal representations of the domain. Rather the objective of the single interview is to identify causal relations that come to the subjects' attention spontaneously or with little prodding by the researcher.

The purpose of limiting the interviews to one session was twofold. First, by conducting the interviews immediately after the aided retroactive protocol sessions it was felt that the subjects would be more likely to provide the causal assertions that they believed to be true of the domain. If they had time over multiple interviews to think about the diagnosis task they would have the opportunity to research the problem or to discuss it with their colleagues. Even if the subjects did not research the problem they would have time to reflect upon the problem and to fabricate their assertions in terms of what they thought they should say rather than what they believe to be true.

The second reason is that the role of the causal map in this case is not to develop a full and complete representation of a domain but rather to gather causal data on specific task representations for the purpose of comparing experts and novices. Schemata theory would suggest that experts have available in memory schemata of the domain that they can call upon to help solve a problem. Therefore, information of a causal nature should be more readily available to the expert and be of higher quality than that of the novice. This difference should manifest itself immediately within the first interview session, particularly since the nature of the causal relationships described at that time may be more at the global or higher level of cognitive organization.

Figure IV.2 provides an example of a causal map. It depicts the causal map of an expert for the more infrequent 30A task. The calculations for the breadth, depth and size of the tree are shown. Table IV.1 presents the data for the breadth, depth and size calculations for the subjects' causal tree representations.

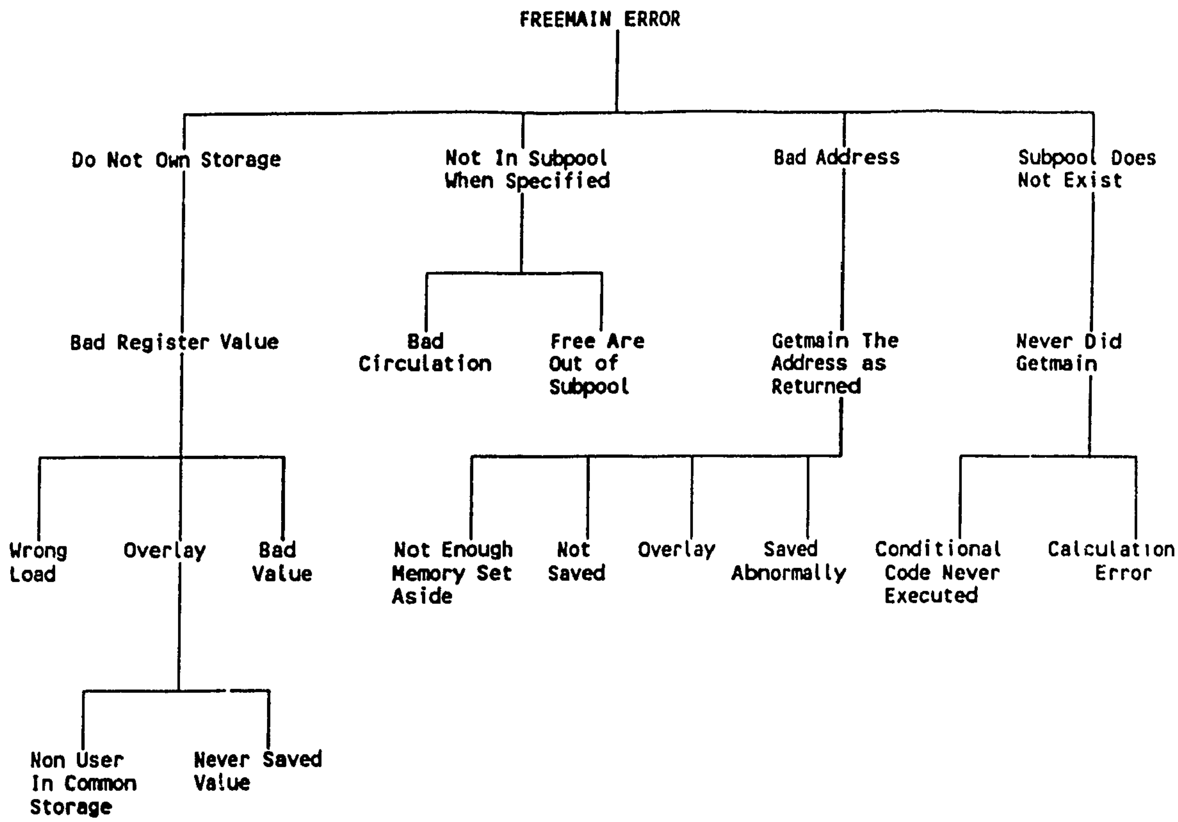
In order to test hypotheses h1.a, h1.b and h1.c a two factor analysis of variance with repeated measures on one factor was performed for each of the dependent variables; depth, breadth and size. Table IV.2 provides the results for the two factor repeated measure tests for the depth, breadth and size of the subjects' causal tree representations of the problem task.

For the assessment of depth, skill level ($F(1,6)=.329$, $p=.587$), task type ($F(1,6)=.193$, $p=.676$) and the interaction term ($F(1,6)=.114$, $p=.747$) were not significant at the .10 level of significance, although the data was in the predicted direction. Experts exhibited a greater depth in their tree representations than novices did for both the oc4 task (m.expert=28.75[10], m.novice=28.0) and the 30A task (m.expert=27.25, m.novice=21.56).

[10]m.expert is used to designate the mean value for the expert group and m.novice designates the mean value for the novice group.

**FIGURE IV.2**

AN EXAMPLE OF AN EXPERT'S CAUSAL MAP
FOR THE UNFAMILIAR 30A TASK WITH
BREADTH, DEPTH AND SIZE CALCULATIONS




DEPTH = (4*1)+(5*2)+(9*3)+(2*4) = 49

BREADTH = 4

SIZE = 21

**TABLE IV.1**

SUMMARY OF CAUSAL TREE MEASURES FOR BOTH SKILL LEVELS

| SUBJECT | TASK 1 BREADTH | TASK 1 DEPTH | TASK 1 SIZE | TASK 2 BREADTH | TASK 2 DEPTH | TASK 2 SIZE |
|---|---|---|---|---|---|---|
| EXPERT 1 | 3 | 15 | 11 | 4 | 30 | 13 |
| EXPERT 2 | 2 | 48 | 17 | 3 | 15 | 11 |
| EXPERT 3 | 4 | 36 | 16 | 4 | 49 | 22 |
| EXPERT 4 | 4 | 16 | 11 | 5 | 18 | 25 |
| MEAN | 3.25 | 28.75 | 13.75 | 4.0 | 28.0 | 17.75 |
| SD | 0.957 | 16.07 | 3.202 | 0.816 | 15.43 | 6.801 |
| NOVICE 1 | 4 | 35 | 16 | 3 | 3 | 3 |
| NOVICE 2 | 3 | 27 | 13 | 3 | 34 | 13 |
| NOVICE 3 | 2 | 27 | 11 | 2 | 38 | 11 |
| NOVICE 4 | 2 | 20 | 10 | 3 | 11 | 7 |
| MEAN | 2.75 | 27.25 | 12.0 | 2.75 | 21.50 | 8.5 |
| SD | 0.957 | 6.13 | 2.46 | 0.500 | 15.43 | 4.435 |

# TABLE IV.2

## TWO FACTOR REPEATED MEASURE RESULTS FOR CAUSAL TREE VARIABLES: DEPTH, BREADTH AND SIZE

### A. DEPTH

| SOURCE | SS | DF | MS | F | P |
|---|---|---|---|---|---|
| BETWEEN SUBJECTS | | | | | |
| SKILL | 64.000 | 1 | 64.000 | 0.329 | 0.587 |
| ERROR | 1168.750 | 6 | 194.792 | | |
| WITHIN SUBJECTS | | | | | |
| TASK | 42.250 | 1 | 42.250 | 0.193 | 0.676 |
| TASK*SKILL | 25.000 | 1 | 25.000 | 0.114 | 0.747 |
| ERROR | 1313.750 | 6 | 218.958 | | |
| TOTAL | 2613.750 | 15 | | | |

### B. BREADTH

| SOURCE | SS | DF | MS | F | P |
|---|---|---|---|---|---|
| BETWEEN SUBJECTS | | | | | |
| SKILL | 3.063 | 1 | 3.063 | 2.673 | 0.153 |
| ERROR | 6.875 | 6 | 1.146 | | |
| WITHIN SUBJECTS | | | | | |
| SOURCE | | | | | |
| TASK | 0.563 | 1 | 0.563 | 2.455 | 0.168 |
| TASK*SKILL | 0.563 | 1 | 0.563 | 2.455 | 0.168 |
| ERROR | 1.375 | 6 | 0.229 | | |
| TOTAL | 12.439 | 15 | | | |

### C. SIZE

| SOURCE | SS | DF | MS | F | P |
|---|---|---|---|---|---|
| BETWEEN SUBJECTS | | | | | |
| SKILL | 110.250 | 1 | 110.250 | 7.475 | 0.034 |
| ERROR | 88.500 | 6 | 14.7[illegible]0 | | |
| WITHIN SUBJECTS | | | | | |
| SOURCE | | | | | |
| TASK | 0.000 | 1 | 0.000 | 0.000 | 1.000 |
| TASK*SKILL$ | 64.000 | 1 | 64.000 | 2.385 | 0.173 |
| ERROR | 161.000 | 6 | 26.833 | | |
| TOTAL | 423.750 | 15 | | | |

For the assessment of breadth, the main effect for skill (F(1,6)=2.673, p=.153), task (F(1,6)=2.455, p=.168) and the interaction effect (F(1,6)=2.455, p=.168) were not significant at the .10 level of significance. The results however were in the direction hypothesized. Experts had a greater breadth score than novices for the OC4 task (m.expert=3.25, m.novice=2.75) and for the 30A task (m.experts=4.0, m.novices=2.75).

For the assessment of size or total number of causal nodes in the tree, the effect for skill level was significant (F(1,6)=7.45, p=.034) but task type was not significant (F(1,6)=0.0, p=1.0). This implies that experts (m=15.7) use significantly more causal references in their causal maps than novices (m=10.5). The interaction term was not significant (F(1,6)=2.385, p=.173).

In summary, hypothesis H1.c is supported by the resul of the repeated measures test. Hypothesis H1.a and H1.b are not supported, however, the mean depth and mean breadth of the experts' causal tree representation is greater than the mean depth and breadth of the novices causal tree representation. The failure of the depth and breadth results to test significantly at the .10 level of significance may be a consequence of the small sample size that is being used. Based on these results the hypotheses warrant further investigation.

Hypotheses h1.d and h1.e consider the quality of the experts' and novices' representation of the problem task as evidenced by their reference to underlying principles of the domain (deep knowledge) or to recognized patterns of information (surface knowledge).

To identify the subjects' references to deep and surface knowledge during the problem solving sessions the following procedure was used. First, all verbalizations coded as "conclusions" in the concurrent verbal protocol transcripts were identified. "Conclusions" were described in section G.1 of Chapter III as being statements of fact or conclusions that the problem solver has reached from the consideration of previous information items. "Conclusions" are therefore the result of the decision maker's consideration of previous information items.

The processing of these information items to reach conclusions may be of two types: it may be the result of applying the underlying principles of the domain or it may be the result of having perceived an information item which automatically brings to consciousness a conclusion that has been associated with that information item. This latter process is what has been referred to as pattern recognition or use of surface knowledge.

For each "conclusion" identified in the transcripts, the protocols prior to and including the conclusions were analyzed to determine whether the conclusions that were drawn were based on the subjects' use of underlying causal principles (deep knowledge) or were based on recognition of familiar patterns or associations between information items (surface knowledge).

After having coded all the "conclusions" appearing on a subject's transcripts, the researcher reviewed the aided retroactive protocol transcripts in which the subjects explained to the researcher what he or she was actually doing during the problem solving session. The researcher used these transcripts to gain a better understanding of the context in which the "conclusions" were drawn and to reevaluate the classification of each "conclusion" as deep or surface knowledge. The reevaluation process led to changing very few classifications and to the addition of relatively few new items to the two classifications.

Table IV.3 provides examples of protocols that were classified as being deep and surface knowledge references. Table IV.4 presents a summary of the total number of deep and surface knowledge references by subject for each of the two tasks. Table IV.5 presents the results of the statistical tests for comparing the subject's references to deep and surface knowledge.

The results of the 2 factor test with skill level as a group factor and the number of references to deep knowledge as the dependent variable indicate that the effect of skill level is significant ($F(1,6)=4.882$, $p=.069$) and that the effect of task type ($F(1,6)=.042$, $p=.844$) and the interaction effect ($F(1,6)=1.056$, $p=.334$) are non significant. Experts were

## TABLE IV.3

## REFERENCES TO SURFACE AND DEEP KNOWLEDGE

A. REFERENCES TO SURFACE KNOWLEDGE

| ID# | PROTOCOL | EXPLANATION |
|---|---|---|
| 1. E30V.15 | And in this case we have higher order byte on which strengthens the supposition that we did a branch and link register 14 at 673C to register 15 at F7E000. | Higher order byte is associated with branch and link without reference to a casual scheme. The presence of one is enough to link it with the other. |
| 2. N40V.61 | Now that's the CDE. That sounds like a program that would be used when you are running AMBLIST - the program you run to get a listing of a module. | CDE is linked to AMBLIST without reference to any underlying knowledge of the problem domain. |
| 3. N23V.57 | We are doing an SVCD - we are taking a DOG. This sort of looks like we were trying to free some piece of storage in the nucleus. | Direct link made between the SVCD and the storage in nucleus without reference to underlying knowledge of the problem domain. |

B. REFERENCE TO DEEP KNOWLEDGE

| | | |
|---|---|---|
| 1. E20V.31 | Because of the way this dump is taken. I mean its a slipdump - the trace table runs until it is captured which is not captured as early as it is in the case of a parameter check. | The observation of the slip dump is interpreted by assessing causal information about the domain that is stored in memory. |
| 2. N40V.46 | But, this is not necessarily a problem because RTM does issue SUCD to get a dump - so that's probably all that means. | The situation is interpreted by referencing causal knowledge about the source of SVCD. |
| 3. E43V.26 | Well it looks like a subpool number but that's reg15 and reg15 is not what contains a subpool. Reg15 really is not used. That's not what a subpool looks like. | Refers to underlying casual knowledge of the domain to interpret the validity of the subpool number. |

**TABLE IV.4**

QUANTITY OF DEEP AND SURFACE REFERENCES
DURING PROBLEM SOLVING SESSIONS

| | OC4 TASK1 | | 30A TASK2 | |
|---|---|---|---|---|
| | SURFACE | DEEP | SURFACE | DEEP |
| EXPERTS: | | | | |
| EXPERT 1 | 3 | 2 | 7 | 0 |
| EXPERT 2 | 11 | 4 | 13 | 2 |
| EXPERT 3 | 5 | 5 | 15 | 5 |
| EXPERT 4 | 6 | 5 | 8 | 7 |
| MEAN | 6.250 | 4.0 | 10.750 | 3.500 |
| SD | 3.403 | 1.414 | 3.862 | 3.109 |
| NOVICES: | | | | |
| NOVICE 1 | 14 | 0 | 8 | 3 |
| NOVICE 2 | 4 | 0 | 9 | 0 |
| NOVICE 3 | 2 | 1 | 3 | 1 |
| NOVICE 4 | 10 | 2 | 17 | 2 |
| MEAN | 7.500 | .750 | 9.250 | 1.500 |
| SD | 5.508 | .957 | 5.795 | 1.291 |

# TABLE IV.5

## RESULTS FOR COMPARING REFERENCES TO DEEP AND SURFACE KNOWLEDGE

UNIVARIATE REPEATED MEASURES ANALYSIS

DEEP KNOWLEDGE

| SOURCE | SS | DF | MS | F | P |
|---|---|---|---|---|---|
| BETWEEN SUBJECTS | | | | | |
| SKILL | 27.563 | 1 | 27.563 | 4.882 | 0.069 |
| ERROR | 33.875 | 6 | 5.646 | | |
| WITHIN SUBJECTS | | | | | |
| TASK | 0.063 | 1 | 0.063 | 0.042 | 0.844 |
| TASK*SKILL | 1.563 | 1 | 1.563 | 1.056 | 0.344 |
| ERROR | 8.875 | 6 | 1.479 | | |
| TOTAL | 71.939 | 15 | | | |

SURFACE KNOWLEDGE

| SOURCE | SS | DF | MS | F | P |
|---|---|---|---|---|---|
| BETWEEN SUBJECTS | | | | | |
| SKILL | 0.062 | 1 | 0.062 | 0.002 | 0.967 |
| ERROR | 200.375 | 6 | 33.396 | | |
| WITHIN SUBJECTS | | | | | |
| TASK | 39.063 | 1 | 39.063 | 3.307 | 0.119 |
| TASK*SKILL | 7.563 | 1 | 7.563 | 0.640 | 0.454 |
| ERROR | 70.875 | 6 | 11.813 | | |
| TOTAL | 317.938 | 15 | | | |

observed to make more references to deep knowledge for the oc4 task (m=4.00) than novices (m=.75) and for the 30A task (m.expert=.3.50, m.novice=1.50).

This result provides direct support for hypothesis H1.d that states that experts make reference to basic principles in the domain (deep knowledge) more often than novices during the process of solving problem tasks in the domain.

When the same 2 factor repeated measures test was performed with the number of surface references as the dependent variable, neither skill level ($F(1,6)=.002$, $p=.967$), task type ($F(1,6)=3.307$, $p=.119$) nor the interaction term ($F(1,6)=.640$, $p=.454$) were significant. This would tend not to support hypothesis H1.e that proposes that novices use pattern recognition during the process of solving problem tasks in the domain more than experts.

In summary, the results of the tests for the first set of hypotheses (H1.a through H1.e) that compared the quality of the representation of a problem task by experts and novices indicate that experts use more causal references (size of causal tree) than novices to represent the problem tasks and that experts refer to deep knowledge of the domain more often during problem solving sessions than novices. These results provide some support for the general proposition that experts have a higher quality representation of the domain than novices.

## B - DECISION MAKING OUTCOME PERFORMANCE

The second group of hypotheses concerns the decision making performance of experts and novices for diagnosing the two experimental problem tasks. Hypotheses H2.a, H2.b and H2.c propose that experts will be more accurate than novices for solving "difficult" tasks in the domain. Hypothesis H2.d proposes that experts will be more accurate in performing tasks that are more frequently found in the domain. Accuracy was scored according to the method described in section G.3 of Chapter III. Accuracy may take on values between 0.0 and 1.0 in .25 increments, with 1.0 designating a totally accurate diagnosis.

To test for hypothesis H2.a which states that experts will be more accurate than novices in diagnosing "difficult" problem tasks, a 2 factor test with repeated measures was performed. The dependent variable, accuracy of the problem solving outcome, was measured for the two tasks and grouped by skill level. Table IV.6 presents the performance scores for the subjects on the two tasks. Table IV.7 presents the results of the 2 factor tests.

The results of the 2 factor test indicate that the effect for skill level ($F(1,6)=17.308$, $p=.127$) and task type ($F(1,6)=7.723$, $p=.032$) were significant at the .01 level. The interaction effect ($F(1,6)=3.128$, $p=.006$) was not significant. The effect for skill level suggests that experts ($m=.656$)

**TABLE IV.6**

ACCURACY SCORES FOR EXPERTS AND NOVICES

| | TASK 1<br>OC4 | TASK 2<br>30A |
|---|---|---|
| EXPERTS: | | |
| EXPERT 1 | 1.000 | 0.250 |
| EXPERT 2 | 1.000 | 0.250 |
| EXPERT 3 | 1.000 | 0.250 |
| EXPERT 4 | 0.750 | 0.750 |
| M = | 0.938 | 0.375 |
| SD = | 0.125 | 0.250 |
| NOVICES: | | |
| NOVICE 1 | 0.000 | 0.250 |
| NOVICE 2 | 0.250 | 0.000 |
| NOVICE 3 | 0.750 | 0.250 |
| NOVICE 4 | 0.000 | 0.000 |
| M = | 0.250 | 0.125 |
| SD = | 0.354 | 0.144 |

# TABLE IV.7

## RESULTS FOR TWO FACTOR TEST WITH REPEATED MEASURES FOR DIAGNOSIS ACCURACY

**UNIVARIATE REPEATED MEASURES ANALYSIS**

**BETWEEN SUBJECTS**

| SOURCE | SS | DF | MS | F | P |
|---|---|---|---|---|---|
| SKILL | 0.879 | 1 | 0.879 | 17.308 | 0.006 |
| ERROR | 0.305 | 6 | 0.051 | | |

**WITHIN SUBJECTS**

| SOURCE | SS | DF | MS | F | P |
|---|---|---|---|---|---|
| TASK | 0.473 | 1 | 0.473 | 7.723 | 0.032 |
| TASK*SKILL | 0.191 | 1 | 0.191 | 3.128 | 0.127 |
| ERROR | 0.367 | 6 | 0.061 | | |
| | 2.215 | 15 | | | |

perform significantly better than novices (m=.188) for diagnosing the problem tasks in the domain. This result supports hypothesis H2.a which proposes that experts will be more accurate in diagnosing difficult tasks in the domain than novices.

The significant effect for task type suggests that both experts and novices perform better for the more frequent OC4 task (m=.594) than they do for the more infrequent 30A task (m=.250). This result supports hypothesis H2.d that proposes that experts perform more accurately for more frequent tasks (OC4) than they do for infrequent tasks (30A). Even though novices perform more accurately for the 30A task, it may be observed from the data that novices are uniformly poor in solving "difficult" tasks whether they are frequent or infrequent in the domain

Hypotheses H2.b and H2.c state that experts will exhibit superior performance compared to novices for the diagnosis of frequent tasks (H2.b) and for infrequent tasks (H2.c). The significant effects of skill and task together support both hypotheses. Experts (m=9.375) perform significantly better than novices (m=.250) for diagnosing the OC4 task and for diagnosing the 30A task (m.expert=.250, m.novice=.125).

In summary, hypotheses H2.a, H2.b, H2.c, and H2.d are supported by the results, suggesting that experts outperform novices for both task types and that the problem solvers' performance is more accurate for "difficult frequent" tasks than it is for "difficult infrequent" tasks.

Hypotheses H2.e, H2.f and H2.g consider the relative time that it takes for experts and novices to solve the two tasks. Hypotheses H2.e and H2.f propose that experts will take less time than novices to diagnose both difficult, frequent problem tasks and difficult infrequent problem tasks. Hypothesis H2.g states that experts will take less time to diagnose difficult, frequent tasks than they will to diagnose difficult, infrequent tasks. Table IV.8 summarizes the time taken by the subjects to perform each of the two tasks. Time was recorded in minutes and seconds and then rounded off to the nearest minute. The maximum time allotted for each task was 60 minutes. It should be noted that the measure of time may be a weak indicator of the quality of performance. For example, a subject may potentially terminate a problem solving session at any time during the sixty minutes if the subject believed that a dead end had been reached and that he or she could go no further. A problem solver who accurately solves a problem may theoretically take longer to perform the task than a subject who voluntarily terminates the session before the 60 minute time limit. The interpretation of the results between and within skill levels may be further hampered by the fact that a subject who solves the problem may take more time to do so than a subject who comes up with an inaccurate judgment. To control for this problem two tests for problem solving time

**TABLE IV.8**

TIME TAKEN BY EXPERTS AND NOVICES
TO COMPLETE DIAGNOSIS SESSIONS

| | TASK1 <u>OC4</u> | TASK2 <u>30A</u> |
|---|---|---|
| EXPERTS: | | |
| EXPERT 1 | 29 | 26 |
| EXPERT 2 | 38 | 39 |
| EXPERT 3 | 21 | 56 |
| EXPERT 4 | 24 | 36 |
| MEAN | 28 | 39.25 |
| SD | 7.439 | 12.473 |
| NOVICES: | | |
| NOVICE 1 | 60 | 60 |
| NOVICE 2 | 22 | 37 |
| NOVICE 3 | 17 | 30 |
| NOVICE 4 | 33 | 57 |
| M = | 33 | 46 |
| SD = | 19.201 | 14.765 |

TIME REPORTED IN MINUTES

are conducted. The first, tests whether decision accuracy is negatively related with the time taken to solve a problem. To the extent that the two are related, the comparisons of problem solving time between subjects may be undertaken with more confidence that task performance is being measured.

In testing the relationship between decision accuracy and the time taken to solve a problem care must be taken to control for spurious correlations. For example, if one task tends to require more time than the other but it is also more difficult, then the correlation observed between accuracy and time may be due to other factors relating to the differences between tasks.

To test for the relationship between accuracy and time, a random block design was used in which the eight subjects were considered to be blocks and the two tasks were treatments. The model used is specified as ACCURACY = CONSTANT + BLOCK + TASK + TIME. In this manner, the relationship between subjects for accuracy and time taken may be assessed controlling for effect of task.

Table IV.9 presents the results of the random block test. The effect of accuracy and time is not significant ($F(1,15)=.013$, $p=.914$) indicating that there is a not a significant relationship between accuracy and the time taken during a problem solving session after controlling for the effect of task.

# TABLE IV.9

RANDOM BLOCK TEST FOR DECISION ACCURACY
AND TIME TAKEN TO COMPLETE DIAGNOSIS SESSIONS

ANALYSIS OF VARIANCE

| SOURCE | SUM-OF-SQUARES | DF | MEAN-SQUARE | F-RATIO | P |
|---|---|---|---|---|---|
| BLOCK | 0.915 | 7 | 0.131 | 1.408 | 0.346 |
| TASK | 0.213 | 1 | 0.213 | 2.294 | 0.181 |
| TIME | 0.001 | 1 | 0.001 | 0.013 | 0.914 |
| ERROR | 0.557 | 6 | 0.093 | | |

The second test is a two factor repeated measure test to determine if there are differences between experts and novices for the total time taken for the problem solving sessions. The results of the 2 factor test with repeated measures on task type indicate that the main effects of skill level ($F(1,6)=.458$, $p=.524$), and the interaction effect ($F(1,6)=.031$, $p=.865$) are not significant. The result of the task effect is significant ($F(1,6)=6.408$, $p=.049$) indicating that problem solvers for both skill levels take significantly longer to perform the 30A task (m=42.625) than they do for the OC4 task (m=30.50). This result is consistent with the observation that the subjects were more accurate for the OC4 task than the 30A task. Table IV.10 summarizes the results of the 2 factor test.

Hypotheses H2.e and H2.f propose that experts will take less time than novices to solve each of the problem tasks. Although the skill level effect was not significant, experts were observed to require less time (m=28.0) to complete the OC4 task than the novices (m=33.0) and less time (m=39.250) than novices (m=46) to complete the 30A task.

Hypothesis H2.g proposes that experts will take less time to diagnose the difficult, frequent tasks than they will for the difficult, infrequent tasks. The results of the study indicate that experts took on average 28.00 minutes for the frequent OC4 task and 39.25 minutes for the infrequent 30A

## TABLE IV.10

### RESULTS FOR TWO FACTOR TEST WITH REPEATED MEASURES FOR THE TIME REQUIRED TO COMPLETE DIAGNOSIS SESSIONS

UNIVARIATE REPEATED MEASURES ANALYSIS

| SOURCE | SS | DF | MS | F | P |
|---|---|---|---|---|---|
| BETWEEN SUBJECTS EFFECTS | | | | | |
| SKILL | 138.063 | 1 | 138.063 | 0.458 | 0.524 |
| ERROR | 1809.375 | 6 | 301.563 | | |
| WITHIN SUBJECTS EFFECTS | | | | | |
| TASK | 588.063 | 1 | 588.063 | 6.048 | 0.049 |
| TASK*SKILL | 3.063 | 1 | 3.063 | 0.031 | 0.865 |
| ERROR | 583.375 | 6 | 97.229 | | |
| | 3121.939 | 15 | | | |

task. Given that the task effect is significant for the time tests, hypothesis H2.g is supported.

In summary, the results do not support the conclusion that time and accuracy are related but rather that they are both a result of differences in task. There is not a significant difference between skill levels for the time taken during the problem solving sessions (H2.e and H2.f). It should be noted however, that the results were in the direction predicted by hypotheses H2.e and H2.f. Hypothesis H2.g is directly supported by the test results.

## C - CONSENSUS FOR INFORMATION CUE USAGE

The test for the consensus of information item cue usage is divided into three sections. The first section describes the actual technique used to transcribe and code the concurrent verbal protocols of the problem solvers. The rationale for the choice of the actual code classification has been described previously. The second section describes the statistical tests used for hypotheses H3.a through H3.d and the results of the tests. The third section describes statistical tests used for hypotheses H3.e, H3.f and H3.g.

Hypotheses H3.a through H3.d consider the degree of consensus for information item use by the subjects during the performance of the problem solving tasks. Hypothesis H3.e and

H3.f consider the degree to which experts and novices use the key factors that have been identified as being pertinent for the solution of the problem tasks. Hypothesis H3.g proposes that diagnosis accuracy is related to the number of key items considered during a problem solving session.

## C.1 Verbal Protocol Analysis

Transcription of the verbal protocols took place in two steps. The first step consisted of playing back the audio portion of the video cassette recordings of the subjects' verbalizations and transcribing them by hand onto paper. Word processing documents of the hand written transcripts were then prepared and saved to disk.

Several difficulties in transcribing the data were encountered because of the subjects' numerous verbal references to addresses and registers that they were viewing on the computer terminal screens. The numbers were very difficult to interpret because of the similarity of the addresses and the poor enunciation of the subjects. For example, the following quotes are taken from the transcript of Novice2's verbalizations. It was impossible to make out the values of the addresses from the verbal recordings.

N23V.47[11] So it looks like we start at AC5DD3 - round it to the double word boundary. So we go to here to ACFDE0 plus 2C - so we have EOC.

N23V.69 It must be further down - Ok let's do a find 01BE7426. Not available. F 81BE7426 - Ok there we are.

To rectify this problem, the video tape recordings were played back a second time. This time the audio portion was used to review and to correct any mistakes in transcriptions that might have occurred. At the same time that the transcriber was listening to the audio portion of the tape he was watching the video portion which was showing the memory dumps that the subject was viewing at the time of the verbalizations. When a reference to the dump was made, numeric or otherwise, that was difficult to interpret, the video tape was put on pause and the actual values that were being referred to were searched for on the video taped image on the terminal screen. In this manner the ambiguous references were clarified and a final document of the transcription was prepared.

[11] The protocols of the subjects are identified according to the following method: the first two characters indicate the subject, for example N2 refers to novice 2; the third character refers to the problem task, 0=OC4, 3=30A; the fourth character refers to the type of protocol, V=video of problem solving session and the last character refers to the number sequence in the transcript.

The next step in the procedure was to code the verbal protocols. The verbalizations were broken down into small units of thoughts as described in section G.1.a of Chapter III. Verbalizations in the smaller units were then coded into categories. It was possible for one unit to contain verbalizations that could be subdivided and coded into more than one category.

The verbalizations of the subjects contained many references that were incomplete or too vague to code into one of the established categories. For example, the protocols described below were taken from the same subject that was quoted previously and are typical of ambiguous verbalizations.

| | |
|---|---|
| N23V.10 | Alright here we are. |
| N23V.42 | Ok this looks like an address. |
| N23V.45 | Ok. Select - sort of looks like the end of a word here. |
| N23V.68 | Ok. Isn't that interesting. Should be Ok, there is the address space. |

In each of these cases there is a reference to an information item that is being considered by the problem solver. The coder can not determine from the verbalization which information item is being referenced. Given this situation, the comparisons of information item use among the subjects is incomplete and may be misleading. To ignore these references or to assign them to categories would have negatively impacted on the quality of data.

To clarify ambiguous references, the video tape recording of the problem solving sessions was reviewed by the coder. The coder read the transcripts and simultaneously listened to the tape recorded verbalizations of the subjects. When an ambiguous verbalization was encountered, the coder paused the video recorder and consulted the image on the screen. If the ambiguous verbalization could not be immediately clarified the tape was reversed for a brief time and then played forward again. The coder followed the audio and video recordings of the events that led up to the ambiguous statement in order to provide a context for clarifying the verbalization. If this was not sufficient, the tape was played forward past the ambiguous comment to see whether subsequent events could clarify the reference. The process of "play back" and "play forward" was repeated until the comment was clarified. In very infrequent cases the coder could not clarify a comment with complete certainty. In this case, he used his knowledge of the dumps and his judgment to make a best estimate of the protocol content.

In this manner ambiguous references to the dump were eliminated. The transcripts were then revised to include the clarifications and the verbal protocol units were then coded.

## C.2 Consensus for Information Item Use

Hypotheses H3.a through H3.d consider the degree of consensus within skill groups and between skill groups for information item use during the two problem solving sessions. Information item use was defined as the consideration by the subject during the problem solving sessions of information items that exist in the memory dumps that have been issued by the system at the time of the system failure. The information items correspond to what has been designated as "cognitions" in the scoring of the protocols (see section G.1 of Chapter III).

The subjects' cognitions were used to construct an information item matrix for each of the two problem diagnosis tasks. The information item matrices provided the basis for the testing and analysis of convergence. The information item matrices for the OC4 diagnosis task and the 30A diagnosis task are presented in appendix II and appendix III respectively.

Two types of measures of convergence for the use of information items were considered. The first measure considered the number of information items used. These tests address the question of whether or not experts and novices use the same number of information items to diagnose a problem task. The second approach measured the degree to which consensus exists among the subjects for the use of each individual information item in the set of information items.

**C.2.a Tests for the quantity of information items used**

Hypotheses H3.c and H3.d suggest that novices will exhibit less within group consensus for information item use than experts for both diagnosis tasks. The first test administered to analyze the quantity of data used by the subjects was a two factor test with repeated measures on one factor. The subjects were grouped by skill level and the number of information items used during a problem solving session was measured over the two task types. This test provided a global comparison of the quantity of information considered by experts and novices.

Table IV.11 presents the number of information items used by the subjects for each task. Prior to performing the 2 factor repeated measure test a square root monotonic transformation was applied to the data (Winer, 1971). The results of the two factor repeated measure test are given in Table IV.12.

The two factor repeated measure test indicates that the main effects of skill ($F(1,6)=.002$, $p=.969$), task ($F(1,6)=.852$, $p=.392$) and the interaction term ($F(1,6)=.153$, $p=.709$) were not significant. The results provide no evidence of a significant difference in information item use between skill levels or across tasks.

**TABLE IV.11**

NUMBER OF INFORMATION ITEMS CONSIDERED BY EXPERTS AND NOVICES DURING THE TWO DIAGNOSIS TASKS

| | TASK1 OC4 | TASK2 30A |
|---|---|---|
| EXPERTS: | | |
| EXPERT 1 | 37 | 43 |
| EXPERT 2 | 45 | 31 |
| EXPERT 3 | 36 | 51 |
| EXPERT 4 | 33 | 38 |
| M = | 37.750 | 40.750 |
| SD = | 5.123 | 8.421 |
| SD SQ ROOT TRANSFORMATION | 0.409 | 0.662 |
| NOVICES: | | |
| NOVICE 1 | 74 | 55 |
| NOVICE 2 | 31 | 55 |
| NOVICE 3 | 23 | 24 |
| NOVICE 4 | 23 | 39 |
| M = | 37.750 | 43.250 |
| SD = | 24.459 | 14.886 |
| SD SQ ROOT TRANSFORMATION | 1.812 | 1.198 |

**TABLE IV.12**

RESULTS FOR TWO FACTOR TEST WITH REPEATED MEASURES
FOR THE NUMBER OF INFORMATION ITEMS CONSIDERED
DURING DIAGNOSIS SESSIONS

UNIVARIATE REPEATED MEASURES ANALYSIS

| SOURCE | SS | DF | MS | F | P |
|---|---|---|---|---|---|
| BETWEEN SUBJECTS | | | | | |
| SKILL | 0.003 | 1 | 0.003 | 0.002 | 0.969 |
| ERROR | 11.713 | 6 | 1.952 | | |
| WITHIN SUBJECTS | | | | | |
| TASK | 0.605 | 1 | 0.605 | 0.852 | 0.392 |
| TASK*SKILL | 0.109 | 1 | 0.109 | 0.153 | 0.709 |
| ERROR | 4.256 | 6 | 0.709 | | |
| TOTAL | 16.686 | 15 | | | |

In summary, there is no evidence of a pattern of consensus in the quantity of information items used between skill level groups nor within groups.

Given that the objective of the above test is to assess the level of convergence or agreement between experts and novices there is a disturbing element in the data depicted in the information item matrices which may account for thefailure of the data to support the convergence hypotheses. The problem arises from the large number of items that are considered by only one subject. That is, the consideration of these items is idiosyncratic.

The number of items that are idiosyncratic for the OC4 problem is 82 of the 134 total items recorded which is 61.2 percent of all items. For the 30A problem 68 of the 134 total or 50.74 percent of the total number of items are idiosyncratic.[12] The number of idiosyncratic items used is in its self an interesting result and will be discussed in chapter V with the discussion of key information item use.

Given that more than half the items for both tasks are idiosyncratic, it would be interesting to note whether there is a difference between experts and novices for the level of agreement for the non idiosyncratic information items. That

---

[12]It should be noted that the fact that the total number of information items for the OC4 task and the 30A task are equal is purely coincidental.

is, do experts agree among themselves for the use of information items more often than novices. This issue is of considerable importance. For example, responses for the following questions may be obtained by this analysis; is there a core set of information items that are used by all problem solvers to solve problem tasks in the domain? Are there differences in the use of these information items by experts and novices? Can these information items be identified and can we use the information to design better decision aids?

Table IV.13 presents frequency tables for each of the two experimental tasks indicating the number of times 4 experts agreed on an item, 4 novices agreed, 3 experts agreed, 3 novices agreed and so on for each possible level of agreement. Considering that the focus of this research study is on the level of "agreement" within and between skill groups, the frequency cells indicating how many times 3 or 4 subjects considered an item are of particular interest.

For the OC4 task, it may be observed that 21 times all four experts considered an information item whereas all four novices considered an information item only 4 times. If we include the times that three subjects within each group considered an information item, then experts are observed to agree 26 times compared to 10 for the novices. These results lead to the speculation that for the OC4 task experts tend to agree more often than novices.

## TABLE IV.13

FREQUENCY TABLE FOR INFORMATION ITEM AGREEMENT WITHIN SKILL LEVEL

| TASK | LEVEL OF AGREEMENT | | | | |
|---|---|---|---|---|---|
| | 4 | 3 | 2 | 1 | 0 |
| OC4 DIAGNOSIS TASK | | | | | |
| EXPERTS | 21 | 5 | 8 | 36 | 64 |
| NOVICES | 4 | 6 | 23 | 71 | 30 |
| 30A DIAGNOSIS TASK | | | | | |
| EXPERTS | 12 | 13 | 13 | 50 | 46 |
| NOVICES | 9 | 15 | 14 | 64 | 32 |

Observing the frequency table for the 30A task, all four experts agreed only 12 times. If the times that three experts agreed is included in the total, then the level of agreement goes up to 25. All four novices could agree only 9 times, rising to 24 when the times that three novices agreed is included. These results indicate that there is no real difference between experts and novices for the less frequent task.

Considering the results of the OC4 task alone, they would tend to support hypothesis H3.c. More generally however, H3a,H3.b and H3.d are still not supported. An interesting question that arises from the comparisons of the level of agreement is why there was a noticeable difference between the results for the OC4 task and the 30A problem. This result will be discussed in more detail in Chapter V.

### C.3 Key Information Item Use

Hypotheses H3.e and H3.f compare experts and novices for their use of key information items during the problem solving sessions. The procedure for identifying the key information items was dependent on the input of the instructional manager. The manager was a software diagnostic expert himself and had extensive experience in designing test problem cases to administer to students in the training programs. The manager was involved in the design of the two problem tasks used in the experiment and had spent considerable time analyzing the two tasks to determine the best solution paths.

In order to identify the key information items, the instructional expert was asked to solve each of the two tasks interactively at a terminal and to provide think aloud protocols during the problem solving sessions. The problem solving sessions were conducted in the same building as the experiment but in an area removed from the subjects and the test room. Several other conferences with the instructional expert were conducted at the training center which was located in another building several miles distant from the test site..

The manager's verbalizations were recorded while he was performing the task and later transcribed according to the procedures for transcribing and coding information described in the study. The transcripts were discussed with the manager to clarify any ambiguities and to verify with the expert that the solution path described by the transcripts was in fact consistent with his "ideal" solution path. A list of the information items that were coded as being "cognitions" was prepared for each of the two tasks. These information items were considered to be the key information items for solving the diagnosis tasks.

The instructor was not required to weight the information according to any particular scale of importance for solving the problems. The objective of hypothesis H3.e and H3.f is to test whether there is a set of key items that are used by the experts and novices during the problem solving sessions. Therefore the relative ranking of each item is not considered in this study. If the results support an association between the key items and the items used by the subjects then future research could test the question of whether there are differences in the weightings of experts, novices and the instructor for key information items and the relationship of the rankings to task performance.

Appendix IV and appendix V present the information item matrices for the OC4 and 30A problem tasks listing the information items that the instructor identified and the use of these items for each subjects.

Given that each key information item is of equal weight, one measure of comparison between experts and novices would simply be the total number of key items used by the subjects. To test for hypothesis H3.e that states that experts will consider more key information items than novices during a problem solving session, a two factor test with repeated measures on one factor was performed. The independent variable was the skill level of the subjects. The dependent variable was the number of key information items considered by the subjects and was measured for each of the two tasks. Table IV.14 presents a summary of the number of key items used by the subjects and Table IV.15 lists the results of the two factor test with repeated measures. Prior to performing the 2
Hypothesis H3.e is therefore supported. The effect of task

**TABLE IV.14**

NUMBER OF KEY INFORMATION ITEMS CONSIDERED
BY EXPERTS AND NOVICES DURING THE DIAGNOSIS TASKS

| | *TASK1 OC4* | *TASK2 30A* |
|---|---|---|
| ***EXPERTS:*** | | |
| *EXPERT 1* | *26* | *18* |
| *EXPERT 2* | *24* | *17* |
| *EXPERT 3* | *24* | *28* |
| *EXPERT 4* | *23* | *25* |
| *M =* | *24.250* | *22.000* |
| *SD =* | *1.258* | *5.350* |
| *SD SQ ROOT TRANSFORMATION* | *0.127* | *0.571* |
| ***NOVICES:*** | | |
| *NOVICE1* | *9* | *23* |
| *NOVICE2* | *18* | *25* |
| *NOVICE3* | *21* | *16* |
| *NOVICE4* | *8* | *9* |
| *M =* | *14.000* | *18.250* |
| *SD =* | *6.481* | *7.274* |
| *SD SQ ROOT TRANSFORMATION* | *0.879* | *0.908* |

## TABLE IV.15

### RESULTS FOR TWO FACTOR TEST WITH REPEATED MEASURES FOR KEY INFORMATION ITEMS CONSIDERED DURING THE DIAGNOSIS SESSIONS

UNIVARIATE REPEATED MEASURES ANALYSIS

| SOURCE | SS | DF | MS | F | P |
|---|---|---|---|---|---|
| BETWEEN SUBJECTS | | | | | |
| SKILL | 1.919 | 1 | 1.919 | 7.177 | 0.037 |
| ERROR | 1.605 | 6 | 0.267 | | |
| WITHIN SUBJECTS | | | | | |
| TASK | 0.380 | 1 | 0.380 | 0.974 | 0.362 |
| TASK*SKILL | 1.286 | 1 | 1.286 | 3.299 | 0.119 |
| ERROR | 2.339 | 6 | 0.390 | | |
| | 7.529 | 15 | | | |

type was not significant therefore failing to support hypothesis H3.f that suggests that experts and novices will use more key information items for solving the frequent OC4 problem than they would for the infrequent 30A.

Hypothesis H3.g states that the decision makers who use the greatest number of key information items during the problem solving session will have the most accurate problem solving outcomes. To test for this proposition a random block test was used in which the following model was specified: ACCURACY = CONSTANT + BLOCK + TASK + KEYITEMS. Table IV.16 presents the results of the repeated measures test.

The results indicate that there is a significant relationship between accuracy and the number of key items used during a problem solving session ($F(1,15)=6.650$, $p=.042$), suggesting that the greater the number of key items used for a problem solving session, the higher the accuracy of the outcome (coefficient=+.039). This result directly supports hypothesis H3.g.

## C.4 Interrater Agreement for Coding Verbal Protocols

To test for the reliability of the protocol coding two raters were used. The first rater who was also the principle researcher, coded the protocols for all the transcripts of the problem solving sessions into four categories: "intentions", "cognitions", "hypotheses" and "conclusions" using the method

**TABLE IV.16**

RANDOM BLOCK TEST FOR DECISION ACCURACY
AND KEY INFORMATION ITEMS USED DURING
DIAGNOSIS SESSIONS

ANALYSIS OF VARIANCE

| SOURCE | SUM-OF-SQUARES | DF | MEAN-SQUARE | F-RATIO | P |
|---|---|---|---|---|---|
| BLOCK | 0.487 | 7 | 0.070 | 1.575 | 0.298 |
| TASK | 0.573 | 1 | 0.573 | 12.974 | 0.011 |
| KEYITEM | 0.294 | 1 | 0.294 | 6.650 | 0.042 |
| ERROR | 0.265 | 6 | 0.044 | | |

described in section D.1 of Chapter IV. A second rater, who was a graduate Business Administration student, randomly selected the protocol transcripts for one expert and one novice and coded the protocols into the above four categories.

The second rater did not view the videos of the problem solving session during the coding of the protocols. This placed the coder at a disadvantage. There were ambiguous verbalizations in the transcripts as described in section D.1 of Chapter IV. The second coder was not familiar with the software diagnosis domain and it was felt that the videos of the problem solving sessions would be too technical for him to apply the information in the videos to the coding of protocols.

A rater without a technical knowledge of the software diagnosis task was chosen for two reasons. The first reason was that it was felt that if a coder was chosen from the company who was familiar with the task then he or she would be inclined to use their own experience to evaluate the protocols and to use their own value judgments in coding the protocols. An example of the potential for a rater to use their own value judgment interfere with the coding of protocols occurred during the pretest activities. During the pretest, an expert was given a partial list of think-aloud protocols for one decision session and asked to comment on what he thought the decision maker was doing at that time. After the expert stated his opinion, the researcher asked him how he came to that conclusion. He responded that given the addresses that the problem solver was looking at he must be searching for information on X because that is what he would be doing if he was looking at those same items.

A second reason was more practical, in that it was difficult to find a coder outside the organization who was familiar with the specific software diagnosis task environment. Given these circumstances, the results of the interrater agreement calculations should be conservative measures.

To measure interrater agreement Cohen's Kappa coefficient (Cohen, 1960) was calculated. Cohen's kappa is the most frequently used measure of interrater agreement for the coding of protocols (Todd and Benbasat, 1987). For the expert's transcript of the protocol session the kappa value was .612 ($p=.069$) and for the novice's transcript the kappa value was .604 ($p=.071$). Both of these values of kappa would indicate a good level of agreement beyond chance (Fleiss, 1981)[13]. Given that these values are considered to be conservative measures of agreement, the interrater agreement results may be interpreted to support the reliability of the codings.

[13]Values of Kappa greater than .75 or so may be taken to represent excellent agreement beyond chance, values below .40 or so may be taken to represent poor agreement beyond chance, and values between .40 and .75 may be taken to represent fair to good agreement beyond chance (Fleiss, 1981).

## D - STRATEGY FOR DIAGNOSING SOFTWARE PROBLEMS

The focus of the study with respect to strategy is to investigate how problem solving strategies influence the use of information items during a problem solving session. In particular, the processes of information item search and interpretation are considered.

### D.1 A Recommended Strategy for Software Diagnosis

In order to provide a context for better understanding the decision making strategies of the subjects, it is important to consider the influence that the host organization may have on the problem solvers' approach to diagnosing software problems. Each subject has completed an advanced training course offered by the organization that specializes in the operating system on which the two experimental problem tasks are based. The subjects spent considerable time during the course diagnosing software problems that were similar to the ones selected for this experiment.

The course instructor provided the students with a problem solving methodology for diagnosing software problems that represented the officially sanctioned approach of the organization. This general problem solving strategy is presented to students across a wide spectrum of courses regardless of the operating system or hardware focus. The approach was carefully followed by the class for each of the diagnosis problems solved during the course.

During interview sessions conducted with the instructional manager he described in detail the problem solving approach that is taught by the organization. The approach that he described may be considered to be a form of hypothetico-deductive strategy in which the problem solver attends to data cues in the environment which suggest hypotheses that are then tested against subsequent data for confirmation.

The strategy recommended by the organization divides the problem solving approach into two formal tasks. The first task is called "problem source identification (PSI)" and requires the problem solver to identify the program that is at fault. The second task is called "problem analysis" and requires the problem solver to interpret information that may be found in the dump to: (i) identify what activities were taking place at the time of the abend (where it took place), (ii) determine what the actual problem is that is causing the error and (iii) verify the causal explanation by selecting supporting data from the dump.

PSI involves a rather straight forward search for data in specified dump locations to find the program that is at fault. The challenge in the problem solving process is the second stage in which the problem solvers analyze the cause of the problem from the data given in the dump. To analyze the cause of the problem the instructor suggested that the first step would be to go to various parts of the dump and collect summary data that would provide general information on the status of the error. The instructor provided guidelines as to what parts of the dump would be most appropriate to search. The guidelines provided a generic approach that could be applied for all types of problems within a specific problem class.

Once adequate data was collected, the next task in the analysis would be to interpret the data and to hypothesize a cause of the fault. To elicit a hypothesis the problem solver could rely on his or her experience or he or she could consult various manuals and aids (microfiche for example).

While the data search at this stage requires the decision maker to consider standard dump reports, the procedure for the initial search procedure is not entirely without problem specific direction. The program at fault, identified in the PSI, tells the problem solver what the program was trying to do at the time of the error. This gives the problem solver some criteria for deciding which data is pertinent during the search. It does not, however, give problem solvers guidelines as to where to search for the items in the data base. The search sequence is a function of the interpretation of the data that is being presented to the problem solver during the search.

This distinction is important for understanding the problem solving strategy. The search sequence that was recommended by the instructor may be characterized as being data driven in the sense that the problem solver's interpretation of data items that have already been considered influences or directs the search for the next data item to be considered with the objective of sequentially gathering enough information to diagnose a cause of the problem . This is in contrast to an approach that may be described as being problem or goal directed in which knowledge of the problem type generates a causal hypothesis that organizes and directs the sequence of data search.

Once a hypothesis is generated, the final task in the diagnosis would be to search out data in the dump that confirms the hypothesis. The instructor would not consider the diagnosis complete if the last step is omitted even if the hypothesis elicited by the problem solver was correct. In the case where a hypothesis could not be substantiated the problem solver could backtrack to an earlier point in the analysis or could adopt an entirely different approach and begin to search anew for relevant data.

The search for and interpretation of data to elicit a hypothesis corresponds to the data directed search described by the hypothetico-deductive strategy. The data search sequence undertaken to confirm a hypothesis corresponds to problem or goal driven search. Therefore, the recommended procedure by the organization for solving diagnosis problems would be to use a combination of a data driven and a goal driven strategy.

Hypothesis H4.a suggests that both experts and novices use the same decision making strategy to diagnose problems in the domain. The expectation of the instructor would concur with this proposition. In addition, the instructor would suggest that the strategy used in both cases would be the formal approach taught in class, that is the hypothetico-deductive approach.

## D.2 An Observed Strategy for Diagnosing Software Problems

To identify and analyze the strategies used by the subjects during the problem solving sessions the concurrent protocols of the subjects were aggregated according to the procedure described in section G.5 of Chapter III. The aggregated transcripts of the subjects indicate that all subjects, regardless of problem task begin with a search sequence designed to review summary data in the dump. Expert2 and Novice1 comment on the initial search pattern in their protocols:

E2OV.1 At this point I'm going to start some very basic look arounds within this dump.

N1OV.1 The first thing I want to do is basically find out what the summary information of the problem is.

The initial search sequences adopted by the subjects provides an interesting result in that the search sequences vary dramatically by skill level. All four experts began both tasks by adopting a data search sequence that corresponds to the PSI step recommended by the instructor, that is they began to search for the program that was at fault. Only novice1 adopted this data search sequence (novice1 did so for both tasks). The three other novices went directly into the problem analysis stage for both tasks.

This result was somewhat surprising, given the response of the instructor to a question concerning the anticipated differences between experts and novices that was posed by the researcher during an interview session prior to start of the experiments:

> The more senior people can jump into the middle, I think. They tend to know just by the title what it is that is happening ... They might not go directly for the task unless they noticed something that made them stop and think.

The retroactive protocols support the suggestion that experts consider the title in the diagnosis of a problem. Expert4 comments about the dump title:

E43V.1 if there is a title it will show up and the title sometimes gives you a clue of what you were trying to do ... so its got a title ... so it looks like some kind of error in dump processing.

Nonetheless, they still chose to begin with the PSI.

The novices, on the other hand, failed to mention the dump title with one exception. Novice1 mentioned that the dump title might be important but then completely ignored it during the problem solving session. Responding to a probe about the dump title, Novice2 summed up the attitude of the novices in the retroactive protocol session:

> It all depends on what the title says ... depending on what it says there are some system generated titles ... I don't know, I didn't read it - I usually ignore it.

Therefore their skipping the PSI step could not be attributed to their having been influenced by the dump title.

A number of reasons may account for these observations. One plausible explanation is that the novices typically deal with very familiar problems[14] and they have adopted the habit of skipping the dump title and the PSI step. This habit is carried over into solving more unfamiliar types of problems.

---

[14]The less familiar problems are typically given to more experienced technical service representatives or are referred to them by the novices.

Experts on the other hand, by virtue of their experience, know that both the title and the PSI step may hold important information for unfamiliar tasks and therefore consider both carefully.

Once the problem solvers have begun the problem analysis process they typically spend a considerable amount of effort performing information searches before proposing their first hypothesis. On average (both skill levels combined) the first hypothesis is proposed by the problem solver in the 37th concurrent verbal protocol. The subjects averaged a total of 66 protocols for a problem solving session. When the results are considered by skill group, the experts mention the first hypothesis after an average of 31 protocols, compared to 43 for the novices.[15]

If prior to verbalizing a hypothesis, the subjects are involved in a lengthy information search from the onset of the problem solving process, the question arises as to what directs this search. The answer to this question would have important implications for understanding and aiding the decision making process. Two possibilities may be identified from the previous discussions: (i) a hypothetico-deductive approach would imply that the problem solver employs a data

[15] A 2 factor test with repeated measures on the task factor indicated that the skill effect ($F(1,6)=1.717, p=.238$), the task effect ($F(1,6)=.112, p=.749$) and the interaction effect ($F(1,6)=.001, p=.973$) were non significant thus providing no support for the proposition that experts generate hypotheses quicker than novices.

driven search leading up to a hypothesis; (ii) a pure goal directed search would imply that prior to beginning the search process the problem solver has an implicit goal or hypothesis in mind that organizes and directs the search.

The data driven approach suggests that the problem solver searches for data items in the dump that provide him or her with insights to the problem that may be used to guide the problem solving process. Insights may take the form of providing the problem solver with conclusions that are synthesized from prior data, it may suggest hypotheses or it may simply prescribe the next item of data to search for in the problem environment.

The second approach suggests that the process is purely goal directed. The problem solvers may use their knowledge of the domain to generate a hypothesis immediately at the onset of the problem and then direct his or her approach to data search and interpretation towards processing data that substantiates the hypothesis. A key factor influencing their choice of initial hypotheses would be the prior expectations of the problem solver based on his or her experience with past problems in the domain.

For the problem solving sessions observed in this study, both experts and novices were observed to adopt the data driven strategy indicated by the hypothetico-deductive approach. Evidence of this approach was provided by the subjects during the retroactive aided protocol sessions. Each subject was probed after viewing his or her initial information search sequence with a question intended to identify whether the search that they were conducting was a generally applied approach to data gathering or whether it was specific to the kind of problem that they were dealing with. An affirmative response to the first alternative was interpreted to mean that at least the initial search pattern was data directed and not goal directed. An affirmative response to the second alternative was taken to support the goal directed sequence.

All the subjects answered to the effect that this was a standard procedure that they used routinely and that their search was not designed with the specific problem task in mind. For example, the following responses were recorded:

| | |
|---|---|
| EXPERT1 | ... it all depends on the (problem solver). I usually start with this in order to get the job name and ... a TCB summary which shows you whether you had an abend or not. |
| EXPERT4 | Check the RTM2WA summary ... that's where I always check. If its there it usually gives you a good snapshot, an overall picture of what you are doing. |
| NOVICE1 | one of the first things we would do here in any type of situation is just find out whatever type of error it is, whether it is an OC4 or any type of error. |

Within the problem solving session there are several instances where experts and novices may be observed to be using non goal directed search patterns to identify data that may be helpful in diagnosing the problem error condition. The interpretation of these instances as data driven is more consistent with the protocols and the subjects' discussions found in the aided retroactive protocol sessions.

One type of search pattern that would imply that a goal did not exist to direct a search may be described as random search, that is a directionless search to find a data item that gives some insight as to how to proceed or to generate a hypothesis. Novice3 comments in the retroactive session:

> I have the impression that I did not know what I was doing at this point ... I am looking randomly at everything.

Later referring to another search sequence Novice3 states:

> Going in there as a grasp at a straw. I don't know why I even went in there.

Expert3 comments:

> What I'm going to try and do is search all the way through the dump - just trying to find a message."

Novice1 provides two different comments:

> I think I started off basically fishing. I was looking for any information that would sort of grab my attention.
>
> I started looking to see if there was any similarity between them. Not really knowing what I was looking for but just looking for something that might have clicked.

In the case where a search is not random, it has been suggested that it is data driven. Data driven search may take one of several forms. First, an information item may direct the problem solver to search for another information item that is related in some manner. In the coding of the concurrent verbal protocols this may take the form as a "cognition" followed by another "cognition" (a "cognition" was defined as an information item) or by a repetitive "intention-cognition" sequence. Expert4 provides an example of this sequence:

| | |
|---|---|
| E43V.8 | Heres the 30A - ok so before this guy we have a user abend (cognition). |
| E43V.9 | Let's go find out what a 30A is all about, 30A-18 (intention). |
| E43V.10 | Private area subpool not found (cognition). |
| E43V.11 | Ok now - dash 18 - see what address space and all that good stuff right here. Look at the task structure - alright there's the guy. Alright there's another DOG (cognition). |

(proceeds to search out data on the DOG)

A second type of data driven sequence occurs when a data item sequence[16] leads to a "conclusion" that is drawn from interpretation of the data sequence. Novice2 exhibits this pattern in the following protocol sequence:

| | |
|---|---|
| N23V.13 | Ok so here we are. So reg 15 is not good. 0 means subpool ... (cognition). |
| N23V.14 | Ok so 0 = 460 so that means we are trying to free 460 bytes in subpool 0 (cognition) |
| N23V.15 | Ok so reg 1 is any major value of getmain or the address of storage to be freed if the request is for a freemain - 0 if it is not (cognition). |
| N23V.16 | Ok so we have 7FFFE1A0 subtract the storage for 460 bytes. Seems awfully high(conclusion). |

The last comment regarding the address being awfully high is an important conclusion that has been derived from a series of data items.

A third type of data driven sequence occurs when a problem solver reaches a "conclusion", and the "conclusion"

---

[16]The term "data driven sequence" is used to denote a sequence of cognitions that are linked together in that each cognition is directed by the previous cognition. The sequence may be composed of one single cognition if the cognition neither directs the search for the next cognition nor is itself the result of being directed by a prior cognition.

then directs the next sequence of data search. Expert2 provides an example of a "conclusion" directing the data search:

| | |
|---|---|
| E20V.55 | And yes we did BALR at that point (conclusion). |
| E20V.56 | Now let's find out if we can pull out the BALR address (intention). |
| E20V.57 | Take a look - we did a load F off register 2 + 355 (cognition). |

The conclusion that a BALR occurred directs the next sequence of data search.

It should be noted that a data search sequence may not be directed by any prior sequence at all. For example if a data search sequence leads the problem solver to a dead end, he or she must then adopt a new direction to search for data in the dump. In the following protocol sequence Expert3 may be observed to reach a conclusion after a data search (not shown) that discounts a possible diagnosis of the problem. Expert3 now has to begin in another direction:

| | |
|---|---|
| E33V.46 | Just to be on the safe side - check my proc access files to check the dump. It really is a 2.2 dump. Nothing wrong with my logon proc (conclusion). |

Expert2 is a bit more direct:

E2OV.44 We have been looking here at a little garden path trip which is really our coming from the dumping of it which does not have anything to do with the actual failure at this point (conclusion).

And again later in the session commenting on the result of a search sequence:

E2OV.76 Interesting - back earlier we made a booboo - we looked in the wrong place (conclusion).

Another possible reason to abandon a search sequence is that the search sequence results in a data item or a conclusion that neither motivates the problem solver to explore this path in more detail nor allows the problem solver to draw any further conclusion. The data considered in these episodes may prove important later in the problem solving process if at that time they can be combined with other data to help provide insight to the problem. In the three protocols given below novice1 may be observed to be following one search sequence in protocol 54, switch to a second search sequence in protocol 55 and then switch again in protocol 56:

N13V.54 Ok I could use one more piece of information about the breakdown of the PSW. Ok I just wanted to pick up what the PSW was when we executed this freemain. Ok our PSW is that there 070C1000 (intention ---> cognition).

N13V.55 Ok now 32 is the address - is 32 the address - No (cognition).

N13V.56 Let me just go back to the abend 30A again. Return code keeps telling me that the private area subpool was not found (conclusion).

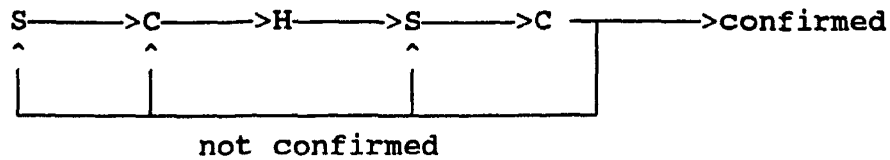
In summary, the data search sequences that the problem solver adopts prior to proposing his or her first hypothesis may be seen to be data driven. In the case where the search is not data driven it is likely the result of reaching a dead end in the search for a solution or the failure of the current data items being considered to generate further search. In this case the problem solver will typically embark on another data driven search. The results of the analysis fail to provide evidence that initial search sequences were goal driven.

Up until this point the data directed search patterns that exist prior to the problem solver's generation of his or her first hypothesis have been considered. To support the assertion that the problem solving strategy used by both the experts and the novices may be classified as a hypothetico-deductive approach it must be shown that the data driven search sequences lead to the generation of a hypothesis and that subsequent data search behaviour is directed by the hypothesis, that is by goal directed search.

Before addressing this issue it would be helpful to consider what a "text book" problem solving pattern would look like under the hypothetico-deductive approach. The pattern may be considered to be a sequence of events displaying the following ordering. First, a data search sequence is conducted which results in the problem solver drawing a conclusion concerning the error state. The conclusion drawn generates a hypothesis that proposes a cause of the error. The hypothesis initiates a goal driven data search sequence to verify the hypothesis that culminates in a conclusion that either accepts or rejects the hypothesis.

In the case of a rejection of the hypothesis the problem solver could: (i) attempt an alternate data search sequence to verify the hypothesis or (ii) could backtrack to a point in the data search sequence prior to the hypothesis generation and pursue a different solution path or (iii) the problem solver could begin a new data search that leads him or her in a new direction.

The "text book" hypothetico-deductive approach described above may be depicted graphically:




Where: S denotes a search sequence consisting of one or more related data items
C denotes a conclusion derived from prior data items
H denotes a hypothesis that makes an assertion concerning the cause of the error state

Five of the subjects (3 experts and 2 novices) displayed the "text book" sequence described above for at least one of the two problem solving sessions. Two of the remaining three subjects displayed the above sequence in a modified form. In both cases the modification amounted to omitting the intermediary conclusion and going directly from the data driven search stage to forming a hypothesis. A possible explanation for the omission may be that the subjects had drawn conclusions but did not verbalize them during the protocol sessions.

The other subject proceeded through the data search and conclusion stages leading to the generation of a hypothesis and began to verify the hypothesis. The verification stage ended abruptly with the subject being unable to draw any conclusion about the hypothesis.

Expert3 displays the hypothetico-deductive approach in the following protocol sequence. Only the most demonstrative statements in the sequence are given. In protocol 37 Expert3 draws the "conclusion" that there is something unusual with the load of register 15. A further data search is conducted to gain more information on register 15 to find a clear explanation for expert3's observation (protocols 38 to 41) leading to the "conclusion" that "I don't see any." Expert3 then states a clear "hypothesis" (which is in fact correct) describing the reason that the error occurred (protocols 44 and 46). Expert3 then engages in an elaborate goal driven data search directed at substantiating the stated "hypothesis" (protocols 47 to 53). After being satisfied that the data corroborates the "hypothesis", Expert3 restates the "hypothesis" as a "conclusion" (protocol 54).

| | |
|---|---|
| E3OV.37 | I'm still puzzled why we're loading register 15 not on a full boundary. |
| E3OV.42 | I don't see any. |
| E3OV.44 | If I assume for a minute that the code somehow got overlaid - the address where we loaded 15 wasn't quite right. |
| E3OV.46 | I postulate that really should have been 354 - one of the quite possible addresses. |
| E3OV.47 | Find 354 and we see that code is there. Lets see - register 2 is 8200 and we assume its plus 354 - then we check 8554. |
| EOV.48 | Anyways lets have a look - F7E0. |
| EOV.49 | Well its code for what its worth. |

| | |
|---|---|
| EOV.51 | Lets go back and check some more ... |
| EOV.53 | Lets try on the other side ... |
| EOV.54 | Ok so my guess is that somebody - the code got overlaid there where we loaded register 15 just before we branched. |

The above sequence lends support for the assertion that the problem solvers follow the hypothetico-deductive approach to solve the diagnosis problems presented to them. However there are problem solving sequences displayed in the protocols by both experts and novices that diverge from what might be expected from the hypothetico-deductive approach. For example, Expert1 did not generate a single "hypothesis" for the unfamiliar 30A task. Novice1 proposed a "hypothesis" without any prior data search sequence or "conclusion" that could be identified as having directed it and then disposes of the "hypothesis" without having first conducted a data search. Novice4 proposes five different "hypotheses", introducing new ones before reaching a conclusion on "hypotheses" that were still being considered.

What then distinguishes between these problem solvers and those that follow the hypothetico-deductive approach? One possibility might be that the differences in problem solving strategies between the subjects may be related to their level of expertise. Differences in problem solving strategies as a function of expertise have been documented in a number of studies (McDermott and Larkin, 1978; Patel and Groen, 1986; Simon and Simon, 1978). These studies have found that experts tend to use a forward reasoning strategy where novices employ backward reasoning. In contrast the studies that found expert subjects to use a hypothetico-deductive approach, found that novices tended to use the same global strategy (Elstein, Shulman and Spafka, 1978, Feltovitch et. al., 1984; Johnson et. al., 1981).

In order to determine whether there is a basis for concluding that the observable differences in the use of the hypothetico-deductive strategy are related to expertise, experts and novices were compared in their use of strategy over the two problem solving tasks. For the familiar OC4 task 4 experts were observed to use the hypothetico-deductive approach or a minor modification of the approach compared to 3 novices. For the 30A task, 3 experts and 2 novices use the hypothetico-deductive approach. In total of the 16 problem solving sessions, subjects in 12 sessions were observed to have used the hypothetico-deductive approach. These results are summarized in Table IV.17.

To test whether skill level of the problem solver has a significant effect with regard to the subjects' use of a hypothetico-deductive strategy a two factor repeated measure test was performed. To code the strategy variable for the test, a score of 1.0 was assigned if the subjects used the hypothetico-deductive strategy for a problem solving session

**TABLE IV.17**

ACCURACY SCORES AND STRATEGY USED BY
EXPERTS AND NOVICES FOR DIAGNOSIS SESSIONS

| | TASK 1 - OC4 | | TASK 2 - 30A | |
|---|---|---|---|---|
| | ACCURACY | HYPOTHETICO DEDUCTIVE | ACCURACY | HYPOTHETICO DEDUCTIVE |
| <u>EXPERTS:</u> | | | | |
| EXPERT 1 | 1.000 | YES | 0.250 | NO |
| EXPERT 2 | 1.000 | YES | 0.250 | YES |
| EXPERT 3 | 1.000 | YES | 0.250 | YES |
| EXPERT 4 | 0.750 | YES | 0.375 | YES |
| <u>NOVICES:</u> | | | | |
| NOVICE 1 | 0.000 | NO | 0.250 | NO |
| NOVICE 2 | 0.250 | YES | 0.000 | NO |
| NOVICE 3 | 0.750 | YES | 0.250 | YES |
| NOVICE 4 | 0.000 | YES | 0.000 | YES |

otherwise the score was 0.

The results of the repeated measure test indicate that the relationship between skill level and problem solving strategy is non significant ($F(1,6)=2.000$, $p=.207$). Therefore there is no support for the proposition that skill level influences the choice of problem solving strategy. The results of the repeated measure test are presented in Table IV.18.

Another possible factor that might contribute to the differences in strategy might be that subjects who accurately diagnosed the problem were the ones who used the hypothetico-deductive approach and that the subjects who were inaccurate in their diagnosis did not. Table IV.17 reports the accuracy scores of the subjects and indicates the strategy that they adopted. It may be observed that in the four cases where the subject did not use a form of the hypothetico-deductive strategy they also did not diagnose the problem accurately (accuracy score < .75). It may also be observed that all the subjects who did not use the hypothetico-deductive approach were not able to diagnose the problem accurately.

To test whether there is a relationship between problem solving performance (accuracy) and the subjects' use of a hypothetico-deductive strategy a random block test was performed specifying the model: ACCURACY = CONSTANT + BLOCK + TASK + STRATEGY. The results of the random block test indicate that there is not a significant relationship between accuracy and use of the hypothetico-

# TABLE IV.18

## TWO FACTOR REPEATED MEASURES RESULTS FOR PROBLEM SOLVING STRATEGY

UNIVARIATE REPEATED MEASURES ANALYSIS

| SOURCE | SS | DF | MS | F | P |
|---|---|---|---|---|---|
| **BETWEEN SUBJECTS** | | | | | |
| SKILL$ | 0.250 | 1 | 0.250 | 0.857 | 0.390 |
| ERROR | 1.750 | 6 | 0.292 | | |
| **WITHIN SUBJECTS** | | | | | |
| STRATEGY | 0.250 | 1 | 0.250 | 2.000 | 0.207 |
| STRATEGY *SKILL$ | 0.000 | 1 | 0.000 | 0.000 | 1.000 |
| ERROR | 0.750 | 6 | 0.125 | | |
| | 9.750 | 16 | | | |

**TABLE IV.19**

RANDOM BLOCK TEST FOR PROBLEM SOLVING ACCURACY
AND PROBLEM SOLVING STRATEGY

**ANALYSIS OF VARIANCE**

| SOURCE | SUM-OF-SQUARES | DF | MEAN-SQUARE | F-RATIO | P |
|---|---|---|---|---|---|
| BLOCK | 0.957 | 7 | 0.137 | 1.560 | 0.302 |
| TASK | 0.255 | 1 | 0.255 | 2.911 | 0.139 |
| STRATEGY | 0.033 | 1 | 0.033 | 0.371 | 0.565 |
| ERROR | 0.526 | 6 | 0.088 | | |

deductive strategy (F(1,15)=.371, p=.565). The results of the repeated measures test are given in Table IV.19.

From the qualitative observation that inaccurate problem solvers were less likely to display the hypothetico-deductive approach during a problem solving session the question may be posed as to whether this was a result of their choice to apply an alternative approach or their inability within the hypothetico-deductive approach to reach conclusions and to generate hypotheses. An analysis of the protocols would tend to support the latter explanation.

The first indication that the strategy used by the "inaccurate" problem solvers is motivated by the hypothetico-deductive approach is that all the subjects begin the problem solving sessions in a data driven mode. The subjects display data driven search sequences that clearly correspond to those described above. The inaccurate problem solvers however tend to reach dead ends more often than the accurate problem solvers or appear to be unable to draw conclusions from the data search sequences as readily. This results in a problem solving session that is characterized by many abrupt terminations in the flow of reasoning. In other words the subjects do not have the knowledge or ability to interpret data items sufficiently to carry them through the stages of the hypothetico-deductive approach. In graphical terms they do not proceed in an organized manner through the S--->C--->H--->S--->C path described previously.

To test this proposition the aggregated concurrent verbal protocols of the subjects were analyzed to identify problem solving sequences in the transcripts that were linked by directed search. For example, the first step in analyzing Expert1's problem solving sequence consisted of aggregating Expert1's concurrent verbal protocols into data driven search sequences. Table IV.20 displays the result of the aggregation procedure for Expert1.

The second step was to review the aggregated transcript and to identify chains of directed search patterns. For example, if a search sequence directed the problem solver to reach a "conclusion" which in turn directed him or her to a perform a second data search which in turn led to another "conclusion" then this sequence of events would be coded as follows: S1--->C1---S2--->C2. The lack of an arrow exiting from C2 indicates in the graph that the sequence terminated with C2 and no further search was directed by this sequence.

Table IV.21 presents the graphic representation of Expert1 for the familiar OC4 problem and the graphic representation of Novice4 for the same problem. The graph provides a useful tool with which to analyze problem solving performance. For example, a comparison of the number of sequences in a problem solver's transcript may be made: (i) between those subjects with an accurate diagnosis and those with an inaccurate diagnosis, (ii) between experts and novices or (iii) for the two problem tasks. The graph is also helpful

**TABLE IV.20**

AN EXAMPLE OF DATA DRIVEN SEARCH SEQUENCES FOR OC4 TASK DERIVED FROM THE AGGREGATED CONCURRENT PROTOCOLS OF EXPERT1

| CONCURRENT PROTOCOLS | PROBLEM SOLVING MODE |
|---|---|
| E1OV.1 - E1OV.5 | SEARCH 1 |
| E1OV.6 - E1OV.14 | SEARCH 2 |
| E1OV.15 - E1OV.20 | SEARCH 3 |
| E1OV.21 | CONCLUSION 1 |
| E1OV.22 - E1OV.26 | SEARCH 4 |
| E1OV.27 | HYPOTHESIS 1 |
| E1OV.28 - E1OV.32 | SEARCH 5 |
| E1OV.33 | CONCLUSION 2 |
| E1OV.34 - E1OV.42 | SEARCH 6 |
| E1OV.43 - E1OV.45 | SEARCH 7 |
| E1OV.46 | CONCLUSION 4 |
| TERMINATE SESSION | |

**TABLE IV.21**

A COMPARISON OF CHAINS OF DIRECTED SEARCH PATTERNS
EXHIBITED BY EXPERT1 AND NOVICE4 FOR THE OC4 TASK

EXPERT1 DIRECTED SEARCH PATTERN

1. S1
2. S2
3. S3-->C1-->S4-->S5-->C2-->S6-->C3-->H2-->C4 TERMINATE

NOVICE4 DIRECTED SEARCHPATTERN

1. S1
2. S2-->H1
3. S3-->C1,C2
4. S4-->C3
5. C4-->S5
6. S6-->C5
7. S7
8. S8-->C6-->S9-->C7,C8-->S10-->H2
9. S11-->H3
10. S12-->C9
11. S13-->H4-->S14-->C10.C11
12. S14 TERMINATE

---

S = SEARCH SEQUENCE
C = CONCLUSION
H = HYPOTHESIS

in assessing the strategies used by the problem solvers because it provides a summary of the events that transpired during the problem solving sessions. Table IV.22 summarizes the information in the graphs in tabular form.

Returning to the original question of whether the subjects with inaccurate diagnoses have significantly more interruptions and dead ends during problem solving sessions than those who diagnose the problems accurately, a comparison may be made of the number of problem solving sequences that accurate and inaccurate subjects use during the problem solving sessions.

To test the relationship between problem solving accuracy and the number of directed sequences that the problem solvers engaged in during the problem solving sessions a random block test was conducted specifying the model: ACCURACY = CONSTANT + BLOCK + TASK + DIRECT. The results of the random block test indicate that problem solving accuracy is significantly related to the number of directed sequences used by the problem solvers ($F(1,15)=11.406$, $p=.015$), thus supporting the suggestion that problem solvers who were inaccurate were less able to sustain directed data search sequences during problem solving sessions (coefficient= -.082). Table IV.23 presents the test results for the simple regression.

In summary, the results support hypothesis H4. Experts and novices use the same strategy for diagnosing software

## TABLE IV.22

SUMMARY OF THE CONTENTS OF DIRECTED SEARCH PATTERNS FOR EXPERTS AND NOVICES FOR THE DIAGNOSIS TASKS

| | SEARCH SEQUENCES | | CONCLUSIONS | | HYPOTHESES | | NUMBER OF DIRECTED SEQUENCES | |
|---|---|---|---|---|---|---|---|---|
| | OC4 | 30A | OC4 | 30A | OC4 | 30A | OC4 | 30A |
| EXPERTS | | | | | | | | |
| EXPERT1 | 6 | 3 | 4 | 4 | 2 | 0 | 3 | 6 |
| EXPERT2 | 14 | 11 | 14 | 7 | 4 | 3 | 6 | 8 |
| EXPERT3 | 11 | 20 | 7 | 14 | 7 | 2 | 7 | 13 |
| EXPERT4 | 9 | 12 | 5 | 14 | 4 | 1 | 4 | 3 |
| M = | 10 | 13 | 7.5 | 9.75 | 4.25 | 1.5 | 5 | 7.5 |
| SD = | 3.367 | 4.830 | 4.509 | 5.058 | 2.062 | 1.290 | 1.826 | 4.20 |
| NOVICES | | | | | | | | |
| NOVICE1 | 23 | 12 | 12 | 19 | 1 | 3 | 22 | 15 |
| NOVICE2 | 14 | 15 | 5 | 11 | 5 | 2 | 10 | 13 |
| NOVICE3 | 4 | 8 | 3 | 3 | 2 | 2 | 2 | 6 |
| NOVICE4 | 14 | 19 | 11 | 18 | 4 | 5 | 12 | 13 |
| M = | 13.75 | 13.5 | 7.75 | 12.75 | 3 | 3 | 11.5 | 11.75 |
| SD = | 7.762 | 4.655 | 4.425 | 7.411 | 1.826 | 1.414 | 8.23 | 3.95 |

**TABLE IV.23**

RANDOM BLOCK TEST RESULTS FOR PROBLEM SOLVING ACCURACY AND THE NUMBER OF DIRECTED SEQUENCES

ANALYSIS OF VARIANCE

| SOURCE | SUM-OF-SQUARES | DF | MEAN-SQUARE | F-RATIO | P |
|---|---|---|---|---|---|
| BLOCK | 0.514 | 7 | 0.073 | 2.287 | 0.166 |
| TASK | 0.188 | 1 | 0.188 | 5.872 | 0.052 |
| DIRECT | 0.366 | 1 | 0.366 | 11.406 | 0.015 |
| ERROR | 0.193 | 6 | 0.032 | | |

problems. Problem solvers who are inaccurate in diagnosing the cause of the software problem tend to have difficulty maintaining the directed search sequences that characterize the hypothetico-deductive approach. It would appear that the digression from the hypothetico-deductive approach is more a result of the inability to solve the diagnosis problem than intention to digress.

## E. SUMMARY

Table IV.24 presents a summary of the two factor tests with repeated measures on one factor for the three sets of hypotheses concerning: (i) the quality of the representation, (ii) diagnosis problem solving performance and (iii) information item consensus. Table IV.25 presents the hypotheses for which significant results were found for any of the statistical tests employed during the study.

Several significant differences between experts and novices were found during the study. These results support the following conclusions:

1. Experts exhibit more causal relations in their causal tree representations of the problem tasks than novices. Experts also make more references to deep knowledge during problem solving sessions. Together these results imply that experts

have a higher quality representation of the problem tasks than novices.

2. Experts are more accurate than novices for both the familiar and the unfamiliar tasks. On average both skill levels of problem solvers are more accurate for the familiar task than they are for the unfamiliar task.

3. Both experts and novices perform the familiar diagnosis task in less time than they do for the less familiar task. In general, subjects who took the least amount of time to complete the tasks were the most accurate in their diagnoses.

4. The number of times that 3 or 4 experts agree on the use of an information item is greater than the number of times 3 or 4 novices agree for the more familiar OC4 task. This observation does not hold for the less familiar 30A task.

5. Experts use more of the key information items deemed to be important for diagnosing the problem tasks than novices for both familiar and unfamiliar tasks. Problem solvers who use more of the key information items during a diagnosis session generate more accurate diagnoses.

6. Both experts and novices use the hypothetico-deductive strategy to diagnose both of the software problems. Problem solvers who used the hypothetico-deductive strategy were observed to generate more accurate diagnoses.

7. Experts were observed to use fewer directed information search sequences than novices for the diagnosis problem tasks. Subjects who used fewer information item sequences during the problem solving

sessions generated more accurate diagnoses.

# TABLE IV.24

## SUMMARY OF HYPOTHESES FOR WHICH SIGNIFICANT EFFECTS WERE FOUND

| HYPOTHESES AND MEASURES | EXPERT TASK 1 OC4 | EXPERT TASK 2 30A | NOVICE TASK 1 OC4 | NOVICE TASK 2 30A | SKILL | TASK | SKILLxTASK | INTERPRETATION |
|---|---|---|---|---|---|---|---|---|
| 1 QUALITY OF REPRESENTATION | | | | | | | | |
| H1 a DEPTH | 28 75 | 27 25 | 28 00 | 21 56 | F(1,6)= 329<br>p=.587 | F(1,6)=.193<br>p=.676 | F(1,6)= 114<br>p= 747 | THE FACT THAT THERE IS A SIGNIFICANT SKILL EFFECT FOR H1 c (SIZE) AND THAT THE DIFFERENCES BETWEEN EXPERTS AND NOVICES FOR DEPTH (H1.a) AND BREADTH (H1 b) OF CAUSAL TREE MEASURES ARE IN THE HYPOTHESIZED DIRECTION LENDS MODERATE SUPPORT FOR THE PROPOSITION THAT EXPERTS HAVE A HIGHER QUALITY REPRESENTATION OF THE PROBLEM TASKS THAN NOVICES |
| H1 b BREADTH | 3 25 | 4 00 | 2 75 | 2 75 | F(1,6)=2.673<br>p= 153 | F(1,6)=2 455<br>p= 168 | F(1,6)=2 455<br>p= 168 | |
| H1 c SIZE | 13 75 | 17 25 | 12 50 | 8 50 | F(1,6)=7.45<br>p=.034 | F(1,6)=0 0<br>p=1.0 | F(1,6)=2 385<br>p=.173 | |
| H1 d: DEEP KNOWLEDGE | 4.00 | 3.50 | 75 | 1 50 | F(1,6)=4.882<br>P=.069 | F(1,6)=.042<br>P= 844 | F(1,6)=1 056<br>P=.334 | EXPERTS MAKE MORE REFERENCES THAN NOVICES TO DEEP KNOWLEDGE FOR BOTH PROBLEM TASKS SUPPORTING THE PROPOSITION THAT EXPERTS HAVE A HIGHER QUALITY REPRESENTATION OF THE PROBLEM TASKS. THERE WAS NO SIGNIFICANT RESULTS FOR SURFACE KNOWLEDGE. |
| H1 e SURFACE KNOWLEDGE | 6 25 | 10 75 | 7 50 | 9 25 | F(1,6)=.002<br>P=.967 | F(1,6)=3.307<br>P= 640 | F(1,6)= 640<br>P= 454 | |
| 2 DECISION PERFORMANCE | | | | | | | | |
| H2 a ACCURACY | 9.375 | .250 | 250 | 125 | F(1,6)=17 308<br>P=.006 | F(1,6)=7.723<br>P=.032 | F(1,6)=3 128<br>P=.127 | HYPOTHESES H2.a, H2.b, H2.c, AND H2.d ARE SUPPORTED. EXPERTS ARE MORE ACCURATE THAN NOVICES FOR DIAGNOSING DIFFICULT TASKS (H2.a) WHETHER THEY ARE FREQUENT (H2.b) OR INFREQUENT (H2.c). EXPERTS ARE MORE ACCURATE FOR DIAGNOSING FREQUENT TASKS THAN THEY ARE FOR LESS FREQUENT TASKS (H2 d). |
| H2 b ACCURACY | | | | | | | | |
| H2 c ACCURACY | | | | | | | | |
| H2.d ACCURACY | | | | | | | | |
| H2 e TIME | 28.00 | 39.25 | 33 00 | 46 00 | F(1,6)=.458<br>P=.524 | F(1,6)=6.408<br>P=.049 | F(1,6)=.031<br>P=.865 | EXPERTS DID NOT TAKE SIGNIFICANTLY LESS TIME THAN NOVICES TO PERFORM THE FREQUENT TASK (H2.e) NOR THE INFREQUENT TASK (H2.f) ALTHOUGH THE RESULTS WERE IN THE HYPOTHESIZED DIRECTION. EXPERTS (WITHIN GROUP) DO TAKE SIGNIFICANTLY LESS TIME TO PERFORM THE FREQUENT TASK (H2.g). |
| H2.f TIME | | | | | | | | |
| H2 g TIME | | | | | | | | |
| 3 INFORMATION ITEM CONSENSUS | | | | | | | | |
| H3 a QUANTITY | 37 75 | 40 75 | 37 75 | 43 25 | F(1,6)= 002<br>P= 969 | F(1,6)=.852<br>P=.392 | F(1,6)= 153<br>P=.709 | THE RESULTS INDICATE THAT THERE IS NO EVIDENCE OF CONSENSUS BETWEEN EXPERTS AND NOVICES OR WITHIN SKILL LEVELS FOR THE QUANTITY OF INFORMATION ITEMS USED |
| H3 b QUANTITY | | | | | | | | |
| H3 c QUANTITY | | | | | | | | |
| H3 d QUANTITY | | | | | | | | |
| H3 e KEY ITEM | 24 250 | 22 00 | 14 00 | 20 750 | F(1,6)=7 677<br>P= 037 | F(1,6)=.974<br>P=.362 | F(1,6)=3 299<br>P= 119 | EXPERTS USE SIGNIFICANTLY MORE KEY INFORMATION ITEMS THAN NOVICES (H3 e) EXPERTS AND NOVICES DO NOT USE SIGNIFICANTLY MORE KEY INFORMATION ITEMS FOR FAMILIAR TASKS THAN FOR UNFAMILIAR TASKS ALTHOUGH THE RESULTS ARE IN THE HYPOTHESIZED DIRECTION |
| H3 f KEY ITEMS | | | | | | | | |
| H3 g KEY ITEMS ACCURACY | | | | | | | | |

## TABLE IV.25

## SUMMARY OF TESTS FOR WHICH SIGNIFICANT RESULTS WERE FOUND

| HYPOTHESIS | STATISTICAL TEST | RESULT | INTERPRETATION |
|---|---|---|---|
| H1.c (SIZE) | 2 FACTOR REPEATED MEASURE | F(1,6)=7.45<br>P=.034 | EXPERTS USE MORE CAUSAL RELATIONS TO REPRESENT THE PROBLEM TASK. |
| H1.d (DEEP KNOWLEDGE) | 2 FACTOR REPEATED MEASURE | SKILL F(1,6)=4.882<br>P=.069 | EXPERTS MAKE MORE REFERENCES THAN NOVICES TO DEEP KNOWLEDGE DURING PROBLEM SOLVING SESSIONS |
| H2.a (ACCURACY) | 2 FACTOR REPEATED MEASURE | SKILL F(1,6)=17.308<br>P=.006 | EXPERTS ARE MORE ACCURATE THAN NOVICES FOR DIAGNOSING FREQUENT AND INFREQUENT TASKS. |
| H2.b | | | EXPERTS ARE MORE ACCURATE DIAGNOSING FREQUENT TASKS THAN INFREQUENT TASKS |
| H2.c | | TASK F(1,6)=7.723<br>P=.032 | |
| H2.d | | | |
| H2.g (TIME) | 2 FACTOR REPEATED MEASURE | TASK F(1,6)=6.408<br>P=.049 | EXPERTS TAKE LESS TIME TO DIAGNOSE FREQUENT TASKS THAN TO DIAGNOSE INFREQUENT TASKS |
| H3.e (KEY ITEM CONSENUS) | 2 FACTOR REPEATED MEASURE | SKILL F(1,6)=7.177<br>P=.037 | EXPERTS USE SIGNIFICANTLY MORE KEY INFORMATION ITEMS THAN NOVICES DURING PROBLEM SOLVING SESSIONS |
| ACCURACY/ KEY INFORMATION ITEMS | RANDOM BLOCK TEST | KEYITEM F(1,15)=6.060<br>P=.042 | SUBJECTS WHO USED MORE KEY INFORMATION ITEMS WERE MORE ACCURATE IN THEIR DIAGNOSIS |
| ACCURACY/ NUMBER DIRECTED SEQUENCES | RANDOM BLOCK TEST | DIRECT F(1,15)=11.408<br>P=.016 | SUBJECTS WHO USED FEWER SEARCH SEQUENCES WERE MORE ACCURATE. |

# CHAPTER V

## INTERPRETATION AND DISCUSSION OF THE RESULTS

In this chapter the results of the study will be interpreted and discussed. The chapter is divided into five sections. The first section discusses the main findings of the study. The second section evaluates the multi-method approach adopted in the study. The third section discusses the implications of the results for the development of computerized decision aids. The fourth section discusses the limitations of the study and presents recommendations for future research. The final section provides a summary and conclusions.

### A - DISCUSSION OF THE RESEARCH RESULTS

The main finding of this study is that experts and novices both use a hypothetico-deductive problem solving strategy for software problem diagnosis. However, significant differences do exist between the two skill levels for the manner in which the hypothetico-deductive strategy is implemented to solve software diagnostic tasks. Experts exhibit: (i) more directed and more efficient information search sequences during problem solving sessions, (ii) refer more often to deep knowledge (iii) display greater consensus for the use of "key" information items and (iv) perform more accurately than novices.

It is suggested that the superior performance of experts is a function of their higher quality representation of the problem domain. In particular, the experts' procedural knowledge is found to be essential for the attainment of the above distinctions. In Section A the results that support these assertions will be discussed.

**A.1 Quality of Representation**

The quality of the subjects' representation of the problem tasks refers to the amount of relevant information that the problem solver has acquired with regard to the domain and the organization of that information in memory. The relevant information that is acquired may be classified as declarative or procedural knowledge.

The first set of hypotheses tested were based on a causal tree analysis and led to the conclusion that experts have a higher quality representation of the problem tasks than novices. This conclusion was based on the fact that there was convergence in the two tests of the quality of the problem solvers' representations of the problem tasks. Given only the results for the causal tree analysis, support for this conclusion would be much weaker considering that only the test for the size of the tree was significant (breadth and depth were non significant but in the right direction). However, when combined with the significant result of the test for the number of references to deep knowledge, the results as a whole are more convincing, thus highlighting the advantages of using multiple measures for the same construct.

In section G.2.1 of Chapter III it was suggested that the causal tree may be considered to be a hierarchical representation of the causal structure of the domain. That is, at the highest level of the tree, more abstract classificatory concepts are found. If these abstract classificatory concepts are considered to be states that are symptomatic of the general class of problems (for example, OC4 abend) then each level down a branch in the tree provides a description of the root level symptom at an increasingly detailed level of specificity where each link between nodes represents a cause-effect relationship.

Given that this description holds true, then the problem solvers' references to deep knowledge during the problem solving process should represent a reference to a sequence of linked nodes that are ordered by level in the tree. The information contained in these causal relationships may be considered to be declarative knowledge in that it represents the problem solvers beliefs about the structure of the problem domain. It is suggested that attached to these causal relationships is procedural knowledge that guides the problem solver through the problem solving process. The procedural knowledge directs the problem solver to search for information cues that help the problem solver to generate a hypothesis at the required level of specificity, that is, at the appropriate level in the causal tree. Depending on the level at which one begins the information search, the problem solver may seek to define the problem at a greater level of specificity by travelling down the tree (backward reasoning) or by moving up the tree to a higher level of abstraction (forward reasoning). Given a combination of factors such as the predisposition of the problem solver to a problem solving strategy, the characteristics of the task, time constraints, information format, risk etc. the problem solver may be inclined to start at the top of the tree (backward reasoning) or at the bottom (forward reasoning).

It would be expected that compared to novices, by virtue of their greater experience in the domain, experts would have a higher quality representation of the classes of problem tasks presented to them in the study. The results of the causal tree and the deep knowledge tests support this expectation for declarative knowledge. It would also be expected that the higher quality declarative knowledge would have associated with it superior quality procedural knowledge. If the declarative knowledge is in fact structured in a hierarchical tree representation then it would be anticipated that the procedural knowledge would lead the problem solver through a sequence of directed searches that would trace the path of the cause-effect relationships.

This is precisely what the results of the directed sequence analysis conducted in section D of Chapter IV indicate. Experts generate more directed search sequences that enable them to more effectively diagnose a problem.

Additional support for the proposition that experts have developed higher quality declarative and procedural knowledge that directs the problem solving sessions is found in the result that experts consider significantly more "key" information items during the problem solving sessions than novices. The key information items sought may be associated with the most fundamental underlying principles (cause-effect relationships) of the domain.

## A.2 Diagnosis Problem Solving Performance

The results for the tests of diagnosis accuracy supported the hypotheses that experts provide more accurate diagnoses than novices and that problem solvers of both skill levels would take more time to diagnose the unfamiliar 30A task than the familiar OC4 task.

The time taken during the task diagnosis sessions was not found to vary significantly by skill group. This finding requires interpretation in light of the findings concerning the analysis of the subjects' problem solving strategies.

There are two factors that may act to suppress real differences in time performance between skill levels. The first factor is the result of the experts having consumed time performing a problem solving activity that novices ignore. In analyzing the subjects' data search sequences it was observed that novices accepted the problem statement (for example OC4 An ABEND WITH SLIP DUMP GENERATED) as a given and then began to search for a diagnosis of the problem. Experts on the other hand did not take the reported problem as a given. They began the problem solving sessions with a data search sequence that was intended to verify that the problem reported was in fact present. Expert2 for example describes this process during the aided retroactive protocol session for the OC4 problem:

> The intent here is to ensure the facts that I am going to deduce and make decisions on are valid and match, then I have a higher probability making a correct decision ... In other words I could make the presumption that it is all valid and just jumped. By nature I don't do that ... There is nothing worse than going and finding out you just put 2 hours work on this problem and it doesn't have anything to do with the report at all.

Expert4 comments in the aided retroactive protocol session for the 30A task about the need to verify that the problem reported is in fact accurate:

> Let's look at this but lets be sure we don't consider this stuff as absolute truth ... could mean the information you have is not valid from the previous.

It is interesting to note that the instructional manager did not talk about verifying the given problem during any discussion, nor did he indicate that this stage was important. This might account for the reason that none of the novices performed a problem verification search. Experts may have learned with experience that problem verification may be worth the effort.

The second factor that may have occupied differential amounts of time between experts and novices is that the experts tended to spend time verifying that their hypotheses were in fact true. For example, three of the four experts participated in what at times (Expert4) was quite a lengthy search to verify their hypotheses for the OC4 task. None of the novices devoted any time to the verification stage.

This result is again rather interesting from the viewpoint of the organization's training program. The instructor did not consider a diagnosis to be complete unless it had been verified and consequently he stressed diagnosis verification in the training program. Yet none of the novices verified their hypotheses. A possible explanation of this result might be that similar to their lack of consideration of the dump titles discussed previously, the novices have developed habits that shortcut the recommended procedures. In their normal job environment where they typically deal with very familiar type problems these omissions may be rather benign, having little consequence for the diagnosis. Experts understand the benefits of these steps, particularly for less familiar problems.

In summary, the test results do not support the conclusion that experts take less time than novices to complete the diagnosis tasks although the results are in the direction postulated by the hypotheses. An analysis of the decision making strategies of the subjects suggest that the test results must be considered in the context of the problem solving process. Consideration must be given to the fact that experts spend time during the problem solving sessions performing problem solving activities that novices ignore and that these activities may enhance the problem solving performance of the experts.

### A.3 Information Item Consensus

The first measure of information item consensus considered the quantity of information items used. No significant effect was observed for the number of information items used by the subjects during a problem solving session. An interesting result, however, was the number of idiosyncratic items used by the subjects. The use of a large number of idiosyncratic items by both experts and novices has been observed in other studies. In a study by Grant and Marsden (1988) of medical students and expert clinicians the authors found that up to 60% of primary knowledge for all cases in their study was individualized. They conclude that "knowledge held in common is a minor component of primary knowledge for all groups and all cases."

The large number of idiosyncratic information items used by the subjects may be the result of a number of different factors. For example: it may be the result of a lack of domain specific knowledge, poorly organized knowledge, highly individualized approaches to the task, among other reasons. Whatever the reason, if experts are observed to perform more accurately than novices they would be expected to converge at some point on certain information items that are important for diagnosing the problem.

The results of this study indicate that such a convergence takes place. First, experts were observed to display more agreement (3 or 4 experts agreeing on an information item) than novices within group for information item use. Secondly, and more importantly, experts were observed to agree significantly more often for the key information items described by the instructional expert.

A possible explanation for this result might be that experts begin a search for information items according to their own habit and experience. At some point, however, their search leads them to consider an information item in their causal representation of the domain that is linked to the eventual correct diagnosis. At that point their search leads them along a search path that may correspond in structure to the causal maps described above. The search path directs them to the correct solution and in doing so the experts consider the key information items required to diagnose the problem. In this manner, each expert will consider many idiosyncratic items but will also consider the key items required to diagnose the problem accurately.

## A.4 Diagnosis Problem Solving Strategy

The results of the study strongly support the proposition that both experts and novices use a form of the hypothetico-deductive strategy to approach the diagnosis of software problems. There are however differences between experts and novices in the implementation of the strategy. The differences lie in the experts' superior ability in comparison to novices to: (i) direct the information search sequences that generate hypotheses for the diagnosis of the problem, (ii) to recognize the implications of the findings of a search for potential diagnosis solutions (generate hypotheses) and (iii) to direct the search to find relevant data that would verify the hypotheses.

Several observations in the study are notable concerning the directed search sequences leading to the generation of a hypothesis. The first and arguably the most important observation is that experts display a superior ability to engage in forward reasoning during the data driven search for a hypothesis.

The forward reasoning process is instrumental for diagnosing software problems because it directs the information search that culminates in the generation of a hypothesis. Forward reasoning may be considered to act as a cognitive filter, selecting certain information items from the problem environment and overlooking others. The key to effective data driven search is to identify and acknowledge those information items that may be interpreted by the problem solver's internal schemata structure to be representative of a known problem or to classify a new problem in the context of known problems.

A number of differences between experts and novices in their data driven search behaviour have been identified in this study. First, experts were observed to be more efficient in directing their searches for information. Experts were found to use fewer but longer directed search sequences, reach fewer dead ends and to consider more key information items.

It was proposed by the instructional manager prior to conducting the experimental problem solving sessions that the experts may likely skip the first step (PSI) described by the instructor in the company's recommended diagnosis approach if information presented to the experts in the dump title warranted it. In this study, experts were found to proceed with the PSI step even if they had considered the dump title. Novices, on the other hand, skipped the PSI step.

This result is contrary to what might be expected from discussions in the literature which suggest that as problem solvers become more experienced and more distanced from their training program (that is they become experts) they tend to modify their approach to problem solving by relying less on text book approaches and more on experience. Novices, it is suggested, being closer chronologically to the training program and less experienced, follow a text book approach (Grant and Marsden, 1988).

One possible reason for the contrary findings in this study may be that the novices had been accustomed to performing relatively routine tasks that they were very familiar with. For these kinds of tasks, skipping the PSI would have little consequence because of the high probability that the problem would be recognizable as an instance of a known problem. Consequently they formed a habit of going directly into the problem analysis.

Experts on the other hand may have learned with experience that skipping steps for unfamiliar problems may increase the possibility of following incorrect lines of reasoning and consequently lead to wasted time. The fact that the experts spent time first verifying that the reported problem was in fact the "real" problem supports this interpretation.

If this interpretation is correct then the implication that experts appear to go back to basics learned during their formal training in order to handle unfamiliar tasks is quite consequential for the design of decision aids.

There is evidence in the study that experts are better able to distinguish between competing hypotheses and to establish a directed search plan to validate hypotheses. There are instances, for example, where novices generate a number of hypotheses, including the correct diagnosis but are unable to formulate a directed search plan to follow through on the hypothesis. For example, in the aided retroactive protocol sessions for the 30A problem Novice1 describes a hypothesis that was being mentioned during the concurrent protocol session that was in fact the correct diagnosis solution:

> I thought or I was thinking at that point that maybe the SVCA was issued inadvertently where we had set the registers up like all the other SVC78's and we issued the wrong SVC ... That was just an idea at that point.

After stating the hypothesis, Novice1 abandons it to begin a search in another direction.

It would appear that novices have the ability to generate hypotheses from the data gathered during the directed search but that they lack internal schemata necessary to interpret the hypothesis and to generate problem verification sequences. That is, the cues available from the task environment do not instantiate the schemata necessary to evaluate the hypothesis as a potential solution alternative nor do they activate the procedural knowledge required to direct the data search to verify the hypothesis.

In the case where a diagnosis hypothesis was proposed by a problem solver with the conviction that the hypothesis was correct, both the expectation of the instructor and the hypothetico-deductive strategy would predict that the problem solvers would verify the hypotheses by conducting a further data search. Experts fulfilled these expectations but novices did not.

In summary, both experts and novices used a form of hypothetico-deductive strategy to diagnose software problems but experts did so in a more efficient and effective manner.

## A.5 Summary of the Research Findings

The results of the study confirm the findings that experts and novices both use the same hypothetico-deductive diagnosis strategy (Elstein, Shulman and Spafka, 1978; Feltovitch et. al., 1984; Glaser, 1984; Johnson et. al., 1981). This is contrary to the finding by others in the literature that experts use pure forward reasoning and that novices use backward reasoning (Patel and Groen, 1986).

One reason for the differences in these findings is likely the result of variation in the nature of the tasks studied. The software diagnosis tasks requires the problem solvers to gather data about a software problem in an incremental fashion with the possibility that the data found may at any time change the problem solvers conceptualization of what is happening. Under these conditions, Reimann and Chi (1989) suggest that decisions about what data ought to be considered are more effectively accomplished by considering alternative hypotheses in advance, lending support for a hypothetico-deductive approach.

A second major finding is that experts are more efficient and more effective than novices at applying the hypothetico-deductive approach to software diagnosis problems. This result was found to be a consequence of the experts' ability to engage in more directed forward reasoning processes leading to the generation of hypotheses. Compared to novices, the superior forward reasoning permits experts: (i) to use fewer data search sequences, (ii) to make more references to deep knowledge in the domain and (iii) to use more "key" information items. Experts are more effective in that they diagnose problems more accurately, use more "key" information items and verify their hypotheses with more regularity than novices.

It is suggested that the difference in performance observed between experts and novices is a function of the experts' superior representation of the problem. More specifically, experts have well formed schemata of the problem domain that contain declarative and procedural knowledge. The declarative knowledge is used by the experts to recognize information cues in the problem environment that are relevant for diagnosing a specific software problem and to direct information search. The information cues trigger other schemata which in turn have attached to them procedural knowledge that guides the problem solver in his or her search for further information items. This process of directed search is continued until sufficient information is gathered for the problem solver to combine the information in order to produce an internal picture of the problem that permits the problem solver to propose a hypothesis that provides a causal diagnosis of the software problem.

## B - EVALUATION OF THE MULTI-METHOD RESEARCH DESIGN

The multi-method approach that was adopted to investigate expert-novice differences for diagnosing software problems employed concurrent verbal protocols, aided retroactive verbal protocols and unstructured interviews. The combination of techniques provided a richness of data that could not have been created by using only one technique. The richness of data provided an understanding of the problem solving process of experts and novices that permitted the researcher to identify significant differences in diagnosis problem solving between the two skill levels and to better understand within skill group performance. The methodological approach did provide certain limitations as well. In this section the strengths and weaknesses of the methodology will be discussed.

## B.1 Methodological Strengths

A major strength of this research project is the use of real subjects performing real tasks in a realistic setting. Much of the research in problem solving and decision making process has relied on tasks that have been artificially constructed in part or whole. Given that a robust finding in problem solving research is that the problem solving process is influenced by the characteristics of the task, any artificiality introduced into the task may limit the generalizability of the results. Issues of task integrity are particularly important for research that involves an in depth analysis of decision making process.

A second strength of the study is that it introduced a causal tree methodology for assessing the quality of a problem solver's representation of a class of diagnosis problems that is grounded in theory. The methodology is amenable to both quantitative and qualitative analysis. The causal tree analysis that was performed was based on schema theory which suggests that a problem solver develops causal schemata to provide structure for interpreting problems and directing problem solving performance. The schemata are hypothesized to be hierarchically organized in memory moving from a level of high abstraction down to levels of increasing detail. The structure of the causal tree captures the problem solvers causal assertions concerning the problem and displays them in a hierarchical structure that parallels the hypothesized organization of schemata in memory.

Given that the causal tree structure is grounded in theory, it provides a measure that may be conceptually integrated with the other measures employed in the study. For example, the analysis of the subjects references to deep knowledge may be considered to be related to the causal relationships depicted in the causal tree. At the level of analysis in this study, one would expect that the quantitative measures of the tree would converge with the quantitative measures of the references to deep knowledge.

The causal tree representation may be developed to provide a much deeper analysis of the problem solvers' representations of a domain. At a more detailed level of analysis, more complete causal tree representations of the problem solvers could be developed and references to deep knowledge could be compared to the tree to establish whether the content of the deep knowledge references match the relationships depicted in the tree.

The analysis of the verbal protocols conducted in this study provided sequences of directed data search that was suggested to be a result of procedural knowledge that is attached to the problem solver's schemata of the domain. The results of this analysis were used to support the proposition that the problem solvers use a hypothetico-deductive strategy for problem solving. The causal tree structure may be used in conjunction with the verbal protocols to provide a qualitatively richer interpretation of problem solving strategy. The casual tree may be considered to represent a goal hierarchy where the task of analysis would be to confirm the state defined by the root of the tree, for example confirm that an OC4 PIC11 abend has occurred. To confirm this state, events at lower levels of analysis must be confirmed to support any conclusion. Relating the sequence of data search observed in the problem solver's protocols to the causal representation depicted in the causal tree, it may be possible to trace the sequence of nodes in the tree that are represented by the data search sequences. In this manner, it may be observed whether the problem solver is moving from lower levels of detail towards the root or goal state (forward reasoning) or traversing the tree from the top down (backward reasoning) or using a combination backward-forward strategy.

In summary, because the causal tree representation of the problem solver's casual relationships in the domain is grounded in schemata theory, the methodology provides significant opportunity to be combined with other techniques to benefit from a synergy that could not be realized by one technique alone.

Another strength of the methodology employed in this study is derived from the use of both concurrent and aided retroactive verbal protocols. The concurrent verbal protocols provide "pure" data in the sense that the data was generated at the time that the subject was engaged in diagnosing the software problem and that there was no probing by the experimenter nor obtrusive interaction with the subjects. For example, the identification of the information items used by the subjects during the problem solving sessions was based on concurrent verbal protocols to avoid introducing experimental bias in the subject's selection of information items. The aided retroactive protocols and the open ended interview sessions were conducted after the subjects had completed both problem solving sessions in order to gather data that permitted a more in depth interpretation and understanding of the concurrent protocols. The two methods of data gathering complemented each other to provide a richness of data that would be impossible using only one method.

Several examples of the synergistic effect of the two methods are evident in the study. For example, it was observed from the concurrent protocol analysis that all four experts began both tasks by adopting a data search sequence that corresponded to the PSI step described by the instructor while the novices jumped directly into the problem solving step. This was contrary to the expectations of the instructor who thought that the experts might jump into problem analysis based on information that was available to them in the dump title. During the retroactive protocol analysis the experts commented on the fact that they were looking at the dump title during the problem solving session "the title gives you a clue of what you are trying to do ..." while a novice commented "... I usually ignore it." From these comments it is clear that the experts consider the dump title but still execute the PSI step and that novices ignore the dump title and skip the PSI step. This led the researcher to suggest possible reasons for the unexpected behaviour which later helped support the important conclusion that experts tend to go back to basics for unfamiliar tasks.

Another advantage of using both protocol techniques is the ability for the aided retroactive protocols to help interpret the reasoning behind thought processes verbalized during the concurrent verbal protocol sessions or to provide evidence to confirm or reject explanations suggested by the researcher. For example, it was suggested in the study that the initial data search process exhibited by the subjects was data driven and not goal driven. Evidence for a data driven approach would be provided if protocol sequences could be found that showed the subjects engaged in a random search. From the concurrent protocols alone, the researcher would have to use his own unaided judgment to identify protocols that exhibited the desired characteristics. In the aided retroactive protocols the subjects provide several descriptions of their being engaged in random search: "going in there as a grasp at a straw," "...I am looking randomly at everything." The researcher can refer to these protocols to help in the interpretation of the problem solving process.

The concurrent verbal protocol sessions were videotaped, recording the subjects' verbal talk-aloud protocols and the images on the terminal screens used by the subjects to access the computer dumps during the problem solving sessions. The videotaping of the data provided several advantages over other techniques. First, from a methodological perspective the subjects were free to select information items during the problem solving sessions without any prompting nor involvement of the experimenter.

This is in contrast to the popular use of information boards or the use of problem scenarios to present a preselected finite set of information items to the subjects for consideration during the problem solving sessions. In the case of information boards, for example, information is hidden from the subject until the subject specifically asks for the information. To inform the subject of which information items are available, the subjects are presented with descriptive labels identifying the information items. In order to access the hidden information content, the subjects must specify the corresponding label. The subjects are free to choose as many of the specified information items as they wish during a problem solving session.

With advances in technology the presentation of information labels to subjects and the recording of information item use has become more sophisticated but the basic mechanism of specifying a label to retrieve information content remains essentially the same.

The problem with information boards and all methods that provide a finite set of information items to the subjects is that the labels of the information are often quite meaningful to decision makers. The labels alert the subjects to the potential importance of the information items and either confirm or disconfirm the subject's preconceived notions about the information (Ford et., al., 1989).

In the current study, the problem solving task was patterned after a real diagnosis problem task and the information content was presented in a typical problem solving format. That is, it was not a preselected subset of the total set of information available for the diagnosis problem. The subjects were free to select whatever information items they wished directly from the dumps without any interference from the experimenter or from the research design. Consequently the results relating to the information items considered by the subjects tend to be more reliable from this perspective.

A second advantage of the videotaped recordings of the problem solving sessions was that it provided a method for the researcher to record information that would not typically be verbalized. This is particularly important when the problem solving task requires the problem solver to refer to a tangible item such as a computer terminal during the sessions. Comments such as, "this address here" or " these instructions do not appear to be valid" would lose meaning if it were not possible to refer to the items that are being mentioned at the time.

A third advantage of videotaping the sessions is its strength as an aid for the subjects during the retroactive verbal protocol sessions. The subjects were not only able to listen to their verbalizations but were also able to see the terminal screen to provide them with the context for the verbalizations. The subjects were also able to freeze an image on the screen and to reverse the tape to review parts of the session as needed.

Within the study, multiple measures were implemented for each set of hypotheses. For example, to assess the quality of the subjects' problem representations both causal tree analysis and deep versus surface references were used. To assess problem solving performance, time and accuracy data was gathered. To assess information consensus three measures were used: (i) the number of information items used, (ii) a measure of within group agreement and (iii) the number of key information items used.

Multiple measures have a number of advantages. The first advantage is that it allows for a comparison of the results. This is particularly helpful in an exploratory study where standard measures may not exist and the effects proposed are not well documented in the literature. A second advantage of multiple measures is that it may permit the measurement of more than one component of a construct and provide an opportunity to gain insight into significant causal relationships underlying the construct being tested. For example, experts were more accurate than novices for diagnosing both tasks but they did not take less time to perform the decision tasks. The two differing results for performance would motivate the researcher to explain the different results, which in this case was found in the analysis of decision strategy.

## B.2 Weaknesses and Limitation of the methodology

The first weakness that must be mentioned is the reliance on a very small sample for the experimental results. The power of the statistical tests employed declines as the sample size is reduced. The question of small sample size for studies of decision making process has been well addressed in the literature (Ericsson and Simon, 1984; Kuippers and Kassirer, 1984; Payne, 1976). Kuippers and Kassirer (1987) summarize the issue:

> A methodology of discovery appropriate to undoubted complexity of human knowledge requires rich data about individuals rather than easily analyzed data about a population. Individual variation is a striking feature of human cognition that any attempt to average data across a population is certain to mask the true structure of the knowledge.

A second limitation described by Payne et. al. (1978) is the failure of protocol analysis studies to consider individual differences as a source of variance among problem solvers. For example, cognitive factors such as early or late closure, cognitive complexity and cognitive style may influence the problem solving performance and strategies chosen by the problem solvers.

A third limitation revolves around the principle researcher's role in participating in the data collection and in the coding and analysis of protocols. It is generally not recommended that the main researcher participate in the coding of the protocols in order to reduce researcher bias (Todd and Benbasat, 1987). In research studies involving protocol analysis, the principle researcher is often the most qualified to code the protocols since he or she has a detailed understanding of both the coding techniques and the problem domain. It was felt that this was the case for the study. To help control for researcher bias a second rater coded the protocols of an expert and a novice who were chosen at random and interrater reliabilities were calculated.

## C - IMPLICATIONS FOR COMPUTER AIDED DECISION SUPPORT

It has been emphasized throughout this study that the role of schemata in memory is to organize knowledge in a manner that provides structure for the understanding of problems in the environment and for guidance towards the solution of problems. Kotteman and Remus (1987) suggest that one, if not the primary, goal of DSS is to bring structure to ill-structured decisions. It would follow from these two propositions that there may exist a significant opportunity to improve the usefulness of a DSS by having it help bring structure to an ill-structured problem in a manner that is cognitively compatible with the schemata structures of the problem solver.

The results of this study provide several insights for the research and development of "cognitively compatible" decision support systems. The implications of the results for computer aided decision making are discussed in this section.

## C.1 Rationalizing DSS Over Several Decision Makers

Both experts and novices have been shown to use a form of hypothetico-deductive reasoning for solving software diagnosis tasks. The implications of a consistent strategy over problem solvers is promising for the development of decision support systems. It permits the developers of DSS to commit significant resources to the research and design of models and interfaces that capture the basic strategic approach of the hypothetico-deductive strategy and to rationalize their investment over several decision makers. The goal of research in this case would be to develop a basic design in the hypothetico-deductive mold that could be adapted as required to meet the needs of individual decision makers and tasks.

The DSS approach to decision modelling could break down the decision process into three phases: (i) a directed search to interpret the problem (forward reasoning), (ii) the generation of problem solving hypotheses and (iii) the verification of hypotheses (backward reasoning). The DSS could support any or all of these phases. This would imply that research could be directed to each of the three stages or to the integration of the three into a complete support system. Thus the application of the hypothetico-deductive approach to DSS research could serve to focus the research effort on process components and to begin a stream of organized research to build upon and to expand the research results.

For example, a knowledge based system could be developed that specialized in helping a problem solver perform the verification of hypotheses that have been generated. A hypothesis could be entered as a goal in the system and the system would perform the backward chaining to verify the hypothesis. The explanation facility of the knowledge base could provide insights as to why the hypothesis was confirmed or rejected. In this manner, the problem solver could concentrate on the creative aspect of hypothesis generation and the knowledge based system could perform the technical procedures required to verify a hypothesis.

It should be noted that the hypothetico-deductive model may not be applicable for all problem solving tasks. For example, Patel and Groen (1986) have demonstrated that for certain tasks experts partake in a pure forward reasoning strategy. It would first be necessary to determine the strategy used by problem solvers and then adapt the development of the DSS to the situation. Protocol analysis, both concurrent and retroactive, could play a key role in DSS development by identifying the strategic approach of the decision maker to problem solving.

A second major finding that would help to rationalize the research and development costs of a DSS over several decision makers is the fact that expert decision makers were found to use more "key" items than novices and that the use of these "key" items was correlated with diagnosis accuracy. An important concern of DSS developers is the ability to justify the cost of the DSS. Part of this problem stems from the perception that each DSS must be tailored to the idiosyncracies of the decision maker. If it can be demonstrated that there is a set of key information items that are important for accurate problem solving then this key set may form the heart of the DSS data base and provide guidelines to direct the data search. Only minor adjustments may be needed for an individual decision maker thus permitting the system to be adapted to a number of decision makers who perform similar functions.

Research to establish multiple methods to construct and validate the set of key information items for a problem solving task is required. In this study, the common approach of using expert opinion was used to generate the list of key variables. However, the method used to extract the key information items in the study warrants consideration for future research. In this study, the expert was requested to perform each of the software diagnosis tasks and to generate think-aloud protocols during the problem solving sessions. The protocols were then coded according to the coding scheme described in the study. The expert's protocols that were coded as "cognitions" were taken to be key items.

Adopting this approach to elicit the key variables has the advantage of permitting the researcher to identify which information items are actually used by the expert during a problem solving session rather than gathering a list of items that are hypothesized to be of importance. Depending on the needs of the researcher, either approach may be deemed to be more appropriate.

It should be mentioned that in this study the expert was in a unique position to provide the set of key information items because of his involvement in generating the decision tasks, his knowledge of the correct diagnosis and his position as an instructor. In general, the task of generating key information items in a problem solving domain has proven to be extremely difficult. An approach that might be more generalizable to the field would be to elicit both a causal map and concurrent protocols from selected subjects and to identify key information items from the perspective of how they fit into the problem solvers' representations of the domain and the problem solving strategies that they adopt.

## C.2 Adapting DSS to the Skill Level of the Decision Maker

While experts and novices may use the same problem solving strategy to diagnose software problems, it has been demonstrated that there are significant differences in performance between skill levels with respect to the execution of that strategy. It has also been shown that there are significant differences within skill levels for different diagnosis tasks. This would imply that even though the general approach of the DSS may be patterned on one problem solving strategy, the role of the DSS may vary by the skill level of the decision maker. For example, a DSS designed to support a decision maker for a problem for which he or she is considered to be an expert may be very different than a DSS to support a less skilled decision maker. The implications of the results of the study for DSS development for each skill group is presented in this section.

**C.2.a DSS for highly skilled decision makers**

For the OC4 more familiar task experts were found to engage in a more directed search process and to use more "key" information items than novices. Given that the expert decision maker has developed with experience well formed declarative and procedural knowledge structures in memory, the role of a DSS would be to provide the decision maker with the information items that are critical to the decision task in an efficient and effective manner. Once provided with the critical information items, the decision maker will be able to instantiate the schemata in memory required to effectively deal with the problem to reach a solution.

If the decision makers have a well formed representation of the problem in memory then it may be argued that they would be able to solve the problem without a decision aid, therefore rendering a DSS under these circumstances irrelevant. The results of the study indicate that there is a role to play for DSS in this situation.

One opportunity for a DSS is to make the decision making process more efficient. For example, although the experts agreed closely with respect to the use of "key" information items for the OC4 task, they were observed to consider a large number of idiosyncratic information items as well. By providing the decision maker with important information cues in a logical sequence, the decision maker may be able to reduce the number of information items considered and to generate problem solving hypotheses earlier in the decision making process, reducing the time and effort required to reach a decision. In addition, the reduction of irrelevant information cues may lead to an overall increase in problem solving accuracy by eliminating information overload and ambiguous stimuli (Gaeth and Shanteau, 1984).

For example, an intelligent checklist could be used in a knowledge based system to select relevant information items in a sequence that has been designed to elicit from the decision maker appropriate procedural knowledge to generate a problem solving hypothesis.

A second potential for a DSS under these circumstances is to ensure that all the steps in the hypothetico-deductive strategy be adhered to during the decision making process. For the OC4 problem, although all the experts generated the correct diagnosis hypothesis they did not all verify the hypothesis. Hypothesis verification was considered to be an essential part of the problem solving process by the instructional manager. A DSS could be designed that requires a decision maker to perform each problem solving step in a logical sequence. In this manner, the DSS may impose normative principles of the domain on a decision maker in a non obtrusive manner.

### C.2.b DSS for less skilled decision makers

In the case where the decision maker is less skilled in a domain he or she is less familiar with the decision task and has not had sufficient experience to determine which information items are relevant for the task nor how to combine the cues together to direct a problem solving search. The role of the DSS for less skilled decision makers is to provide critical information items and to impose a structure for the information that permits the decision maker to create new problem solving procedures to solve the problem. In other words, the DSS must provide the key information cues, help the decision maker to interpret the information and then provide the procedural knowledge to guide the problem solving search.

A critical component of the DSS for less skilled decision makers would be the interface. The interface should be designed in a manner that reflects the familiarity of the decision maker with the problem task and is cognitively compatible with the user's ill formed representation of the domain. Lamberti and Wallace (1990) have begun to address this issue. For example, considering the role of knowledge presentation in diagnostic expert systems as a function of the problem solver's skill level they concluded:

> These results are among the first in studying the impact of question type on user performance. Based on these findings, it appears that certain knowledge organization formats are more conducive to high versus low skilled employee's performance.

It is interesting to note that Lamberti and Wallace used Vitalari and Schenk's definition of skill level as "the possession and organization of domain-specific knowledge and procedural skill that can be accessed and efficiently applied during problem solving." It was observed by Lamberti and Wallace (1990) and in the current study that the performance of the experts deteriorated for the more unfamiliar task. This would imply that task unfamiliarity is closely related to task performance and to problem solving behaviour. This result may be explained by positing that for unfamiliar tasks experts do not have well developed schemata to guide their problem solving search. Therefore their performance decreases and their problem solving search behaviour becomes considerably less directed.

This result would imply that an approach to DSS design that considered tailoring the system cognitively for experts and for novices may be an oversimplification. It may be necessary to develop a DSS that considers at minimum the decision maker's level of expertise in the domain and his or her familiarity with the problem if the DSS is to be truly compatible with the decision maker's cognitive representation of the problem task.

### C.2.c cognitively compatible DSS for training

Hershey et. al. (1990) state:

> We assume that expert skill comes from the repeated experience within a particular problem domain. Specifically it is proposed that enough encounters with the same problem will lead to the acquisition of a solution script: a well-practiced, generic sequence of procedural knowledge which experts can evoke when faced with a problem situation.

A recurring theme throughout the study and one reiterated by Hershey et. al. (1990) is that problem solving skill comes with experience. A key role of a DSS may be to provide decision makers with the practice needed to become proficient problem solvers in a domain. The challenge for the design of a DSS that trains decision makers is to make the system cognitively compatible for the skill level that the decision maker has attained. That is, as the decision maker becomes more experienced in a domain, the problems in the domain become more familiar and the role of the DSS changes in a manner similar to the discussions in sections C.2.a and C.2.b. Therefore a DSS may be constructed for training decision makers of all skill levels, expert or novice.

## D - RECOMMENDATIONS FOR FUTURE RESEARCH

A number of recommendations for future research may be derived from the results of the study. The recommendations are directed at: (i) basic research to better understand how decision makers of varying skill level make decisions and how their decision making behaviour varies within and between skill level groups for differing problem types and task environments and (ii) applied research to better understand how to design, develop and implement computer aided decision support systems.

The first recommendation for future research is directed at answering the following questions: (i) what decision making strategies are used by decision makers and (ii) what conditions influence the decision maker's choice of strategies. The results of this study suggest that three problem solving strategies in particular should be considered as being promising for the development of DSS: (i) the hypothetico-deductive strategy, (ii) pure forward reasoning strategies and (iii) pure backward reasoning strategies.

A key factor that influences the decision maker's choice of strategy appears to be the type of decision task that he or she is being confronted with. Research is required to identify which task characteristics are most influential in affecting the choice of problem solving strategy and to determine how the influence is manifested.

The use of a multi-method approach is highly recommended for research addressing these questions. The multi-method approach provides the richness necessary to study the intricacies of the decision making strategy enabling the researcher to better identify the key behavioural components of the strategy and to make comparisons with other strategies.

A second line of research should be directed towards identifying the variables that affect how a decision maker implements a given decision making strategy and the interaction effects of these variables on decision making behaviour. For example, in this study two factors were identified, task familiarity and decision maker skill level. The main effects of the factors and their interaction effects for dependent measures of decision making behaviour provides important information for designing a DSS that is cognitively compatible with the decision maker. Other research variables may be identified and included in an expanded model of decision making process.

The hypothetico-deductive strategy provides a model for dividing such research into three distinct stages. The first stage would explore how a decision maker searches for and interprets information items in the problem environment to direct information search towards the generation of hypotheses. The second stage would study how decision makers generate and choose between competing hypotheses. The third stage would consider how decision makers engage in a directed search to verify or reject hypotheses and what behaviour follows a rejection. In addition research could be conducted to identify how decision makers integrate the three stages into an overall strategy for problem solving.

Research directed at the hypothetico-deductive strategy would overlap with research of other strategies. For example, stage one described above would share a number of similarities with the pure forward reasoning strategy of problem solving, while stage three research would be applicable to a pure backward reasoning strategy.

The third recommendation for research addresses the need to expand the results of the current research study to different problem solving domains to test for the generalizability of the results to other decision tasks. It is promising that the results of the study are consistent with the results of expert-novice studies in the domains of chess (Chase and Simon, 1973a), medicine (Johnson et., al., 1981), software programming (Lamberti and Wallace, 1990) and accounting (Shields, 1983). This would support the assumption that the results are generalizable to other problem solving tasks in other domains.

The ability to generalize results has important implications for the research and development of DSS. The robustness of the results across domains will help to move DSS design away from an art to more of a science, promoting credibility for the field of DSS research and applications.

At the same time that research is conducted to test for the generalizability of the results, concurrent research is needed that focusses in depth on one problem solving domain. The current study provides an opportunity to begin a stream of research to build an in depth understanding of decision making process within a focussed domain. The subjects in this study were selected from a large multinational organization. There are people located throughout the world (multiple offices in Canada and the United States) with homogeneous profiles with respect to age, training, job responsibilities, problem solving experience, etc. These employees would be expected to display similar problem solving behaviour for the same tasks as the subjects in the study. Therefore, the experiment could be conducted in a number of offices[17] and the results combined with the current study to increase the sample size.

[17]The computer dumps that provided the basis of the problem solving tasks is stored on tape and may be replicated at any of the organizations's offices. At each office the employees have available the same on line database so that the experimental setting may be identically replicated.

With a larger sample size, the results of the statistical tests would have greater power therefore increasing the confidence in the results and the methodology. The research results could also be tested for predictive validity and variations in the research hypotheses and design could be implemented to test the sensitivity of the results. In summary, the study provides a first step in a potential stream of research to explore decision making process.

A fourth direction for future research is to consider the role of declarative and procedural knowledge in problem solving and how to address each type of knowledge in a DSS. The ability to provide structure for unstructured problem tasks has been described as being critical to DSS success. Declarative and procedural knowledge form the basis of the schemata in memory that provide structure for interpreting and solving problems. It is therefore essential that DSS research begin to address these issues. Closely related to this issue is research into the use of deep and surface knowledge during the decision making process. The methodology employed in this study provide an opportunity to explore these issues in detail (see section B.1 of Chapter V).

A fifth recommendation for future research is linked to the need to develop measures to evaluate the effectiveness of DSS that are designed to be cognitively compatible with the user. Surrogate measures such as satisfaction with the DSS, confidence in the decision outcome and system usage may be employed in conjunction with decision accuracy to measure effectiveness of decision support systems. The opportunity exists, however, to augment these measures with measures that are based on the cognitive processes of the user. These measures may provide more meaningful tests of decision effectiveness. For example, decision makers using a DSS could be compared to unaided decision makers for the number of directed information searches conducted during a problem solving session, the number and timing of the hypotheses generated, the efficiency of information cue usage, etc. That is, measures of cognitive processing efficiency and effectiveness may provide more direct measures of DSS effectiveness than current surrogate measures or at least permit the evaluation of DSS from a different perspective.

The final recommendation for research centers on the need to study the degree to which one DSS may be used by multiple decision makers who perform similar decision tasks. It has been mentioned that the results indicating that decision makers of varying skill levels use the same problem solving strategy combined with the observation that there appears to be a key set of information items for diagnosing a problem task provide a basis for designing a DSS that may be adapted to multiple users. Research is required to identify other factors that may influence the use of a DSS by multiple users and to test DSS designs.

In summary, the results of this study provide promise for the development of DSS that are cognitively compatible with the user. The decision making strategy of the user of the system presents a focus for organizing the research and development of DSS that are cognitively compatible. Research aimed at investigating the decision makers' choice of strategy and the implementation of the chosen strategy may provide critical insights for the development of DSS that improve decision making performance and facilitate the design of DSS for multiple users.

## E - SUMMARY AND CONCLUSIONS

Early in Chapter 1, Bronner and De Hoog (1984) were quoted:

> ... many so called decision support systems fail because too much attention is paid to providing information, instead of trying to elicit the underlying decision process in which the information must fit in order to be meaningful for the decision maker.

The present study has attempted to open up the black box and has provided evidence that what is contained in the black box is amenable to study and does in fact exhibit a structure and reason that may be tapped to better understand the processes involved and to apply the knowledge of the processes to help decision makers make "better" decisions.

To achieve these results a multi-method approach has been presented that provides rich data on decision making process and strategy for complex ill-structured diagnosis tasks. The methodology describes how to use "pure" non probing concurrent verbal protocols with aided retroactive verbal protocols in a manner that takes advantage of the strengths of each method. The methodology should be generalizable to many domains where the problem solver is engaged in performing an act that requires him or her to manipulate objects or to communicate with other individuals during the problem solving sessions.

The scheme used for coding protocols and the analysis of the aggregated and directed search sequences used to represent patterns of information search and interpretation may be used to represent a large selection of problem solving tasks. It is particularly suited to the analysis and interpretation of problem solving strategy.

The evidence supporting the use of a hypothetico-deductive strategy by both experts and novices is an important finding, generalizing previous research results across subject domains. It points out the influence of problem solving task on the selection of a problem solving strategy and on the implementation of a strategy to diagnose software problems.

Combined with the result that accurate decision makers use more "key" information items for a diagnosis task, important implications for the rationalization of DSS across decision makers may be derived.

The fact that experts revert to a text book application of the hypothetico-deductive strategy for unfamiliar tasks has had some support in the recent literature (Patel, Groen and Arocha, 1990). The implications are significant for the design of DSS for training and for helping novices to perform more difficult, unfamiliar tasks.

In conclusion, the present study contributes to the DSS and expert-novice literature in a number of important ways. The study is the first in what is anticipated to be a stream of research based upon using subjects from different branches of an organization who have similar profiles, job responsibilities and diagnosis task experience.

**REFERENCES**


Abelson, R. P., Script processing in attitude formation and decision making, Cognition and Social Behavior, J. S. Carrol and J. W. Payne (eds.), Hillsdale, N.J.: Laurence Erlbaum Associates.

Adelson, Beth, 1984, When novices surpass experts: The difficulty of a task may increase with expertise, Journal of Experimental Psychology: Learning, Memory and Cognition, 10, 3: pp. 483 - 495.

Anderson, John R., 1983, The Architecture of Cognition, Cambridge Mass.: Harvard University Press.

Baldwin, Dirk, and Kasper, George M., 1986, Toward representing management-domain knowledge, Decision Support Systems 2: pp. 159-172.

Bartlett, F. C., 1932, Remembering, A Study in Experimental and Social Psychology: Cambridge University Press.

Biggs, Stanley, F., et. al., 1985, The effects of task size and similarity on the decision behavior of bank loan officers, Management Science, 31, 8: pp. 970-986.

Bobrow, D.G. and Norman, D.A., 1975, Some principles of memory schemata, In Representation and Understanding: Studies in Cognitive Science, D.G. Bobrow and A.M. Collins (eds.), New York: Academic Press.

Bronner, A., and De Hoog, R., 1984, The intertwining of information search and decision aiding, Acta Psychologica, 56: pp. 125-139.

Chandrasekaran, B. and Mittal, S., 1983, Deep versus compiled knowledge approaches to diagnostic problem solving, International Journal of Man-Machine Studies, 19: pp. 425-436.

Chase, W. G., and Simon, Herbert A., 1973a, Perception in chess, Cognitive Psychology, 4: pp. 55-81.

Chase, W. G., and Simon, Herbert A., 1973b, The mind's eye in chess, in Visual Information Processing, W. G. Chase (ed.), New York: Academic Press.

Chi, Michelene, Feltovitch, Paul J., and Glaser, Robert, 1981, Categorization and representation of physics problems by experts and novices, Cognitive Science, 5: pp. 121-152.

Chi, Michelene, Glaser, Robert and Rees, Ernest, 1982, Expertise in problem solving, in Advances in the Psychology of Human Intelligence, Robert Sternberg (ed.), Hillsdale, New Jersey: Lawrence Erlbaum: pp. 7-76.

Chi, Michelene, T. H., and Glaser, Robert, 1985, Problem Solving Ability, in Human Abilities an Information Processing Approach, New york: W. H. Freeman and Company.

Cohen, J., 1960, A coefficient of agreement for nominal scales, Educ. Psychol. Meas., 20: pp. 37-46.

Eden, C., Jones, S., and Simms, D., 1983, Messing about in Problems: An Informal Structured Approach to their Identification and Management, New York: Pergammon Press

Einhorn, Hillel J., 1974, Expert Judgment: Some necessary conditions and an example, Journal of Applied Psychology, 59, 5: pp. 562-571.

Einhorn, Hillel J., 1980, Learning from experience and suboptimal rules in decision making, in Cognitive Processes in Choice and Decision Behavior, T. S. Wallsten (ed.), Hillsdale N.J.: Laurence Erlbaum Associates.

Einhorn, Hillel J., and Hogarth, Robin M., 1981, Behavioural Decision Theory: Processes of Judgment and Choice, Annual Review of Psychology, 32: pp. 53-88.

Einhorn, H.J., Kleinmuntz, D.M. and Kleinmuntz, B., 1979, Linear regression and process tracing models of judgment, Psychological Review, 86: pp. 465-485.

Elam, J. and Mead, M., 1986, Can software influence creativity, Working Paper, Harvard University, Boston, MA.

Elstein, A. S., Shulman, and Sprafka, S. A., 1978, Medical Problem Solving, Cambridge Mass.: Harvard University Press.

Ericsson, K. Anders and Simon, Herbert, A., 1980, Verbal reports as data, Psychological Review, 87, 3: pp. 215-251.

Ericsson, K. Anders and Simon, Herbert, A., 1984, Protocol Analysis, Cambridge, MA: MIT Press.

Estes, W. K., 1978, The information processing approach to cognition: A confluence of metaphors and methods, in Handbook of Learning and Cognitive Processes, Vol. 5, W. K. Estes (ed.), Hillsdale, N.J.: John Wiley and Sons.

Feigenbaum, E., 1961, The simulation of verbal learning behavior, Proceedings of the Western Joint Computer Conference, 19: pp. 121-132.

Feltovitch, P.J., Johnson, P.E., Moller J.H. and Swanson, D.B., 1984, The role and development of medical knowledge in diagnostic expertise, in Readings in Medical Artificial Intelligence, W.J. Clancey and E.H. Shortliffe (eds.), Reading MA: Addison:Wesley.

Fleiss, Joseph, 1981, Statistical Methods for Rates and Proportions, Second Edition, New York: John Wiley and Sons.

Ford, Kevin J., Schmitt, Neil, Schechtman, Susan L., Hults, Brian M. and Doherty, Mary L., 1989, Process tracing methods: Contributions, problems and neglected research questions, Organizational Behavior and Human Decision Processes, 43: pp. 75-117.

Gaeth, Gary J., and Shanteau, James, 1984, Reducing the influence of irrelevant information on experienced decision makers, Organizational Behaviour and Human Performance, 33: pp. 263-282.

Glaser, R., 1984, Education and thinking: The role of knowledge, American Psychologist, 39, 2: pp. 93-104.

Grant, J. and Marsden, P., 1988, Primary knowledge, medical education and consultant expertise, Medical Education, 22: pp.173-179.

Greeno, James G., 1978, Natures of problem-solving abilities, in Handbook of Learning and Cognitive Processes, Vol. 5, W. K. Estes (ed.), Hillsdale, N.J.: Lawrence Erlbaum Associates.

Harmon, Paul and King, David, 1984, Expert Systems: Artificial Intelligence in Business, New York: John Wiley.

Hershey, Douglas A., Walsh, David, A., Read, Stephen, J., and Chulef, Ada, S., 1990, The effects of expertise on financial problem solving: Evidence for goal-directed problem solving scripts, Organizational Behavior and Human Decision Processes, 46: pp. 77-101.

Huber G., 1983, Cognitive style as a basis for MIS and DSS designs: Much ado about nothing? Management Science, 29, 5: pp. 372-386.

Jacoby, J., Mazursky, D., Troutman, T. and Kuss, A., 1984, When feedback is ignored: Disutility of outcome feedback, Journal of Applied Psychology, 69: pp. 531-545.

Jarvenpaa, S.L., Dickson, G.W. and DeSanctis, G.,1985, Methodological issues in experimental IS research: Experiences and recommendations, MIS Quarterly, 9, 2: pp. 141-156.

Johnson, E.J., 1985, Expertise and decision under uncertainty: Performance and process, In The Nature Of Expertise, M. Chi, R. Glaser and M. Farr (eds.), Columbus, Ohio: National Center For Research in Vocational Education.

Johnson, Eric J., and Payne, John W., 1985, Effort and accuracy in choice, Management Science, 31, 4: pp. 395-414.

Johnson, Paul. E. et. al., 1981, Expertise and error in diagnostic reasoning, Cognitive Science, 5: pp. 235-283.

Johnson, Paul E., et. al., 1982, Multimethod study of clinical judgment, Organizational Behavior and Human Performance, 30: pp. 201-230.

Johnson, Paul E., 1984, The expert mind: A new challenge for the information scientist, in Beyond Productivity: Information System Development for Organizational Effectiveness, M. A. Behmelmans (ed.), North Holland: Elsevier Science Publishers.

Johnson, Paul E., Zualkernan, Imran, and Garber, Sharon, 1987, Specification of expertise, International Journal of Man-Machine Studies: pp. 161-181.

Kahneman, D., Slovic, P., and Tversky A., 1982, Judgment under uncertainty: Heuristics and biases, in Judgement Under Uncertainty, D. Kahneman, P. Slovic. A. Tversky (eds.), Cambridge: Cambridge Press.

Kelly, George A., 1955, The Psychology of Personal Constructs, New York: Norton.

Keppel, Geoffrey, 1982, Design and Analysis: A Researcher's Handbook, Englewood Cliffs, NJ: Prentice Hall Inc.

Kerlinger, Fred, N., 1973, Foundations of Behavioral Research, Second Edition, Holt, New York: Rinehart and Winston.

Kleinmuntz, Don K., 1985, Cognitive heuristics and feedback in a dynamic decision environment, Management Science, 31, 6: pp. 680-702.

Kotteman J.E., and Remus, W.E., 1987, Evidence and principles of functional and dysfunctional DSS, Omega, 15, 2: pp. 135-143.

Kuipers, Benjamin and Kassirer, Jerome P., 1984, Causal Reasoning in medicine: Analysis of a protocol, Cognitive Science, 8: pp. 363-385.

Kuipers, B., and Kassirer, J.P., 1987, Knowledge acquisition by analysis of verbatim protocols, in Knowledge Acquisition for Expert Systems: A Practical Handbook, Allison C. Kidd ed., New York: Plenum Press: pp. 45-71.

Lamberti, D.M. and Wallace, W.A., 1990, Intelligent interface design: An empirical assessment of knowledge presentation in expert systems, MIS Quarterly, September: pp. 279-311.

Langley, Pat, 1985, Learning to search: from weak methods to domain specific heuristics, Cognitive Science, 9: pp. 217-260.

Larkin, Jill H., et. al., 1980, Models of competence in solving physics problems, Cognitive Science, 4: pp. 317-345.

Little, John, D., 1986, Research opportunities in the decision and management sciences, Management Science, 32, 1: pp. 1-13.

McDermott, J. and Larkin, J. H., 1978, Re-representing textbook physics problems, Proceedings of the 2cnd National Conference of the Canadian Society for Computational Studies of Intelligence, Toronto, Ontario: University of Toronto Press.

Mckeithen, K. B., Reitman, J. S., Reuter, H. H., and Hirtle, S. C., 1981, Knowledge organization and skill differences in computer programmers, Cognitive Psychology, 13: pp. 307-325.

Miller, George A., 1965, The magical number seven plus or minus two: Some limits on our capacity to process information, in Readings in the Psychology of Cognition, Richard C. Anderson and David P. Ausubel (eds.), New York: Holt, Rinehart and Winston Inc.

Minsky, M., 1975, A framework for representing knowledge, in The Psychology Of Computer Vision, P. Winston (ed.), McGraw Hill.

Mintzberg, H., Rasinghani, D., and Theoret, A., 1976, The structure of unstructured decision processes, Administrative Science Quarterly, 21, 2: pp. 246-275.

Munro, M.C., and Davis, G.B., 1977, Determining management information needs: A comparison of methods, MIS Quarterly, 1, 2: pp. 55-67.

Newell, A., Shaw, J. and Simon, H., 1960, Report on a general problem solving program for a computer, In Information Processing: Proceedings of the International Conference on Information Processing, Paris: UNESCO.

Newell, A. and Simon, H. A., 1972. Human Problem Solving, Englewood Cliffs, NJ: Prentice- Hall.

Nisbett, R. E., and Wilson, T. D., 1977, Telling more than we can know: Verbal reports on mental processes, Psychological Review, 84: pp. 231-259.

Norman, Donald A., 1981, What is cognitive science?, in Perspectives on Cognitive Science, Donald A. Norman (ed.), Norwood, N.J.: Ablex Publishing.

Patel, Vimla L. and Frederikson, Carl L., 1984, Cognitive processes in comprehension and knowledge acquisition by medical students and physicians, in Tutorials in Problem Based Learning, M. C. Devoider and H. G. Schmidt (eds.), Assen Holland: Van Goram.

Patel, V.L., and Groen, G.J., 1986, Knowledge based solution strategies in medical reasoning, Cognitive Science, 10: pp. 91-115.

Patel, V. L., Groen, G.J. and Arocha, J.F., 1990, Medical expertise as a function of task difficulty, Memory and Cognition, 18, 4: pp. 394-406.

Payne, John, W., 1976, Task complexity and contingent processing in decision making: An information search and protocol analysis, Organizational Behaviour and Human Performance, 16: pp. 366-387.

Payne, John W., Braunstein, Myron L., and Carroll, John S., 1978, Exploring predecisional behaviour: An alternative approach to decision research, Organizational Behaviour and Human Performance, 22: pp. 17-44.

Pitz, Gordon F., and Sachs, Natalie J., 1984, Judgment and decision: Theory and application, Annual Review of Psychology, 35: pp. 139-163.

Posner, Michael I., and McLeod, Peter, 1982, Information processing models- in search of elementary operations, Annual Review of Psychology, 33: pp. 477-514.

Ramaprasad, Arkalgud, 1987, Cognitive processes as a basis for MIS and DSS design, Management Science, 33, 2: pp. 139-148.

Reed, Stephen K., 1982, Cognition: Theory and Applications, Monteey, Ca.: Brooks/Cole Publishing Company.

Reimann, P. and Chi, M. T. H., 1989, Human expertise, in Human and Machine Problem Solving, K.J. Gilhooly (ed.), New York: Plenum Press: pp. 161-191.

Reiter, R., 1986, A theory of diagnosis from first principles, Artificail Intelligence, 32, 1: pp. 57-95.

Remus, W., and Kotteman, J., 1987, Semi-strucutured recurring decisions: An experimental study of decision making models and some suggestions for DSS, MIS Quarterly, (June): pp. 233-243.

Rorer, L. G., and Widiger, T. A., 1983, Personality, structure and assessment. in Annual Review of Psychology, M. Rosenzweig, Porter, L., eds., Palo Alto: American Reviews Inc.

Rumelhart, D.E. and Norman, D.A., 1983, Representation in memory, Report No. CHIP 116, San Diego: University of California, San Diego, Institute of Cognitive Science.

Schank, Roger C., 1980, Language and Memory, Cognitive Science, 4: pp. 143 - 284.

Schweiger, David, Anderson, Carl and Locke, Edwin, 1985, Complex decision making: A longitudinal study of process and performance, Organizational Behavior and Human Processes, 36: pp.245-272.

Sharda, R., Barr, S. and McDonnell, J., 1988, Decision support system effectiveness: A review and an empirical test, Management Science, 34, 2: pp. 139-159.

Shields, Michael D., 1983, Effects on information supply and demand on judgment accuracy: Evidence from corporate managers, The Accounting Review, LVIII, 2: pp. 284-303.

Siegel, Sidney, 1956, Nonparametric Statistics For the Social Sciences, New York: McGraw-Hill.

Simon D. P. and Simon, H. A., 1978, Individual differences in solving problems, in Children's Thinking: What Develops?, R. Siegler (ed.), Hillsdale, N.J.: Lawrence Erlbaum Associates.

Simon, Herbert A., 1977, The New Science of Management Decision, Englewood Cliffs, N.J.: Prentice Hall Inc.

Simon, Herbert A., 1978, Information processing theory of human problem solving, in Handbook of Learning and Cognitive Processes, Vol. 5, W. K. Estes (ed.), Hillsdale, N.J.: Laurence Erlbaum Associates.

Simon, Herbert A., 1979, Information processing models of cognition, Annual Review of Psychology, 30: pp. 363-396.

Simon, Herbert A., 1981, Cognitive Science: The newest science of the artificial, in Perspectives on Cognitive Science, Donald A. Norman (ed.), Norwood, N.J.: Ablex Publishing.

Sternberg, Robert, 1985, Introduction: What is an information processing approach to human abilities, in Human Abilities an Information Processing Approach, New York: W. H. Freeman and Company.

Sternberg, Robert J., and Lasaga, Maria I., 1984, Approaches to human reasoning: An analytic framework, Artificial and Human Intelligence: pp. 213-227.

Svenson, O., 1979, Process descriptions of decision making, Organizational Behavior and Human Decision Processes, 23: pp. 86-112.

Svenson O., 1989, Eliciting and analyzing verbal protocols in process studies of judgement and decision making, in Process and Structure in Human Decision Making, H. Montgomery and O. Svenson (eds.), New York: John Wiley and Sons.

Sviokla, John, B., 1986a, Business implications of knowledge-based systems,Data Base, 17, 4: pp. 5-19.

Sviokla, John, B., 1986b, Business implications of knowledge-based systems. Part II, Data Base, 18, 1: pp. 5-16.

Todd, Peter and Benbasat, Izak, 1985, The use of protocol analysis in MIS research, Proceedings Information Systems Division, Administrative Association of Canada Conference, Montreal, Quebec.

Todd, Peter and Benbasat, Izak, 1987, Process tracing methods in decision support systems research: Exploring the black box, MIS Quarterly: pp. 493-512.

Tversky, Amos, and Kahneman, Daniel, 1980, Causal schemas in judgments under uncertainty, in Progress in Social Psychology, Vol. 1, M. Fishbein (ed.), Hillsdale: Erlbaum.

Vitalari, N., 1985, Knowledge as a basis for expertise in systems analysis: An empirical study, MIS Quarterly, 9, 3: pp. 221-241.

Vitalari, N. and Schenk, K., 1989, An examination of the cognitive limits of novice systems analysts, Working Paper, Graduate School of Management, University of California, Irvine, CA.

Waldron, V., 1985, Process tracing as a method for initial knowledge acquisition, Proceedings of the Second Conference on Artificial Intelligence Applications, Miami Beach, (Dec.): pp. 661-665.

Wallsten, T. S., 1980, Preface, in Cognitive Processes in Choice and Decision Behavior, T. S. Wallsten (ed.), Hillsdale N.J.: Laurence Erlbaum Associates.

Winer, B.J., 1971, Statistical Principles in Experimental Design, New York: McGraw Hill.

Winston, P. H., 1978, Learning by creatifying transfer frames, Artificial Intelligence, 10: pp. 147-172.

## APPENDIX I

## RESEARCH QUESTIONNAIRE

Participant Identification ________________ SEX ___________

Each individual who participated in the research project has their own personal level of experience which they bring to the problem solving session. In order to be able to put the results of the project in a proper context, it is important to be able to identify the level of experience of each individual participant.

The following questions have been designed to help the researcher determine your basic level of experience. Please note that all responses to these questions will be kept strictly confidential.

1. What is your age? ____ years

The next few questions concern your formal work experience.

2. How long have you been working for IBM

____ years ____ months

3. How long have you worked for the ISC department?

____ years ____ months

4. Have you ever worked for another company in which you performed a job that is similar to the job you perform now with the ISC department or is similar to a job you have performed in the past for the ISC department?

____ yes ____ no

If NO then go to question 5.

4a. Describe the job with the other company or companies briefly.

____________________

____________________

____________________

____________________

4b. How long in total did you work at that job or jobs?

____ years ____ months

4c. In the above job(s) did one of your primary job responsibilities include diagnosing software problems.

____________________

____________________

____________________

Concerning your level of formal education:

5. Have you a technical college or commercial college degree?

____ yes ____ no

If yes, what degree(s) do you have?

____________________

Year completed ____

6. Have you a university degree?

____ yes ____ no

If yes, what degree(s) do you have?

______________________________________________

Year completed ____

7. Have you any other formal educational training?

____ yes ____ no

If yes, please specify.

______________________________________________

______________________________________________

Year completed ____

The following questions concern your IBM sponsored training.

8. Which of the following courses have you completed?

MVS PART A ____ yes ____ no year ____

MVS PART B ____ yes ____ no year ____

MVS XA ____ yes ____ no year ____

PLEASE SPECIFY ANY OTHER MVS RELATED SEMINARS THAT YOU HAVE ATTENDED

______________________________ year ____

______________________________ year ____

PLEASE SPECIFY ANY MVS CUSTOMER EDUCATION SEMINARS THAT YOU HAVE ATTENDED

______________________________ year ____

______________________________ year ____

PLEASE SPECIFY ANY SPECIAL MVS ASSIGNMENTS THAT YOU HAVE PARTICIPATED IN.

________________________________ year ____

________________________________ year ____

The last set of questions address your experience with MVS

9. When was the first time that you formally began diagnosing MVS problems on the job.

____ year

10. Does your current job responsibilities include diagnosing MVS problems?

____ yes ____ no

If not then when was the last time your job responsibilities included diagnosing MVS problems.

____ year

CALCULATION = NUMBER OF YEARS DIAGNOSING MVS PROBLEMS

_____________________________________

CONFIRM: continuity of MVS responsibilities.

According to my calculations then you have been working at diagnosing MVS problems for _____ years. During this time, were you continuously involved in diagnosing MVS problems, that is at least on a weekly basis.

____ yes ____ no

IF NO PROBE:

___________________________________________________________

___________________________________________________________

___________________________________________________________

11. IF CURRENTLY DIAGNOSING MVS:

On the average, how would you describe the amount of time that you currently spend during a week specifically diagnosing MVS problems?

____ 0 to 5 hours

____ between 5 hours and 10 hours

____ between 10 hours and 20 hours

____ more than 20 hours

PROBE: You mentioned that you have been diagnosing MVS problems since _____ . Have you always averaged this amount of time diagnosing MVS problems?

____ yes ____ no

IF NO THEN PROBE:

______________________________________________

______________________________________________

______________________________________________

12. IF CURRENTLY NOT DIAGNOSING MVS:

On the average, how would you describe the amount of time that you spent during a week specifically diagnosing MVS problems at the time you were last responsible for MVS diagnosis?

____ 0 to 5 hours

____ between 5 hours and 10 hours

____ between 10 hours and 20 hours

____ more than 20 hours

PROBE: You mentioned that you have been diagnosing MVS problems since _____ . Have you always averaged this amount of time diagnosing MVS problems?

____ yes ____ no

IF NO THEN PROBE:

______________________________________________

______________________________________________

______________________________________________

## APPENDIX II

## OC4 INFORMATION ITEM USAGE

| NO. | DESCRIPTION | EXPERTS 1 | 2 | 3 | 4 | TE | NOVICES 1 | 2 | 3 | 4 | TN | ALL | IN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 001 | DUMP SUMMARY (SUMDUMP) | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 4 | 8 | 1 |
| 002 | TCB SUMMARY STRUCTURE | 1 | 1 | 1 | 1 | 4 | 1 | 0 | 1 | 1 | 3 | 7 | 1 |
| 003 | OC4 VISUALLY VERIFIED | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 0 | 1 | 3 | 7 | 1 |
| 004 | TCB ATOE (AFA268) | 1 | 1 | 1 | 1 | 4 | 1 | 0 | 0 | 1 | 2 | 6 | 1 |
| 005 | JOBNAME (ATOE) MVSPT2A | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 2 | 3 | 0 |
| 006 | ASID = 100 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 007 | CONTENTS TCB ATOE (AFA26B) | 1 | 1 | 1 | 1 | 4 | 1 | 0 | 1 | 1 | 3 | 7 | 1 |
| 008 | PRB AT AFAICO (FOR AFA268) | 1 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 1 | 3 | 1 |
| 009 | WLIC = 00060011 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 2 | 0 |
| 010 | PSW ADDRESS ATOE (F7E000) | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 4 | 8 | 1 |
| 011 | CDE (AFAB40) | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 2 | 3 | 0 |
| 012 | MODULENAME HMBLIST | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 3 | 4 | 0 |
| 013 | ENTRYPOINT TO HMBLIST IS 61F8 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 2 | 0 |
| 014 | SURB AT AFE708 (FOLLOWING ABEND) | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 2 | 0 |
| 015 | RTM2WA ATOE (CONTENTS) | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 016 | REGISTERS ATOE IN RTM2WA | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 017 | PIC II (PGM) ATOE | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 4 | 8 | 1 |
| 018 | CONTENTS F7E000 NOT AVAILABLE | 1 | 1 | 0 | 0 | 2 | 1 | 1 | 1 | 1 | 4 | 6 | 1 |
| 019 | REG 15 (ATOE)=F7E000 | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 020 | (F7E000) TRANSLATION EXCEPTION ADDRESS (TEA) | 1 | 0 | 1 | 1 | 3 | 0 | 1 | 1 | 0 | 2 | 5 | 1 |
| 021 | TEA IS SAME AS PSW (F7E000) | 1 | 1 | 1 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 3 | 1 |
| 022 | REGISTER 14=673C | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 023 | CONTENTS OF 673C=41 | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 024 | LOAD INSTRUCTION FOR REG15=05EF | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 025 | MODULENAME (673C) HMBLKCTL | 1 | 0 | 1 | 1 | 3 | 0 | 1 | 0 | 0 | 1 | 4 | 0 |
| 026 | MODULE ID HDP2230 (HMBLKCTL) | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 2 | 0 |
| 027 | ENTRY POINT TO MODULE=6224 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 2 | 0 |
| 028 | BALR 14, 15 | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 029 | 58 LOAD INSTRUCTION PRECEDE 05EF | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 2 | 0 |
| 030 | REG2 ATOE IS 8200 | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 031 | LOAD ADDRESS REG15= REG2 + 355 | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 032 | = 8555 | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 033 | INSTRUCTION 235505EF | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 034 | CONTENTS 8555=F7E000 | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 035 | CONTENTS 8554=F7EO | 1 | 1 | 1 | 1 | 4 | 0 | 0 | 1 | 0 | 1 | 5 | 1 |
| 036 | CONTENTS F7EO IS VALID ADDRESS | 1 | 1 | 1 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 3 | 1 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 037 8555 LOAD IS NOT ON FULL WORD BOUNDARY | 1 | 1 | 1 | 1 | 4 | 0 | 0 | 1 | 0 | 1 | 5 | 1 |
| 038 TRACE SUMMARY (MTRACE) | 0 | 1 | 1 | 0 | 2 | 1 | 1 | 0 | 1 | 3 | 5 | 0 |
| 039 VERIFICATION OF SLIP DUMP (VISUAL) | 0 | 1 | 1 | 1 | 3 | 1 | 0 | 1 | 1 | 3 | 6 | 0 |
| 040 LCCA (CONTROL BLOCKS) | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 041 PIC II IS ON 6 BYTE INSTRUCTION | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 042 PSW ATOE POINTS ABOVE LINE | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 043 RTM2WA AT AD3600 (ADDRESS) | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 2 | 0 |
| 044 LOG DATA | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 2 | 3 | 0 |
| 045 NO SOFTWARE MESSAGE IN LOG DATA | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 2 | 3 | 0 |
| 046 SYSTEMTRACE (PSA...) | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 047 ONE SID IN TCB STRUCTURE | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 048 PSW ATOS IS IBC2AOE (PIC II OCCURRED) | 0 | 1 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| 049 CONTENTS OF IBC2AOE | 0 | 1 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| 050 REGISTER C , TOS | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 051 ADDRESS F7D.FA | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 052 MATCH F7DFFA IN 2 REGISTERS | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 053 MODULE NAME(AT AD3600) NOT AVAILABLE | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 054 REG 11 ATOE (40006202) | 0 | 1 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| 055 REG 11 CONTAINS FLAG | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 056 CONTENTS 6202 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 057 SEQ. OF BASE REGISTER LEAD TO REG2 (3BASE REGS) | 0 | 1 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| 058 CONTENTS 8200 | 0 | 1 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| 059 HIGTER ORDER BYTE ON (REG 14) | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 060 ASSEMBLY DATE HMBLKCTL 12/85 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 061 ADDRESSES DONOT REPEAT (6700) NO LOOP | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 062 CODE AFTER 673C | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 063 CODE BEFORE F7E0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 064 CODE AFTER F7E0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 065 PROGRAM IS PRIVATE AREA SUBPOOL (ATOE) | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 066 REGISTERS ATOS | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 067 MODULE ATOS IEAVTRAD | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 2 | 0 |
| 068 BASELEVEL CODE=EBB2220 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 069 ENTRY POINT TO IEAVTRAD=/BC2860 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 070 OFFSET INTO IEAVTRAD=1AE | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 071 DUMP TITLE | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 2 | 2 | 0 |
| 072 DUMP 10=CSOP | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 073 SYMPTOMS NOT AVAILABLE | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 074 WARNING MESSAGES WITH TITLE | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 075 ASCB FB9200 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 076 PRB AFEF78 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 077 PSW ADDRESS (AFEF78) 070C1000 81034438 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 078 TCB AFA8A8 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 079 PRB AFA800 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 080 PSW(AFA800)=070C1000 00F3A2E6 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 081 SYSTEM CPU STATUS (SYSTEM END SUMMARY) | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 082 PSW IN CPU STATUS DIFFERENT PSW F3A2E6 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 083 SYSTEM ENV. WORKSHEET | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 084 TCB AFE4A8 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 085 CURRENT TCB IN WORKSHEET IS DIFFERENT AFE4A8 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 086 ASCB IN WORKSHEET SAME AS ASCB FB9200 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 087 PSW IN WORKSHEET=PSW IN PRB AFE150 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 088 PRB AFE150 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 089 PSW IN PRB AFE150= 070C1000 81B83CE | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 090 SVCD IN MTRACE (ATOE) | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 091 PSW FOR SVCD=07803000 810FEB06 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 2 | 2 | 0 |
| 092 MULTIPLE SVC AND SVCR PRIN SVCD (AFA268) | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 093 SVC'S COMMENCE AT E5C418 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 094 TCB AT AFE708 CONTAINS 0'S | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 095 PSA AT LOCATION 8000 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 096 PSW STARTED AT 10FDB00 (LEAD TO ATOES) | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 097 CONTENTS 10FEB06 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 2 | 2 | 0 |
| 098 MODULE IVAFRLK (AT FEB06) | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 099 CONTENTS 1B83C3E NOT AVAILABLE | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 100 PSW IN CPU STATUS 81BC2A0E | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 101 ADDRESS 81BC2A0E NOT AVAILABLE | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 102 PSW AT 070C000 A05A8B8 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 103 CONTENTS 105A8B8 NOT AVAILABLE | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 104 CURRENT DUMP TASK ABEND IS AFE4A8 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 105 SVC3C ISSUED | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 106 TCB FLAGS IN AFA268 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 107 CONTENTS 61F8 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 108 STORES PRIOR TO 61F8 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 109 TCB STRUCTURE GIVES NO REGISTERS | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 110 AFTER PIC 11-I/O TO CHANNEL 223 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 111 AFTER I/O CHANNEL 223 ABEND ISSUED | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 112 MULTIPLE GETMAIN/FREE-MAIN PRIOR PIC 11 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 113 TRACE RECORDS AT 3225 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 114 SVCA IN TRACE | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 115 REGISTERS 15,0,1 FOR SVCD | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 2 | 2 | 0 |
| 116 SVCA IS FREEMAIN FREED UP STORAGE | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 117 AT 5F94 FOR 32F0 BYTES STARTING AT 31D10 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 118 SVC5 FREEMAIN FOR 32F0 BYTES FROM 5F94 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 119 SVC4 GETMAIN FOR 56EB BYTES STARTING AT 5FA0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 120 SVC5 FOLLOWS SVC4 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 121 CONTENT 5F94 AVAILABLE | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 122 PIC 11 FOLLOWS SVC5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 123 REG 1 POINTS TO PARM LIST AT 5F94 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 124 SUCH IS UNCRUDITIONAL GETMAIN SUBPOOL 16 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 125 SVC5 IS A VARIABLE REQUEST | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 126 OFFSET INTO HMBLKCTL 81518 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 127 REG C CONTAINS 55C0 (ATOE) | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 128 CODE BEFORE 673C | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 129 BRANCH TRACE NOT AVAILABLE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 2 | 0 |
| 130 SDWA CONFIRMS PIC 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 131 MULTIPLE PGM (PIC 11) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 132 CODE PRIOR 10FEB06 MANY F,0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 133 CONTENTS 10FEB06 VALID CODE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 134 ADDRESSES AT SVCD NOT SAME (LOOP) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |

# APPENDIX III

## 30A INFORMATION ITEM USAGE

| NO. | DESCRIPTION | EXPERTS 1 | 2 | 3 | 4 | TE | NOVICES 1 | 2 | 3 | 4 | TN | ALL | IN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 001 | SYSTEM SUMMARY | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 002 | TCB ATOE = AFA2B8 | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 0 | 3 | 7 | 1 |
| 003 | CURRENT JOB NAME ATOE = 599 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 2 | 1 |
| 004 | CURRENT ASID ATOE = 100 | 1 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| 005 | SUMMARY FORMAT (DUMP) | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 4 | 8 | 1 |
| 006 | TCB STRUCTURE FOR AFA268 | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 0 | 3 | 7 | 1 |
| 007 | COMPLETION CODE 30A IN TCB AFA268 | 1 | 1 | 1 | 1 | 4 | 1 | 0 | 1 | 0 | 2 | 6 | 1 |
| 008 | PRB AT AFA210 | 1 | 0 | 1 | 1 | 3 | 1 | 1 | 1 | 0 | 3 | 6 | 1 |
| 009 | WLIC = 2000 FOR AFA210 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 010 | PSW = 81D00F50 FOR AFA210 | 1 | 0 | 1 | 0 | 2 | 1 | 1 | 1 | 0 | 3 | 5 | 0 |
| 011 | SVRB AT AFE7D8 | 1 | 1 | 0 | 1 | 3 | 0 | 1 | 1 | 0 | 2 | 5 | 1 |
| 012 | SVRB AT AFE8F0 | 1 | 0 | 1 | 1 | 3 | 0 | 1 | 0 | 0 | 1 | 4 | 0 |
| 013 | PRB AT AFA4C0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 014 | WLIC = 20033 FOR AFA4C0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 015 | PSW = 8/BE4720 FOR AFA4C0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 016 | RTM2WA SUMMARY | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 0 | 1 | 3 | 7 | 1 |
| 017 | RTM2WA(1) | 1 | 0 | 1 | 1 | 3 | 0 | 1 | 0 | 0 | 1 | 4 | 1 |
| 018 | COMPLETION CODE USER 257 (RTM1) | 1 | 0 | 1 | 1 | 3 | 0 | 1 | 0 | 0 | 1 | 4 | 1 |
| 019 | PSW ATOE = 81D00F50 (RTM1) | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 2 | 3 | 0 |
| 020 | PRB, PSW IN RTM2WA (1) = PSW, PRB IN CURRENT | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 021 | RTM2WA(2) | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 0 | 1 | 2 | 6 | 1 |
| 022 | COMPLETION CODE 30A IN RTM2WA(2) | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 0 | 1 | 2 | 6 | 1 |
| 023 | PSW ATOE = 8113C252 | 1 | 0 | 1 | 1 | 3 | 0 | 0 | 0 | 1 | 1 | 4 | 0 |
| 024 | SYSTEM TRACE (VERBX TRACE) | 1 | 1 | 1 | 0 | 3 | 1 | 1 | 1 | 1 | 4 | 7 | 1 |
| 025 | SVC D (AFTER SVCE) | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 3 | 4 | 0 |
| 026 | SVCR A (AFTER SVCE) | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 3 | 4 | 0 |
| 027 | 816FEB06 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 028 | SVCE D | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 4 | 8 | 1 |
| 029 | SVC A (AT SVCE) | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 4 | 8 | 1 |
| 030 | PIC 11'S IN TRACE | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 031 | CONTENTS 8113C252 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2 | 0 |
| 032 | MODULE NAME AROUND 8113C252 = 1EV235ERR | 1 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 1 | 1 | 3 | 0 |
| 033 | FMID = JBB2220 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 034 | APPLICATION JOB SYSTEM LOG NOT AVAILABLE | 1 | 1 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 1 | 3 | 0 |
| 035 | USER 257 ABEND DUMP NOT AVAILABLE | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 036 USER ABEND ISSUES 30A ABEND (2 ABENDS) | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 0 | 1 | 2 | 6 | 1 |
| 037 SVCA (ABOVE SVCE) IS GETMAIN | 1 | 0 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 3 | 1 |
| 038 BRANCH TRACE ON | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 0 | 1 | 3 | 7 | 0 |
| 039 BRANCH AT 1F00EF0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 2 | 0 |
| 040 BRANCH AT 1112BD6C | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 2 | 0 |
| 041 SVCR D MUCH BEFORE SVCE | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 2 | 0 |
| 042 SVCR D AT CODE FF9500 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 2 | 0 |
| 043 SVC A (AT SVCE) ISSUED AT 1BE7426 | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 4 | 8 | 1 |
| 044 MTRACE NOT AVAILABLE | 0 | 1 | 1 | 1 | 3 | 0 | 0 | 1 | 0 | 1 | 4 | 0 |
| 045 SUBPOOL ID 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 2 | 3 | 0 |
| 046 CDE AT AC6028 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 2 | 0 |
| 047 SERIES OF SVC78'S PREVIOUS SVCE | 0 | 1 | 1 | 0 | 2 | 1 | 1 | 1 | 1 | 4 | 6 | 1 |
| 048 SVC 3C AFTER SVC78 BEFORE SVCA | 0 | 1 | 0 | 1 | 2 | 1 | 0 | 1 | 0 | 2 | 4 | 0 |
| 049 SVC 33R PREVIOUS SVCE | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 050 SVCRA MUCH PREVIOUS SVCE | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 2 | 0 |
| 051 SVCRD MUCH PREVIOUS TO SVCE | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 2 | 0 |
| 052 SVCR 23 (BELOW SVCR33) | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 053 SVCR 38 (ABOVE SVC33) | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 054 LOAD MODULE NAME IGCOIOIC | 0 | 1 | 1 | 1 | 3 | 0 | 0 | 0 | 1 | 1 | 4 | 1 |
| 055 CONTENTS IBE7426 NOT IN STORAGE | 0 | 1 | 1 | 1 | 3 | 1 | 0 | 0 | 1 | 2 | 5 | 0 |
| 056 AMBLIST FOR IGCOIOIL | 0 | 1 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 3 | 1 |
| 057 SUC 33 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 058 DUMP TITLE IMPLIE RM PROBLEM | 0 | 1 | 1 | 1 | 3 | 1 | 0 | 0 | 0 | 1 | 4 | 0 |
| 059 SVCR78 JUST BEFORE SUCA | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 3 | 4 | 1 |
| 060 CURRENT ERROR TYPE I | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2 | 1 |
| 061 REG 15,0,1, ARE IMPORTANT FOR SVCA | 0 | 0 | 1 | 1 | 2 | 1 | 1 | 0 | 1 | 3 | 5 | 1 |
| 062 REG 15 - FD03 (SUB POOL) | 0 | 0 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 4 | 6 | 1 |
| 063 REG 0 460 BYTE (FREE MAINED | 0 | 0 | 1 | 1 | 2 | 1 | 1 | 0 | 1 | 3 | 5 | 1 |
| 064 REG 1 7FFEIAO (ADDRESS OF STORAGE TO FREEMAIN | 0 | 0 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 4 | 6 | 1 |
| 065 MODULE IGCOIOIC IS IC IN SVC13 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 066 DISPLACEMENT INTO IGCOIOIC IS B426 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2 | 1 |
| 067 CSECT FOR FAILING INSTRUCTION (IN AMBUST) | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 068 VSMDATA | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 2 | 0 |
| 069 VSM SUBPOOL TRANSLATION TABLE NOT AVAILABLE | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 070 REG 15 IN RTM2WA(2) = 18 | 0 | 0 | 1 | 1 | 2 | 1 | 0 | 0 | 0 | 1 | 3 | 1 |
| 071 RETURN CODE 18 - PRIMARY SUBPOOL NOT FOUND | 0 | 0 | 1 | 1 | 2 | 1 | 1 | 1 | 0 | 3 | 5 | 1 |
| 072 PSA AT SVC A IS FD (253DECM) | 0 | 0 | 1 | 1 | 2 | 1 | 0 | 0 | 0 | 1 | 3 | 0 |
| 073 SUBPOOL FD NOT IN VSMDATA | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 074 SEVERAL MESSAGES IN VSMDATA | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 2 | 0 |
| 075 LOG ON PROC 2.2 (IS CORRECT) | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 076 CONTENTS TFFFE1AO NOT AVAIL | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 2 | 3 | 0 |
| 077 IPCS VERB ANALYZE | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 078 LSQA | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 2 | 0 |
| 079 TCB AFA2B8 NOT IN LSQA (VSMDATA) | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 080 ADDRESS 7FF16B38 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 081 CONTENTS 7FF16B38 NOT AVAILABLE | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 082 7FFFE1AO IN TCB AFE8FO | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 083 REG 0 CONTAINS HIGH ORDER BYTE | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 2 | 3 | 0 |
| 084 AMBLIST - FREEMAIN LOCATED | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 085 AMBLIST - INSERT CHAR LOADS REG F | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 086 XA DUMP (HIGH STORAGE) | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 087 BFDEF7 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 088 MASK IS D | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 089 30A IS FREEMAN PROBLEM WITH COMPLETION CODE IN SDWA | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 4 | 4 | 1 |
| 090 RTM2WA IS AT ADDRESS AC40F8 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 091 CONTENTS OF AC40F8 | 0 | 0 | 0 | 0 | 0 | . | 0 | 0 | 0 | 1 | | |
| 092 ADDRESS AC5B40 (SDWA) | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 3 | 3 | 0 |
| 093 CONTENTS AC5B40 (ARE VALID) | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 3 | 3 | 0 |
| 094 SDWA420 = AFE7D8 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 095 CONTENT AFE7D8 IS ALL 0'S | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 096 SDWA BREAKDOWN | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 2 | 2 | 0 |
| 097 SDWA CRC FEILD IS AT AC5E98 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 098 PREVIOUS TO AC5ED8 IS RC 18 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 099 SUBPOOL 0 IS PRIVATE LOW | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 2 | 2 | 1 |
| 100 SUC78 PREVIOUS SVCA ARE GETMAINS | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 101 SVC78 ID (BEFORE SUCA) IS 01 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 102 SVC78 SUBPOOL FD01 (MULTIPLE) | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 103 SVC78 SUBPOOL E601 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 104 ADDRESS O7OCIOOO IS SAME FOR SK78 AND SVCA | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 105 SV78 CAN BE USED FOR STORAGE ABOVE LINE | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 106 MULTIPLE BRANCHES | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 107 DISPATCH (DSP)8112CBD | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 2 | 2 | 0 |
| 108 112CBDd NOT AVAILABLE IN DUMP | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 109 ADDRESS 1BE7522 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 110 IBE7522 NOT AVAILABLE IN STORAGE | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 111 PSW IBE7426 IN SURB AFE7D8 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 112 ADDRESS 7FFEIAO (IN REG1) (ABOVE LINE) IS VERY HIGH | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 3 | 3 | 0 |
| 113 SDWA NOT IN RTM2WA (1) (ALL 0'S) | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 114 ASCB IN RTM2WA | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 115 ASXB IN RTM2WA | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 116 FRWA NOT AVAILABLE | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 117 ACG600 ADDRESS PREVIOUS RTM2WA | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 118 CONTENTS 1D00F50 NOT AVAILABLE | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 119 SDWA+238=AC5078 ACSCD8 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 120 FD000008 IN AFA210 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 121 SUC'S PRIOR TO SUCA ARE POSITIVE | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 122 SUC30 IS FIRST NEGATIVE | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | | 0 |
| 123 SUC78 IS FREEMAIN | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 124 SYMPTOMS | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 125 CONTENTS OF ADDRESS 000F50 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 126 (R)LOGREC | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 127 SUCA REG1 CONTAINS POS VALUE FREEMAIN | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 128 SUCA FREEMAIN ABOVE LINE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 129 7FFFEIAO APPEARS ONCE IN DUMP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | ? | 0 |
| 130 (R)CPUDATA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 131 NO SOWA IN CPUDATA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 132 (2) VERBX LOGDATA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 133 ABEND B37 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 134 SDWA IN LOGDATA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |

# APPENDIX IV

## OC4 INFORMATION ITEM KEY ITEM USAGE

| | | EXPERTS | | | | | NOVICES | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NO. | DESCRIPTION | 1 | 2 | 3 | 4 | TE | 1 | 2 | 3 | 4 | TN | ALL | IN |
| 001 | DUMP SUMMARY (SUMDUMP) | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 4 | 8 | 1 |
| 002 | TCB SUMMARY STRUCTURE | 1 | 1 | 1 | 1 | 4 | 1 | 0 | 1 | 1 | 3 | 7 | 1 |
| 003 | OC4 VISUALLY VERIFIED | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 0 | 1 | 3 | 7 | 1 |
| 004 | TCB ATOE (AFA268) | 1 | 1 | 1 | 1 | 4 | 1 | 0 | 0 | 1 | 2 | 6 | 1 |
| 005 | CONTENTS TCB ATOE (AFA26B) | 1 | 1 | 1 | 1 | 4 | 1 | 0 | 1 | 1 | 3 | 7 | 1 |
| 006 | PRB AT AFAICO (FOR AFA268) | 1 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 1 | 3 | 1 |
| 007 | PSW ADDRESS ATOE (F7E000) | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 4 | 8 | 1 |
| 008 | RTM2WA ATOE (CONTENTS) | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 009 | REGISTERS ATOE IN RTM2WA | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 010 | PIC II (PGM) ATOE | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 4 | 8 | 1 |
| 011 | CONTENTS F7E000 NOT AVAILABLE | 1 | 1 | 0 | 0 | 2 | 1 | 1 | 1 | 1 | 4 | 6 | 1 |
| 012 | REG 15 (ATOE)=F7E000 | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 013 | (F7E000) TRANSLATION EXCEPTION ADDRESS (TEA) | 1 | 0 | 1 | 1 | 3 | 0 | 1 | 1 | 0 | 2 | 5 | 1 |
| 014 | TEA IS SAME AS PSW (F7E000) | 1 | 1 | 1 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 3 | 1 |
| 015 | REGISTER 14=673C | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 016 | CONTENTS OF 673C=41 | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 017 | LOAD INSTRUCTION FOR REG15=05EF | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 018 | BALR 14, 15 | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 019 | REG2 ATOE IS 8200 | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 020 | LOAD ADDRESS REG15= REG2 + 355 | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 021 | = 8555 | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 022 | INSTRUCTION 235505EF | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 023 | CONTENTS 8555=F7E000 | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 1 | 0 | 2 | 6 | 1 |
| 024 | CONTENTS 8554=F7EO | 1 | 1 | 1 | 1 | 4 | 0 | 0 | 1 | 0 | 1 | 5 | 1 |
| 025 | CONTENTS F7EO IS VALID ADDRESS | 1 | 1 | 1 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 3 | 1 |
| 026 | 8555 LOAD IS NOT ON FULL WORD BOUNDARY | 1 | 1 | 1 | 1 | 4 | 0 | 0 | 1 | 0 | 1 | 5 | 1 |

# APPENDIX V

## 30A INFORMATION ITEM USAGE

| NO. | DESCRIPTION | EXPERTS 1 | 2 | 3 | 4 | TE | NOVICES 1 | 2 | 3 | 4 | TN | ALL | IN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 001 | TCB ATOE = AFA2B8 | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 0 | 3 | 7 | 1 |
| 002 | CURRENT JOB NAME ATOE = 599 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 2 | 1 |
| 003 | SUMMARY FORMAT (DUMP) | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 4 | 8 | 1 |
| 004 | TCB STRUCTURE FOR AFA268 | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 0 | 3 | 7 | 1 |
| 005 | COMPLETION CODE 30A IN TCB AFA268 | 1 | 1 | 1 | 1 | 4 | 1 | 0 | 1 | 0 | 2 | 6 | 1 |
| 006 | PRB AT AFA210 | 1 | 0 | 1 | 1 | 3 | 1 | 1 | 1 | 0 | 3 | 6 | 1 |
| 007 | SVRB AT AFE7D8 | 1 | 1 | 0 | 1 | 3 | 0 | 1 | 1 | 0 | 2 | 5 | 1 |
| 008 | RTM2WA SUMMARY | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 0 | 1 | 3 | 7 | 1 |
| 009 | RTM2WA(1) | 1 | 0 | 1 | 1 | 3 | 0 | 1 | 0 | 0 | 1 | 4 | 1 |
| 010 | COMPLETION CODE USER 257 (RTM1) | 1 | 0 | 1 | 1 | 3 | 0 | 1 | 0 | 0 | 1 | 4 | 1 |
| 011 | RTM2WA(2) | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 0 | 1 | 2 | 6 | 1 |
| 012 | COMPLETION CODE 30A IN RTM2WA(2) | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 0 | 1 | 2 | 6 | 1 |
| 013 | SYSTEM TRACE (VERBX TRACE) | 1 | 1 | 1 | 0 | 3 | 1 | 1 | 1 | 1 | 4 | 7 | 1 |
| 014 | SVCE D | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 4 | 8 | 1 |
| 015 | SVC A (AT SVCE) | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 4 | 8 | 1 |
| 016 | USER ABEND ISSUES 30A ABEND (2 ABENDS) | 1 | 1 | 1 | 1 | 4 | 0 | 1 | 0 | 1 | 2 | 6 | 1 |
| 017 | SVCA (ABOVE SVCE) IS GETMAIN | 1 | 0 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 3 | 1 |
| 018 | SVC A (AT SVCE) ISSUED AT 1BE7426 | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 4 | 8 | 1 |
| 019 | SERIES OF SVC78'S PREVIOUS SVCE | 0 | 1 | 1 | 0 | 2 | 1 | 1 | 1 | 1 | 4 | 6 | 1 |
| 020 | LOAD MODULE NAME IGCOIOIC | 0 | 1 | 1 | 1 | 3 | 0 | 0 | 0 | 1 | 1 | 4 | 1 |
| 021 | AMBLIST FOR IGCOIOIL | 0 | 1 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 3 | 1 |
| 022 | SVCR78 JUST BEFORE SUCA | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 3 | 4 | 1 |
| 023 | CURRENT ERROR TYPE I | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2 | 1 |
| 024 | REG 15,0,1, ARE IMPORTANT FOR SVCA | 0 | 0 | 1 | 1 | 2 | 1 | 1 | 0 | 1 | 3 | 5 | 1 |
| 025 | REG 15 - FD03 (SUB POOL) | 0 | 0 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 4 | 6 | 1 |
| 026 | REG 0 460 BYTE (FREE MAINED) | 0 | 0 | 1 | 1 | 2 | 1 | 1 | 0 | 1 | 3 | 5 | 1 |
| 027 | REG 1 7FFEIAO (ADDRESS OF STORAGE TO FREEMAIN | 0 | 0 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 4 | 6 | 1 |
| 028 | DISPLACEMENT INTO IGCOIOIC IS B426 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2 | 1 |
| 029 | CSECT FOR FAILING IN-STRUCTION (IN AMBUST) | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 030 | REG 15 IN RTM2WA(2) = 18 | 0 | 0 | 1 | 1 | 2 | 1 | 0 | 0 | 0 | 1 | 3 | 1 |
| 031 | RETURN CODE 18 - PRIMARY SUBPOOL NOT FOUND | 0 | 0 | 1 | 1 | 2 | 1 | 1 | 1 | 0 | 3 | 5 | 1 |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 032 AMBLIST - FREEMAIN LOCATED | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 033 30A IS FREEMAN PROBLEM WITH COMPLETION CODE IN SDWA | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 4 | 4 | 1 |
| 034 SUBPOOL 0 IS PRIVATE LOW | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 2 | 2 | 1 |
| 035 ADDRESS 070CI000 IS SAME FOR SK78 AND SVCA | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 036 SV78 CAN BE USED FOR STORAGE ABOVE LINE | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| ARE POSITIVE | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 037 SVC'S PRIOR TO SVCA | | | | | | | | | | | | |
| ARE POSITIVE | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 038 SUC78 IS FREEMAIN | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| POS VALUE FREEMAIN | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |