

MERGING DECISION SUPPORT AND EXPERT SYSTEM TECHNOLOGIES:
AN EVOLUTIONARY ANALYSIS AND PROPOSALS
FOR AN ARCHITECTURAL FRAMEWORK.

A thesis submitted to the Department of
Accounting, University of Cape Town, in
fulfillment of the requirements for the degree
Master of Commerce.

Gideon Daniel Smith

August 1987

The University of Cape Town has been given
the right to reproduce this thesis in whole
or in part. Copyright is held by the author.

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

ACKNOWLEDGEMENTS

While there are many people one could thank the following deserve special mention:

Dr. Peter Lay and Trevor Wegner, my supervisors, for their help in the form of constructive criticism and useful suggestions.

My family for their tolerance and moral support.

My parents for their sustained interest in my academic activities.

Except as noted above this work is entirely my own and all references are adequately cited.

Gideon Smith

August 1987

NOTE

There are some words in the English language which have alternative spellings. Some of these differences arise because of English and American preferences. The convention in South Africa is to use the English version in certain circumstances (colour rather than color) while in other circumstances there is no convention.

In this thesis there are numerous words for which there are optional spellings such as the choice between s or z, or using a single or double l in adding a suffix. The choice has been to use the z as in formalize and the double l as in modelling. However where direct quotes have been used or in titles of papers these have been reproduced using the spelling in the source document.

A further convention which has been followed is the use of a hyphen to connect the individual words describing a single concept as in decision-maker.

ABSTRACT

The fact that competence in decision-making differentiates the effective manager from the ineffective manager (Harrison, 1981:1), makes decision-making one of management's principle concerns (Uliana, 1986:1). The implication of this for DSS research is that means have to be sought that could contribute towards raising the quality of managerial decision-making.

The objective of this research is therefore, *firstly*, to establish whether traditional DSS is adequately equipped to adapt to changing user needs. *Secondly*, alternative theoretical foundations of Expert System (ES) enhanced DSS are examined.

A basic assumption made by the author is that DSS is an evolving area of interest. A literature survey confirmed that DSS is experiencing demand pull and that technological innovation is called for in order to solve the more sophisticated user demands. More specifically, the survey has shown that (i) some user requirements, such as the need for training, have not yet been fully addressed, (ii) user demands, such as the expansion of DSS to cover problem areas tradition-

ally regarded as non-applicable to computer-based DSS, are difficult to meet using traditional methods and (iii) researchers in various other disciplines are developing new and exciting techniques which are worth investigating.

The successful implementation of a number of expert systems has focussed attention on ES technology as a possible source of solutions for some of the problems experienced in the DSS field. The similarities and differences between ES and DSS as well as the applicability of ES technology to problem areas in DSS have therefore been investigated. Inferencing, automated learning and the ability to perform goal-seeking are amongst the ES features that could transform DSS into computerized models of managerial judgement and decision-making.

To capture the problem-solving skills of experts in a particular domain and then make this knowledge available to both experienced and inexperienced managers for decision-making and/or training purposes require a system with the following components:

- (i) a knowledge-based database management system;
- (ii) a knowledge-based model management system;
- (iii) an intelligent man-machine interface;
- (iv) an intelligent computer-aided training system;

- (v) a comprehensive tool kit and a tool manager;
- (vi) a knowledge-based scenario management system; and
- (vii) a communication and control component.

To provide a reader with a conceptual view of the structure of, what has been termed the new generation of expert DSS (EDSS), two alternative architectural frameworks have been developed.

The components and frameworks referred to above were presented to a panel of academics and practitioners for evaluation in terms of necessity, sufficiency and acceptability. The results of the Delphi study indicated that a comprehensive toolkit and a knowledge-based scenario component were the two most likely components that would contribute towards enhancing traditional DSS.

It is expected that this research could lead to the widening of DSS boundaries so as to include a larger number of support-providing components. It is also expected to provide clarity on the structural composition of such expanded DSS through identification of possible system components and their interfaces. Since the characteristic features and the expected impact of EDSS on the decision-maker's organizational role are highlighted, this research could contribute towards overcoming user resistance.

CONTENTS

	PAGE.
ACKNOWLEDGEMENTS.	(i)
NOTE.	(ii)
ABSTRACT.	(iii)
CHAPTER 1: <u>INTRODUCTION.</u>	1-14
1.1 PREAMBLE.	1
1.2 PROBLEM STATEMENT.	5
1.3 OBJECTIVES.	7
1.4 RESEARCH METHODOLOGY.	9
1.5 LIMITATIONS.	11
1.6 EXPECTED CONTRIBUTIONS.	11
1.7 ORGANIZATION OF THESIS.	13
CHAPTER 2: <u>DECISION MAKING AND DECISION SUPPORT.</u>	15-42
2.1 INTRODUCTION.	15
2.2 DECISIONS AND DECISION-MAKING.	15
2.3 THE DECISION-MAKER.	25
2.4 DECISION SUPPORT SYSTEMS.	32
2.5 CONCLUSION.	41

CHAPTER 3: <u>THE NEED FOR CHANGE.</u>	43-71
3.1 INTRODUCTION.	43
3.2 SHORTCOMINGS AND LIMITATIONS OF DSS.	44
3.3 OTHER PRESSING REASONS.	67
3.4 CONCLUSION.	70
CHAPTER 4: <u>THE PROMISE OF EXPERT SYSTEMS.</u>	72-112
4.1 INTRODUCTION.	72
4.2 OVERVIEW OF MAJOR ES COMPONENTS.	73
4.3 CHARACTERISTICS OF ES.	78
4.4 SIMILARITIES/DIFFERENCES BETWEEN ES AND DSS.	84
4.5 APPLICABILITY OF ES TECHNOLOGY TO PROBLEMS OF DSS.	89
4.6 CONCLUSION.	111
CHAPTER 5: <u>EDSS: THE MERGING OF ES AND DSS TECHNOLOGIES.</u>	113-135
5.1 INTRODUCTION.	113
5.2 CHARACTERISTICS OF EDSS.	114
5.3 USER'S VIEW OF AN EDSS.	123
5.4 IMPACT ON THE DECISION-MAKER'S ROLE.	127
5.5 CONCLUSION.	134

CHAPTER 9: <u>CONCLUSIONS AND RECOMMENDATIONS.</u>	196-207
9.1 RESTATEMENT OF OBJECTIVES.	196
9.2 EVALUATION IN TERMS OF HYPOTHESES/OBJECTIVES.	197
9.3 REMAINING PROBLEMS.	202
9.4 RESEARCH CONTRIBUTIONS.	203
9.5 FUTURE RESEARCH DIRECTIVES.	205
9.6 CONCLUSION.	207
APPENDIX.	208-217
BIBLIOGRAPHY.	218-236

CHAPTER 1.

INTRODUCTION.

1.1 PREAMBLE.

Competence in decision-making differentiates the effective manager from the ineffective manager (Harrison, 1981:1). Decision-making is therefore one of management's principle concerns (Uliana, 1986:1). The broad objective of this research is to examine means that could raise the level of managerial effectiveness by improving the quality of their decision-making.

To reach a decision a manager normally relies on intuition, experience, and information (Radford, 1981:1-2, 19). The quality of the decision and therefore the effectiveness of the decision-maker are consequently determined by the decision-maker's own flair or expertness at problem-solving and the quality of the available information.

But, "human beings are information processing systems operating largely in serial fashion, and possessing very modest computational powers in comparison with the complexity of the problems with which their environment confronts them" (Simon, 1977b:173). It was against the background of this obvious need for some form of support that Scott Morton (1977) introduced the concepts of Decision

Support Systems under the term "management decision systems".

DEFINITION.

A *Decision Support System* (DSS) is an interactive computer-based system that assists managers in dealing with ill-structured problems; it supports rather than replaces managerial judgement and is aimed at improving the effectiveness of decision-makers (Keen and Scott Morton, 1978:1).

DSS, like all computer-related fields, is an evolving area of interest. This can be attributed to two main reasons:

(i) Demand Pull.

As the users of DSS become more sophisticated over time, their requirements tend to change. They ask for (demand) more advanced decision support facilities, the development of systems that cover problem areas traditionally regarded as non-applicable to computer-based support and changes to existing systems to increase the efficiency of the system and/or the effectiveness of the user (Hurst, Ness, Gambino and Johnson, 1983:123-124).

(ii) Technology Push.

DSS is an area of interest which brings together a number of independent disciplines, examples of which are Operations Research/Management Science, Database technology, and Systems Engineering (Bonczek, Holsapple and Whinston, 1984:142). Researchers in these fields

are constantly developing new ideas and improved techniques. In addition, DSS researchers continuously scan *related fields* for techniques that could be applied successfully in DSS [Remus and Kotteman (1986); Blanning (1984a,1984c); Elam and Henderson (1983); Gorry and Krumland (1983)]. When such techniques are incorporated in DSS, whether to meet user demands or simply to raise the level of system sophistication, it represents a case of technology push.

Expert Systems, bearing many similarities to DSS (Ariav and Ginzberg,1985:1052), is one example of an area of research that could be termed a *related field*. The substantial body of research that has been done on expert systems for established professionals, has already led to suggestions that such systems might also be *developed for managers* [Blanning (1985:153) referring to, inter alia, Gorry and Krumland (1983) and Reitman (1983)].

DEFINITION.

An *Expert System* (ES) is a problem-solving program that achieves expert levels of performance in a specialized problem domain where tasks normally have no algorithmic solutions [Ford (1985:23); Sowa (1984:280); Gevarter (1983:39)].

Hayes-Roth (1984:28) predicts that the nature of professional work as we know it will dramatically be altered by

developments in the field of ES. Since a professional, as seen by Hayes-Roth, is someone who is paid to make well-informed decisions, the following questions arise:

- (i) Have DSSs as we currently understand them, come to the end of their life cycle?
- (ii) Will DSSs (and the decision-maker) be replaced by automated decision-making systems collectively referred to as Expert Systems?
- (iii) Or will we see a synergistic cross-fertilization of ES and DSS technologies resulting in a new generation of "intelligent" DSSs?

Judging by developments in the ES field where "market demand and new technical advancements are already spawning *intelligent systems* which will *supercede* the more task-specific expert systems" (Hayes-Roth, 1987:86), it seems as if it is only a matter of time before *Expert Decision Support Systems* (EDSS) start emerging. Since an EDSS will perform knowledge processing in addition to information processing, it could be defined as follows:

An *EDSS* is an interactive computer-based information processing system that draws on a large body of modelling, domain-specific and general problem-solving knowledge to help decision-makers solve ill and unstructured problems. It is (i) heuristic, (ii) transparent, and (iii) flexible.

The above definition represents a hybrid between the Keen and Scott Morton (1978:1) definition of DSS and the ES definition suggested by Buchanan and Duda (1983:164).

1.2 PROBLEM STATEMENT.

A survey of the available literature has revealed that research with respect to the application of ES technology to DSS is being carried out in a *fragmented* way. In addition, researchers and system developers, inter alia, Dolk and Konsynski (1984) and Elam and Henderson (1983), are concentrating their efforts on the traditional components of DSS, such as the model- and databases. They are totally ignoring the opportunities being offered by ES to enhance the supporting role of DSS via *new knowledge-based components*.

The above presents a reader with the following two problems:

- (a) Does this "fragmented" approach represent a case of undisguised technology push or is it symptomatic of an underlying deficiency in DSS such as an inability to adapt to changing user needs?
- (b) Would it not be in the interest of the user if the concepts underlying DSS were to be expanded in order to provide a wider range of support?

A case in hand is that of Blanning (1984a, 1984b) who has been propagating the idea of an expert system for managers (ESM) for some time without any apparent success. An ESM is

in essence an ES with DSS features (Blanning, 1984a:491). The problem is that there is an ever-increasing possibility that ES practitioners will heed Blanning's call and develop such systems.

In the author's opinion, however, it would be in the interest of the organizational decision-maker if managerial support is provided from the traditional DSS frame of reference. This opinion is based on the fact that DSS has at its disposal a store of knowledge that has been derived from the development of many successful (and sometimes not so successful) support systems [Bennett (1983); Ginzberg, Reitman and Stohr (1982)]. ES, on the other hand, lacks this expertise.

In order to provide the decision-maker with a support system that incorporates ES features, any one of the following two approaches could be adopted:

- (a) The enhancement route: The traditional DSS components are enhanced with ES features, in which case the current DSS methodological framework should prove to be adequate; or
- (b) The EDSS route: A comprehensive knowledge-based decision support environment is developed. Being a new kind of system, its development will most likely require a new conceptual framework.

Both of these approaches will be examined in the course of this research.

1.3 OBJECTIVES.

The objectives of this research are therefore:

- (a) To establish the adequacy of traditional DSS to adapt to changing user needs; and
- (b) To examine alternative conceptual frameworks of Expert Decision Support Systems.

The task of developing the foundations of a new type of system could be regarded as three-phased. In the *first* phase the needs of the potential user community are determined. If it is found that there is general satisfaction with the level of DSS support in terms of, inter alia, problem-solving capability, input procedures and system availability, then it could be concluded that DSS is in fact successfully adapting to changes in user requirements over time. In order to achieve the first part of the objective, it is therefore only necessary to reach the following subgoal:

SUBGOAL 1: Establish whether DSS is meeting user requirements or whether it exhibits limitations and shortcomings which negatively affect its usability, accessibility and flexibility.

In the *second* phase user needs are mapped to system components. This requires both innovative thought and the ability to recognize the usability of someone else's innovation. Therefore, as a next step towards achieving the stated objective, the following subgoal is set:

SUBGOAL 2: Survey the ES field for possible components for inclusion in EDSS and for tools and techniques that could assist in the construction of knowledge-based components.

The *third* phase in laying the foundations for a new type of system involves interfacing the components. If performed properly, the synergy of their interaction will produce a system in which the whole is more than just the sum of the parts. In the context of this research, phase three implies achieving the following subgoal:

SUBGOAL 3: Develop an architectural framework for a new generation of knowledge-based DSS that incorporates elements of ES.

It was possible to associate a hypothesis with each of the subgoals outlined above. They served as milestones against which progress towards the goal was measured. In a sense they represent what was expected en route.

Hypothesis 1.

DSSs which are developed using traditional concepts and development methodologies are *not meeting* user requirements due to shortcomings in their design and limitations imposed on them by technology and developers.

Hypothesis 2.

The field of ES will prove to be a rich source of tools and techniques to assist in the construction of knowledge-based components for EDSS.

Before stating the last hypothesis, a reader is reminded that the approach whereby *only* the traditional DSS components are enhanced with ES features, was termed the *enhancement route*. A knowledge-based approach to DSS design and development, on the other hand, was labelled the *EDSS route*.

Hypothesis 3.

The EDSS route will be more complete in terms of meeting decision-maker needs than the enhancement route.

1.3 RESEARCH METHODOLOGY.

In conducting this research various methods were followed with literature survey and deductive reasoning the main approach. Since one of the central ideas on which this research hinges, "is provided by the application or transfer of new principles and techniques which have been discovered in other fields" (Beveridge, 1957:129), the transfer method also features prominently. In order to prove hypothesis 3 it was considered necessary to conduct an empirical survey in the form of a Delphi study during the final stages of the research.

Research progressed along the following lines:

- (1) From the literature, substantiating evidence was sought for the hypothesis that traditional DSSs are not fully meeting user requirements and are therefore in need of either technological enhancement or structural revision or both.
- (2) Since DSS and ES are both concerned with the reaching of decisions of some kind or another, those ES tools and techniques that could possibly assist in enhancing the capabilities of DSSs were identified from the available ES literature. Whenever available, the results of empirical studies performed by eminent researchers in the field were used to substantiate the claims made.
- (3) The type of system that will evolve when DSS and ES technologies are merged was then analyzed in terms of characteristics. The possible impact of such systems on decision-makers and their organizational role was also investigated.
- (4) The results of the research up to this point indicated that EDSS users could possibly benefit by the inclusion of some specialized additional components. The literature search was therefore resumed to find supporting arguments for the inclusion of a training, a scenario and a tools component.

(5) The identification of additional components immediately rendered the existing architectural framework for DSS non-usable. Alternative frameworks for the development of EDSSs had therefore to be constructed.

(6) To determine whether:

(a) the identified components were necessary and sufficient for effective EDSS functioning; and

(b) which framework best conveyed the concepts underlying EDSS.

the components and frameworks were presented to a panel of experts for evaluation and comments.

1.4 LIMITATIONS.

Some of the steps outlined in the previous section are concerned with the building and evaluation of alternative frameworks. Since the technology required to support such frameworks is not currently available, it has not been feasible to develop and test any EDSSs empirically. The author was therefore forced to rely on deductive reasoning and the results of a Delphi study to prove his recommendations.

1.5 EXPECTED CONTRIBUTIONS.

The kind of research outlined in this chapter is expected to focus attention and generate discussions on the

real issues in a discipline. In the case of DSS these issues are user-friendliness, flexibility, adaptivity and *decision-maker support*.

By deviating from the traditional data-model-dialog view of DSS the author expects an outcry from the more conservative section of the DSS research community. However, it is also expected that some of the ideas expressed in this thesis will result in further research efforts. One possibility is an empirical survey of DSS users to determine whether they are experiencing the shortcomings and limitations identified in the literature survey phase of this research.

The re-iteration of the fact that researchers should adopt a systemic view of DSS in order to design effective DSSs (Ariav and Ginzberg, 1985), is seen as the main contribution towards the discipline. It is expected that attempts will be made to integrate the diversity of fragmented research efforts in order to create a solid basis for the design of future generation DSSs.

For system builders the research is expected to provide clarity on the structural composition of knowledge-based DSS through identification of system components and their interfaces.

The user, however, is expected to benefit most from this research. EDSSs developed in accordance with the guidelines

in the latter part of this thesis will be better equipped than their DSS counterparts to meet the unique requirements of managerial decision-makers.

1.6 ORGANIZATION OF THESIS.

Although not explicitly divided into separate parts the thesis does have three logical parts.

PART ONE. Chapters 2 and 3 deal with decisions, decision-makers and the support provided by traditional DSS. The rationale behind the inclusion of a discussion of these topics is that to design a system which will truly support decision-makers, the designer must understand decision-makers, their needs and limitations, as well as the strong and weak points of existing support systems. In this part evidence is gathered on which the rejection/acceptance of hypothesis 1 is based.

PART TWO. Chapter 4 starts with a discussion of ES and then compares ES and DSS to illustrate the close relationship between the two technologies. Those ES features that could contribute meaningfully towards the development of more useful DSSs are investigated in some detail. The characteristics of the type of system that will result when DSS is enhanced with ES techniques, are discussed in Chapter 5 together with the ideal view that users should take of such systems. The expected impact of EDSS on the organizational

role of the human decision-maker is also discussed. This part produces the facts on which the decision whether hypothesis 2 should be accepted or rejected, is based.

PART THREE. This part has a *theoretical* and an *empirical* section, both of which are intended to contribute towards making a decision on whether hypothesis 3 should be accepted or rejected. In the *theoretical* section, Chapter 6 is devoted to the identification of components that should be considered for inclusion in EDSS followed by an arrangement of these components in two architectural frameworks in Chapter 7. In a discussion of the modular framework (the other being a generic framework) it is pointed out how systems based on the framework, will assist in alleviating some of the problems surrounding traditional DSS.

In the *empirical* section (Chapter 8) a reader is presented with a discussion of the results of a Delphi study that was conducted to test the necessity and sufficiency of a set of EDSS components. Also tested was the acceptability of a number of architectural frameworks. Both, components and frameworks, were identified and/or developed in the theoretical section described above.

Finally, Chapter 9 presents a reader with a summary of the highlights of the research, some remaining problems, directives for future research and a final conclusion.

CHAPTER 2.

DECISION-MAKING AND DECISION SUPPORT.

2.1 INTRODUCTION

One of the objectives of this thesis is to find better ways in which to assist decision-makers in making more effective decisions during the decision-making process. A prerequisite to achieving this goal is an in-depth understanding of decisions and the decision-making process as well as the characteristics, needs and problems of decision-makers (Remus and Kotteman, 1986:403). What is currently being done to support decision-makers should also be critically evaluated.

In this chapter, therefore, the focus is on (1) decisions and the decision-making process, (2) the decision-maker and (3) systems that provide decision support. This will form the basis for subsequent analysis and tests of the Expert System/Decision Support System synthesis.

2.2 DECISIONS AND DECISION-MAKING.

2.2.1 DECISIONS.

An abundance of definitions for the term "decision" is encountered in the literature [(Ofstad, 1961); (Shull, Delbecq and Cummings, 1970); (Simon, 1960); (Emory and Niland, 1968)]. Most of these definitions indicate that "the decision-maker has several alternatives and that his

choice involves a comparison between these alternatives and an evaluation of their outcome." (Eilon, 1969).

To the author it appears as if the definition by Harrison (1981:3) best captures the essence of a decision and it is therefore adopted for use in further discussions:

"A decision is defined as a moment, in an ongoing process of evaluating alternatives for meeting an objective, at which expectations about a particular course of action impel the decision-maker to select that course of action most likely to result in attaining the objective."

Over the years many different types or classes of decisions have been identified by researchers, some of which are presented in Figure 2.1. According to Harrison (1981:12) all of these schemes can be reduced to two basic categories: (1) routine, recurring and certain, and (2) non-routine, non-recurring and uncertain. Figure 2.2 shows all the classes divided into the two categories labelled Category I and Category II Decisions by Harrison. It is pointed out by Radford (1981:8) that authors have tended towards the use of "well-structured" and "ill- or unstructured" for Category I and Category II decisions respectively.

In practice, however, a continuum of decision situations exists with those lying somewhere in the middle of the spectrum being referred to as "semi-structured" decisions

AUTHOR AND TYPE	DESCRIPTION
<u>SIMON (1960)</u>	
PROGRAMMED	REPETITIVE AND ROUTINE, A PROCEDURE FOR HANDLING SUCH DECISIONS EXISTS.
NON-PROGRAMMED	NOVEL, UNSTRUCTURED, CONSEQUENTIAL. PRECISE NATURE IS COMPLEX, ELUSIVE.
<u>DRUCKER (1976)</u>	
GENERIC	SAME AS "PROGRAMMED" ABOVE
UNIQUE	SAME AS "NON-PROGRAMMED"
<u>DELBECQ (1967)</u>	
ROUTINE	AGREED-UPON GOAL AND TECHNOLOGIES EXIST TO ACHIEVE GOAL.
CREATIVE	NO AGREED-UPON METHOD OF DEALING WITH PROBLEM.
NEGOTIATED	OPPOSING FACTIONS CONFRONT EACH OTHER CONCERNING EITHER MEANS OR ENDS OR BOTH.
<u>GORE (1962)</u>	
ROUTINE	AS ABOVE.
ADAPTIVE	PROBLEMS OF NON-RECURRING NATURE
INNOVATIVE	LEADS TO CHANGES IN GOALS, POLICIES AND PURPOSES.

Figure 2.1

Types of Decisions.

[A summary of Harrison (1981:10-11)]

(Gorry and Scott Morton, 1971:1). For these decisions only part of, the decision process can be fully specified and a synthesis of human judgement and computer capabilities seems to be the best strategy in dealing with them (Radford, 1981:9). Early researchers and practitioners in the decision support field concentrated on assisting managers in dealing with this class of decisions.

2.2.2 DECISION-MAKING.

Decision-making is an integral part of the management of any organization. Simon (1977a:39) even takes the position that decision-making is synonymous with management. Although this is strongly put, it can be shown that many of the basic functions performed by managers, such as planning, organizing, staffing, directing, and controlling [(Koontz and O'Donnell, 1972); (Thierauf, 1984:5)] involve decision-making in one form or another. According to Harrison (1981:1) competence in decision-making differentiates the effective manager from the ineffective manager. This explains why organizational decision-making has frequently been the object of research.

The branch of scientific research that studies the decision-making process is known as Decision Analysis or Decision Theory. It has its roots in the classical model of the so-called "homo economicus" (economic man), the representation of the purely rational decision behavior of economic man (McGuire, 1964). As was the case with decisions, efforts to

describe the nature of decision-making can also be classified into various categories. Some researchers follow what is known as the "process" approach as opposed to the so called "content" approach (Behling and Striesheim, 1976). *Process* frameworks deal with the phases or stages of decision-making while the *content* approach is more concerned with the characteristics of resultant decisions or the nature of the decision-maker (Bonczek, Holsapple and Whinston, 1984:143). In the author's opinion this dichotomy does not fully describe the total spectrum of approaches. A third category in which the focus is more on the interaction between the decision-maker and the decision-making environment, is therefore suggested. It is labelled the "behavioural approach" in the discussion that follows.

(a) PROCESS-BASED DECISION-MAKING MODELS.

Two process-based models that are frequently referred to in the literature are the Rational and Bounded Rational Models. A short discussion of each follows.

(i) Rational Model.

Traditionally, analysis of the decision-making process was approached in a highly analytic-rational way. It was assumed that the decision-maker had complete information about the decision situation; he supposedly knew all the decision alternatives from which he could choose and had complete information about the profit that each alternative would offer him in each situation. The objective of the decision-maker was unquestionably that of maximizing profit

(Kickert, 1980:38). A similar approach is presented by Bahl and Hunt (1985:83) who describe a type R model and Thierauf (1984:82-85) with his description of the so-called quantitative approach.

In this classical, rational approach three different types of decision-making processes were distinguished. They are decision-making under (1) certainty, (2) risk, and (3) uncertainty. All three types, however, still assume that the set of possible actions, as well as the set of possible states and a preference ordering over states are known by the decision-maker. Risk implies that the decision-maker does not know exactly what state will occur but only knows a probability distribution over the possible occurring states. In the absence of knowledge of probabilities of occurrence, one speaks of uncertainty.

(ii) Bounded Rationality Model.

In 1945 Simon queried the validity of the rational approach on the basis of his theory of limited human cognitive capabilities [(Simon, 1945); (March and Simon, 1958)]. The two main theses of the concept of bounded rationality are:

- (1) alternative actions and effects are not given but have to be discovered and developed after a search process;
- (2) no optimal decision will be taken, but the decision-maker will be content with satisficing solutions.

(Kickert, 1980:44)

The word "satisficing" means finding and selecting a satisfactory alternative as opposed to the best one (Thierauf, 1984:86).

In their article on problem-solving strategies for DSS design, Bahl and Hunt (1985:84) refer to this approach to decision-making as the type B model whereas Thierauf (1984:86-89) calls it the descriptive or decision-centered approach. According to them, it is virtually impossible for managers who are confronted with complex information to make perfect decisions mainly because of limited mental information processing capacities, uncertainty, time pressures and individual biases.

After the introduction of the concept of bounded rationality, decision-making theory gradually evolved from an analytical rational theory towards a theory of cognitive learning processes (Kirsch, 1977) which in turn has led to the taking over of the phase scheme that is characteristic of problem-solving (Kickert, 1980:53).

Various views of the decision-making process based on the phased approach are presented in Figure 2.3. The majority of these views are strongly related to the three problem-solving stages:

(a) What is the problem? (b) What are the possible solutions? and (c) Which solution is the best?

Furthermore, a quick comparison reveals that all frameworks

SIMON (1960): 1. Intelligence: finding occasions for making a decision.
2. Design : finding possible courses of action.
3. Choice : choosing among courses of action.


WITTE (1972): 1. Information gathering.
2. Development of alternatives.
3. Evaluation of alternatives.
4. Choice.

SCHRENK (1969): 1. Problem recognition.
2. Problem diagnosis.
3. Action selection.

BRIM, GLASS, LAVIN and GOODMAN (1962):
1. Problem identification.
2. Information gathering.
3. Development of possible solutions.
4. Evaluation of these solutions.
5. Selection of a strategy for performance.
6. Actual performance of an action.

HARRISON (1980): 1. Setting managerial objectives.
2. Searching for alternatives.
3. Comparing and evaluating alternatives.
4. The act of choice.
5. Implementing the decision.
6. Follow-up and control.

Figure 2.3
Models of the Decision-making Process. (From various sources.)



	CATEGORY I	CATEGORY II
CLASSIFICATIONS	Programmable; routine; generic; negotiated; computational; compromise.	nonprogrammable; judgemental; creative; adaptive; innovative; inspirational; unique.
STRUCTURE	Procedural; predictable; recurring; Certainty regarding cause/ effect relationships; Within existing technologies; Well-defined information channels; Definite decision criteria; Outcome preferences be certain or uncertain.	Novel; unstructured; non-recurring; consequential; elusive and complex; uncertain cause/effect relationships; Information channels undefined; Incomplete information; Decision criteria may be unknown; Outcome preferences may be certain or uncertain.
STRATEGY	Reliance upon rules and principles; Habitual reactions; Prefabricated response; Uniform processing; Computational techniques and accepted methods for handling.	Reliance on judgement, intuition and creativity; Individual processing; Heuristic problem-solving techniques; Rules of thumb; General problem-solving processes.

Figure 2.2

A Categorization of Decision Characteristics.

Harrison (1981:12)



include the three basic stages outlined by Simon. The adoption of the phased approach has led to the formulation of definitions of the decision-making process of which the definition by Thierauf (1984:80) is an example:

The decision-making process is "a series of steps that start with an analysis of the information and ultimately culminate in a resolution."

(b) THE CONTENTS APPROACH.

An example of the content approach is provided by Holsapple and Moskowitz (1980). This framework identifies a set of seven mutually independent and collectively complete capabilities which must be possessed by a decision-making system. They are the abilities to collect information, to formulate, to govern, to analyse, to recognize problems and to implement (Bonczek, Holsapple and Whinston, 1984:144). Bonczek et al. (1984) claim that by exercising these abilities in various configurations and sequences, a variety of process models can be generated.

(c) BEHAVIOURALLY ORIENTED MODELS.

It is generally accepted that decision-making is a dynamic, interrelated process [(Harrison, 1981:24-28); (Radford, 1981:9); (Sprague and Carlson, 1982:110)]. The individual phases are not exercised in isolation but form an integral part of a larger entity (Wadia, 1966). The different stages or phases are not necessarily executed in

strict sequential fashion but follow each other in an arbitrary way depending on endogenous or environmental situations determined by political factors, conflicts, available information and so on [(Kickert, 1980:54); (Mintzberg, Raisinghani and Theoret, 1976)]. Furthermore, in a study of 233 cases of decision-making Witte (1972) has found that the activities of the different phases were distributed over the total duration of the process. It would therefore seem as if decision-making is not as orderly a process as advocates of the phased approach make it out to be.

What follows is a short discussion of models that take the above arguments into consideration. Some of them also describe decision-making as a sequence of phases but the phases are not derived from the cognitive problem-solving process. For example, Cyert and March (1963) propose that organizational decision-making should be viewed as being divided into four types of operations:

- (1) quasi-resolution of conflict;
- (2) uncertainty avoidance;
- (3) problem-oriented search; and
- (4) organizational learning.

(i) Garbage Can Model.

In their alternative to the phased approach Cohen, March and Olsen (1972) propose a behavioural theory of organized anarchy which contains four elements:

model). What should be obvious, however, is that the decision-maker will find him/herself frequently in positions typified by more than one model, as Huber (1981:5) concludes:

"The great majority of significant organizational decision settings have their rational aspects, political aspects, predictable process aspects, and chance aspects."

2.3 THE DECISION-MAKER.

The decision-maker is the human actor in the decision-making process who, in order to reach a meaningful decision, must fuse, integrate and coordinate problem-relevant information with applicable statistical and mathematical models as well as with intuition, judgement and experience. In addition to the basic tasks of problem recognition, the invention, development and analysis of possible courses of action, and the evaluation of alternatives, other seldomly-mentioned tasks which the decision-maker must perform during the decision-making process are the following:

- (a) Selection of relevant data.
- (b) Aggregation of data into totals, averages, and frequency distributions.
- (c) Estimation of the parameters in a probability distribution (by performing statistical analyses of data to determine relationships between important variables).
- (d) Simulation to calculate the anticipated consequences

organizational decisions are viewed as products of negotiation and social interaction.

(iii) Programming Model.

In the Programming Model decisions are regarded as the consequences of the programs (standard operating procedures, group norms and budget limitations) and the programming (professional training, reinforcement of past decision-related behaviour) of the units involved (Huber, 1981:5). In this type of organizational decision setting, decision-making behaviour at a specific point in time is largely predictable from past behaviour of the decision-maker. This model probably has its roots in the very similar Organizational Process Model of Simon (1976) who, according to Bahl et al. (1985:82), postulated that organizational units rely on their standard operating procedures or past practices for responding to problems.

What could be concluded from the preceding discussion of the decision-making process is that it is a "highly variable affair" (Bahl and Hunt, 1985:82). Often it is not orderly, much less rational than once thought and subject to endogenous and exogenous influences that are hard to identify, anticipate or control. Another impression is that there is no consensus on what the focal point should be: the decision (rational model), the decision-maker (bounded rationality model), social interaction (political model), time and chance (garbage can model) or functions (content

(1) problems, (2) solutions, (3) participants, and
(4) opportunities for choice. In their view:

"an organisation is a collection of choices looking for problems, issues and feelings looking for decision situations in which they may be aired, solutions looking for issues to which they might be the answer and decision-makers looking for work." (Cohen et al., 1972:2)

This so-called "Garbage Can Model" of the decision-making process could be interpreted as an antithesis to the usual phased model approach (Kickert, 1980:56) and is called type N (for Non-rational) models by Bahl et al. (1985:85). The contribution that this model makes is that it emphasizes the role of chance and timing in the organizational decision-making process.

(ii) Political/Competitive Model.

This alternative to the phased approach portrays an environment where organizational decisions are consequences of the application of strategies and tactics by units seeking to influence decision processes in directions that will result in choices favorable to themselves (Huber, 1981:3). This type of decision-making is especially noticeable in group decision-making situations and is covered extensively by authors such as March (1962), Lerner (1976) and Pettigrew (1972). Bahl et al. (1985:84) refer to this as the type F (for Functional) model in which

of proposed decisions and/or possible changes in the corporate environment.

- (e) Optimization to determine decisions that will maximize or minimize a single measure of performance or cost without violating constraints on other such measures (Blanning, 1979:88).

However, the psychological and intellectual characteristics of decision-makers are such that they (the decision-makers):

- (a) are imperfect in the way in which they process information (Benbasat and Taylor, 1982). That is, they combine multiple cues in a far less sophisticated way than is warranted by the complexity of the problems they face, they frequently do a poor job of making subjective probability estimates, often violating the laws of probability theory and they downplay data which contradicts their views [(Remus and Kottmann, 1986:406); (Blanning, 1984a:493)].
- (b) have individual personality differences (Zmud, 1979). One such difference is in cognitive style, which describes the way in which they acquire and process information. Others relate to risk aversion and ambiguity tolerance [(Blanning, 1984a:493); (Remus and Kottmann, 1986:405)].

It is therefore to be expected that decision-makers will experience problems which may be compounded by, inter alia,

their unique information requirements and the type of support they receive from professional DP people and Operations Research / Management Science / Decision Analysis practitioners. What follows is a short discussion of some of the problems facing decision-makers.

(a) DECISION ORIENTED PROBLEMS.

Decision-makers have unique information requirements mainly in category II (non-routine, non-recurring and uncertain) decisions identified in section 2.2. According to Radford (1981:7) and Harrison (1981:12) no complete and well-established procedures exist for dealing with such problems. The available data is usually incomplete and ill-defined and the ways of dealing with situations of this nature are not usually agreed upon by all concerned. A further problem is the lack of quantitative measures that truly and completely describe the costs and benefits of the available alternatives (Radford, 1981:16).

(b) DECISION-MAKER ORIENTED PROBLEMS.

Working from the assumption that decision-makers do have limited cognitive capabilities, it is logical to assume that they will experience problems when faced with situations in which "the inputs are extremely complex, the outcomes uncertain and circumstances change rapidly" (Stephenson and Stephenson, 1983:391). One such problem is identified by Radford (1978:144) as being the failure to recognize the significance of information especially when

such information is encountered in disordered fragments intermingled with large amounts of other information.

Decision-makers also experience serious behavioural problems such as the misinterpretation or misperception of the contents of information acquired. According to Holsti (1971:30) decision-makers, especially those under stress, tend to ignore, re-interpret or reconstruct information that is not in accordance with their subjective convictions. Further examples are the "screening out" of information which is not in a form acceptable to the decision-maker (Radford, 1978:146) and inconsistent use of heuristics (Remus and Kottmann, 1986:406).

(c) PROBLEMS RELATED TO SUPPORTING GROUPS.

A limited number of functional groups have traditionally provided support to decision-makers. Amongst these are Data Processing (DP), Management Information Systems (MIS), Decision Analysis (DA), Management Science (MS) and Operations Research (OR).

In MIS the accent is on the provision of timely, accurate and meaningful information whereas in DA the focus is on the development of normative decision aids, called decision analysis models, to step a user through an axiomatically correct process for using knowledge to solve a problem (Lehner, Probus and Donnell, 1985:471). MS/OR can be considered as being the application of scientific methods by

interdisciplinary teams to problems involving the control of organized man-machine systems so as to provide solutions which best serve the purposes of the organization as a whole (Ackoff and Sasieni, 1968:6).

(i) In evaluating the support provided by traditional DP/MIS one often repeated conclusion is that typified by Andriole (1985:12):

"Over and over again complaints are heard today about how difficult it is to "find" data once it has been entered into a data base management system".

Andriole underlines the point made above by stating that it often takes a lot of time to satisfy requests for information, causing key information to be excluded from periods of corporate crisis management. This problem is related to the misconception that managers suffer from lack of information whereas in fact they suffer from an information overload (Ackoff and Sasieni, 1968:440).

Watson and Hill (1983:86) provide the following reasons why many organizations' MIS have been unsuccessful in supporting semi- and unstructured decision-making tasks:

- (i) MIS personnel are unfamiliar with the decisions that need support;
- (ii) MIS personnel are overworked with backlogs measured in years;
- (iii) Managers cannot specify their information

	Integrated Management Information System	Decision Maker Requirements
Type of system.	Backward-looking control system.	Forward- and backward looking planning and control system.
Reports prepared.	Output reports directed to all levels of management for past operations.	Output reports directed to all levels of mgmt. and some ops. personnel for past, current and future operations.
Exception reporting.	Management exception reports.	Future plans are tied in with "mgmt. by perception". Current plans and budgets are related to mgmt. exception reports.
Information orientation.	Output oriented.	Input/output oriented with graphics CRT.
Processing mode.	Mainly batch mode.	Interactive processing where user retains contr.
Data elements.	Common data bank.	Enlarged data bases.
Types of files.	Sequential and random access file storage.	Greater accent on random, on-line file storage.
Mathematical models.	Limited use of standard operations research models.	Great use of both standard and complex operations research models.

Figure 2.4

MIS characteristics vs Decision-maker needs.

[Adapted from Thierauf (1984:63)]

requirements except through an interactive process; and

- (iv) Managers' information requirements are subject to changes in the decision-making environment.

The reasons why MIS failed to provide the support required by organizational decision-makers are highlighted when the essential characteristics of MIS are contrasted with the requirements of decision-makers. This has been done in Figure 2.4 from which it can be seen that whereas management information systems are mainly backward-looking systems that produce exception reports in batch processing mode decision-makers require control over interactive systems which provide them with powerful modelling capabilities.

(ii) As far as Decision Analysis is concerned, decision aids have been developed that use decision models as prescriptive problem representations to help guide users through the decision-making process. These decision aids, however, require users to subjectively assign values to the bottom-level factors. In many instances this is problematic, since a common by-product of problem structuring via DA models is defining a set of primitive factors different from the set of problem elements normally perceived by the user. The user is therefore required to translate his/her perception of the problem into the input requirements of the DA-based decision aid (Lehner, Probus and Donnel, 1985:471).

(iii) With regards to the support forthcoming from Operations Research Ackoff (1979) states that researchers in this field have failed to respond to the need for decision-making systems that can learn and adapt quickly and effectively in rapidly changing situations. As a consequence, the application of OR is increasingly restricted to those problems that are relatively insensitive to their environments. This view is shared by Holmes (1985:5) when he says:

"... the early promise of Operations Research failed to be realized because it was too esoteric and theoretical to meet the day-to-day decision-making needs of management."

Since a large number of organizations are "open" systems (that is, interacting with their environments), the application of OR to decision-making in such organizations is limited to structured decision situations.

Management Science/Operations Research insists on formal structures (Vazsonyi, 1978), which makes it less useful for semi- and unstructured decisions. It should, however, be kept in mind that research in this area has made great progress in modelling and optimization (Watson and Hill, 1983:83), offering great potential benefits to the decision-maker. For example, MS models can help to externalize multiple objectives or to force the user to make decision criteria more explicit (Henderson and Schilling, 1985:157-162).

Because of the problems outlined above a new area of great activity within the information systems field emerged. The term Decision Support Systems (DSS) became a rallying cry for researchers, practitioners and managers concerned that MS and MIS have become unnecessarily narrow in focus (Keen, 1980a:1). The objectives of these new systems were to extend the decision-maker's capabilities without replacing his/her judgement and to support the manager in those decisions where judgement was required (Ginzberg and Stohr, 1982:9).

2.4 DECISION SUPPORT SYSTEMS.

2.4.1 DEFINITION AND CHARACTERISTICS.

(a) DEFINITION.

For a number of years after the conception of DSS there was little consensus about what qualifies a system as a DSS with the result that many definitions were suggested [(Gorry and Scott-Morton, 1971); (Little, 1970); (Alter, 1980); (Moore and Chang, 1980); (Bonczek, Holsapple and Whinston, 1980); (Keen, 1980)]. A tendency existed to give a brief definition augmented by a list of characteristics that a system should display in order to be classified as a DSS. For example, Little (1970:B470) defines a DSS as a "model-based set of procedures for processing data and judgement to assist a manager in his decision-making" and then extends it by stating that the system should be simple, robust, easy to control, adaptive, complete on important issues and easy to communicate with.

After critically examining a number of definitions while focussing on the issues highlighted by and ignored by each, Ginzberg and Stohr (1981:12) conclude that supporting and improving decision-making is *the* issue in DSS. They then propose the following definition which is quite close to the early definition of Gory and Scott-Morton:

"A DSS is a computer-based information system used to support decision-making activities in situations where it is not possible or desirable to have an automated system perform the entire decision process."

(b) CHARACTERISTICS.

The term 'characteristics' as used in this section, indicates those features that are regarded as essential to DSS. That is, they distinguish DSS from all other systems and, as was done by Keen and Scott Morton (1978:1), could be quoted instead of a formal DSS definition. The most notable of these characteristics are:

- (i) DSS assist managers in their decision processes in semi-structured tasks;
- (ii) DSS support, rather than replace, managerial judgement;
- (iii) DSS improve the effectiveness of decision-making rather than its efficiency.

Ever since the concept of DSS was introduced researchers have kept themselves busy with devising (determining)

capabilities that a system should have in order to qualify as DSS. With regards to these performance requirements, however, it should be noted that:

- (i) there is no universal set of capabilities for all DSS; the criteria for any particular DSS will be entirely dependent on the task, the organizational environment and the decision-maker(s) involved (Sprague and Carlson, 1982:29);
- (ii) if a particular system possesses the identified capabilities it does not automatically qualify as a DSS; it is only when specific components with special capabilities are combined in a particular way that a system manifests itself as a DSS.

Consequently, a multitude of DSS capabilities/requirements have been suggested by researchers. Since each of these could be seen as contributing towards the realization of one of the fundamental DSS characteristics, they are presented below in four separate groups. For the sake of readability only one reference per capability is included.

CAPABILITIES TO ASSIST DECISION-MAKERS IN SEMI-STRUCTURED TASKS.

- (i) Comprehensive with regard to the types of decision-making processes supported and/or with regard to the functions it can perform (Blanning, 1979:88).
- (ii) A structure which is understood by the manager, reflecting his own way of thinking (Wagner, 1981:78).
- (iii) Support all types of decisions such as independent and

interdependent decisions (Sprague and Carlson, 1982:29).

CAPABILITIES TO SUPPORT, NOT REPLACE, MANAGERIAL JUDGEMENT.

- (i) Output directed to organizational personnel at all levels (Thierauf, 1982:72).
- (ii) Adaptive system over time (Keen, 1980a:9).
- (iii) Flexible to handle varied situations (Keen, 1981:4).
- (iv) Responsiveness (Keen, 1981:4).
- (v) User-initiated and controlled (Watson and Hill, 1983:83).
- (vi) Modular and extensible over time (Moore and Chang, 1980).

CAPABILITIES TO IMPROVE DECISION-MAKER EFFECTIVENESS.

- (i) Utilization of appropriate mathematical and statistical models (Thierauf, 1982:67).
- (ii) Query capabilities to obtain information by request (Thierauf, 1982:70).
- (iii) Comprehensive database (Thierauf, 1982:74).
- (iv) Easy-to-use approach (Keen, 1981:4).
- (v) Simplified man-machine interface, alternative modes of presentation (Henderson and Schilling, 1985:166).
- (vi) Interactive or allow two-way communication (Turner, 1980).

OTHER CAPABILITIES.

- (i) Support ad hoc and exploratory data analysis and decision modelling (Bonczek, Holsapple and Whinston, 1982:62).
- (ii) Allow usage at irregular, unplanned intervals (Moore and Chang, 1980).
- (iii) Allow usage patterns to develop over time (Keen, 1980b).

Some of the capabilities listed above have been claimed to be characteristic of DSS. For example, Keen (1980a:9) states that:

"The label 'Support System' is meaningful only in situations where the final system must emerge through an adaptive process of design and usage."

Ginzberg and Stohr (1982:21), however, point out that care must be taken not to ascribe to all DSS something which is characteristic only of some.

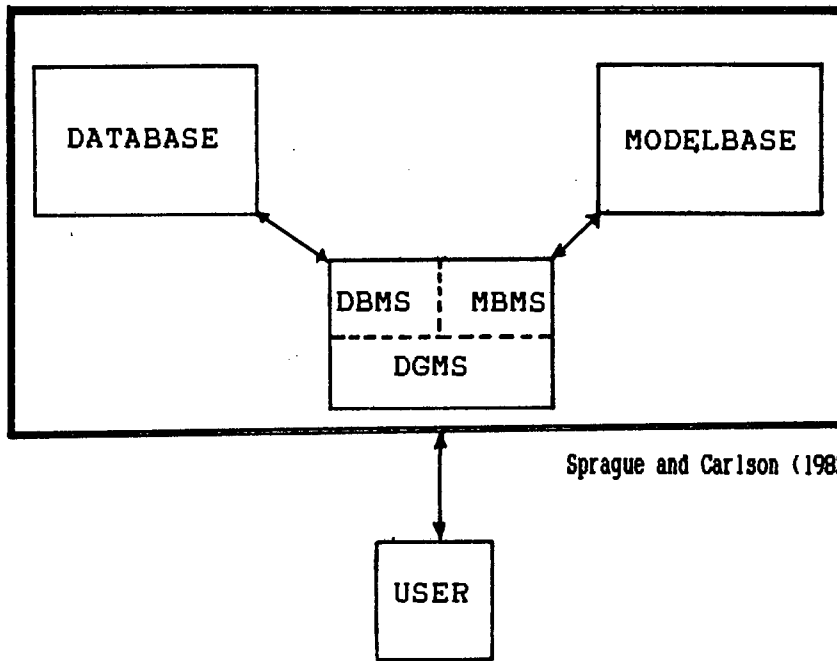
2.4.2 ARCHITECTURAL FRAMEWORK OF DSS.

"The emerging theory of DSS addresses a blend of design strategy, system characteristics, and required technological building blocks."

(Henderson and Schilling, 1985:167)

The architectural framework put forward by Sprague and Carlson (1982:33) and depicted in Figure 2.5, is typical of

The DSS



Sprague and Carlson (1982:33)

Figure 2.5

Components of DSS

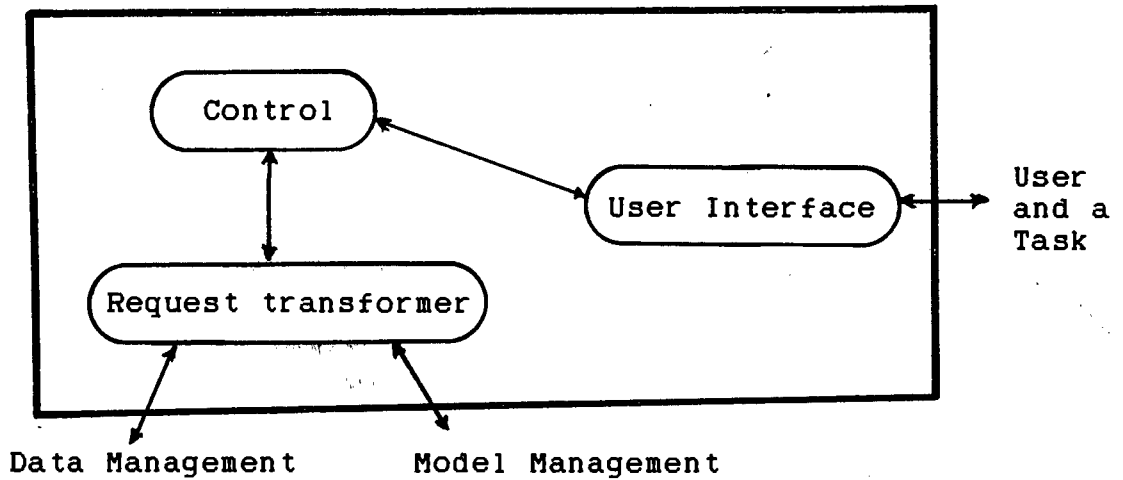


Figure 2.6

DSS Dialog Management

Ariv and Ginzberg (1985:1048)

the majority of designs found in the literature [(Wang and Courtney, 1984:704); (Minch and Burns, 1983); (Ariav and Ginzberg, 1985); (Buede, Yates and Weaver, 1985:465); (Thierauf, 1982:70)]. According to this framework three major components are necessary for a DSS: (i) a dialog subsystem (ii) a database component and (iii) a modelling subsystem.

(i) The Dialog Subsystem.

Much of the power, flexibility, and usability characteristics of a DSS derive from capabilities in the interaction between the system and the user, which Sprague et al. (1982) call the dialog subsystem. It is the hardware and software that provide the user interface to the system and should ideally be able to:

- (i) handle a variety of dialog styles;
- (ii) accommodate user actions with a variety of devices;
- (iii) present data with a variety of formats and devices and
- (iv) provide flexible support for the user's level of knowledge (Sprague et al., 1982:34-35).

The dialog component represents the framework in which outputs are presented as well as the context for user inputs and according to Ariav and Ginzberg (1985:1047) this suggests three necessary dialog management sub-components (Figure 2.6):

- (i) a user interface to handle the syntactic aspects of the interaction;
- (ii) a dialog control function to determine the basic semantics of interactions; and
- (iii) a request transformer to provide the necessary translations between the users' vocabulary and the system's internal modelling and data access vocabulary.

(ii) The Database Component.

Typical of a DSS is the much richer set of data sources than are usually found in non-DSS applications (Sprague et al., 1982:35). A DSS database contains, inter alia, data from internal and external sources, transactional as well as non-transactional data, accounting and non-accounting data. This diversity has led to the practice of creating DSS-specific databases logically separate from other operational databases which in turn facilitates rapid additions and changes in response to unanticipated user requests [(Sprague et al., 1982:36)]; (Hirouchi and Kosaka, 1984)].

Ideally the database subsystem should be able to:

- (i) combine a variety of data sources through a data capture and extraction process;
- (ii) add and delete data sources quickly and easily;
- (iii) portray logical data structures in user terms;
- (iv) handle personal and unofficial data so that the user can experiment with alternatives based on

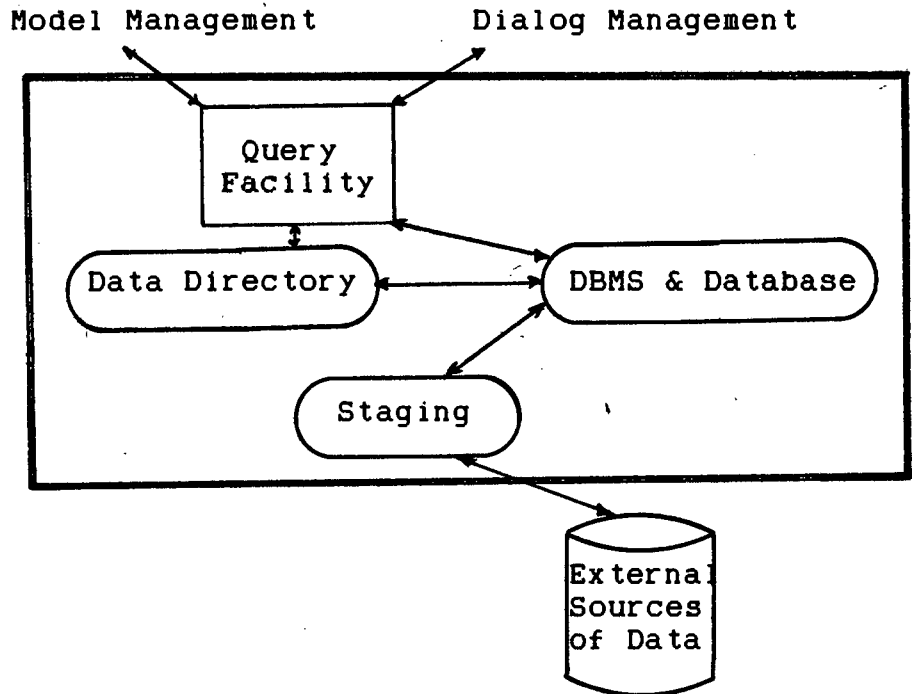


Figure 2.7

DSS Data Management

Ariav and Ginzberg (1985:1048)

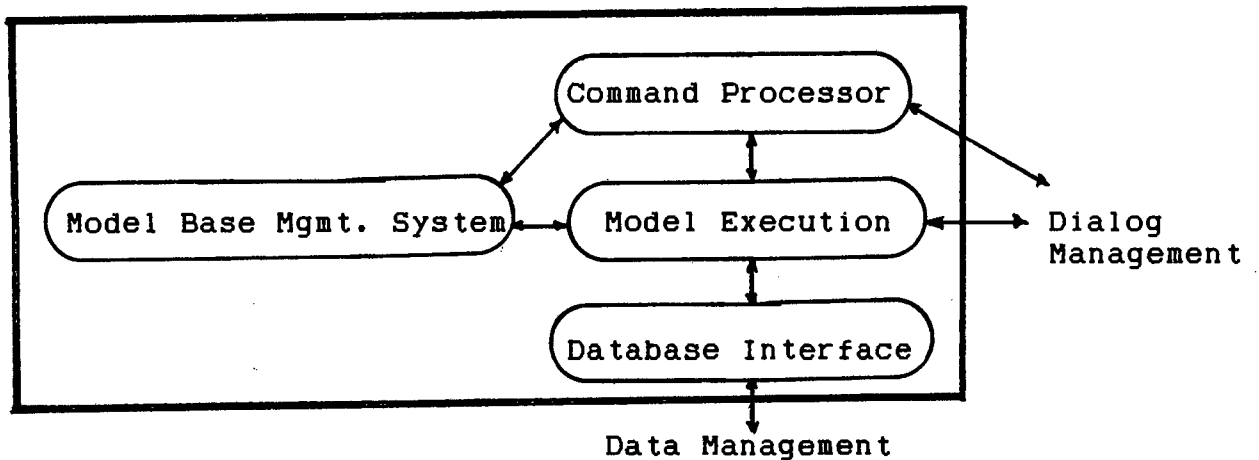


Figure 2.8

DSS Model Management

Ariav and Ginzberg (1985:1049)

Management Science/Operations Research and Information Systems activities [(Chervany and Perkins,1975); (Scott, 1978)].

According to Ariav et al. (1985:1048) this component is characterized by its ability to invoke, run, change, combine and inspect models. Sprague and Carlson (1982:37) on the other hand suggest that the key capabilities are:

- (i) the ability to create new models quickly and easily;
- (ii) the ability to access and integrate model building blocks;
- (iii) the ability to catalog and maintain a wide range of models;
- (iv) the ability to interrelate these models with appropriate linkages in the database; and
- (v) the ability to manage the model base with management functions analogous to data base management.

In order to provide the functions outlined above Ariav et al. (1985:1048) suggest that the modelling component of a DSS should include the following sub-components (Figure 2.8):

- (i) a model-base management system to generate, retrieve and update parameters, to restructure models, and to include a model directory for maintaining information about available models;

- (ii) a model execution component to control the actual running of the model and to link models together when integration is needed;
- (iii) a modelling command processor to accept and interpret modelling instructions; and
- (vi) a database interface to retrieve data items from the database for running models and, eventually, to store model outputs in the database for further processing, perusal or input to other models.

The majority of DSS characteristics/capabilities listed in the previous section can easily be identified from the foregoing discussion of the components of a DSS. However, the list will not be complete since the synergistic effect of combining the individual components into an integrated whole, is to give the system additional (and sometimes unanticipated) features.

2.5 CONCLUSION.

Organizational decision-makers are faced with problems that range from structured to unstructured. The approaches to solving these problems differ from organization to organization and from decision-maker to decision-maker with the rational, bounded-rational, political/competitive, programming and garbage can models amongst the more well-known approaches.

Decision-makers have to perform a wide variety of tasks in order to reach a decision, for example gather and validate data, identify and quantify objectives, diagnose and structure the problem, generate and assign values to alternatives. However, their psychological and intellectual make-up is such that they often experience problems in arriving at optimal and/or objective decisions.

MIS and OR/MS/DA provided (and still are providing) some form of support to decision-makers to overcome, for example, their limited cognitive capabilities. However, specially tailored systems that could support decision-makers in solving semi-structured problems effectively proved to be necessary. This need led to the emergence of Decision Support Systems (DSS) as a discipline separate from those mentioned above.

Whereas this chapter has described the components, functions and characteristics of a DSS as a foundation for further analysis the next chapter is dedicated to highlighting those DSS objectives that are not fully met by current systems.

CHAPTER 3.

THE NEED FOR CHANGE.

3.1 INTRODUCTION.

In comparison with fields such as EDP and MIS, DSS is a fairly newcomer to the scene. Since its conception in the early 1970's there have been rapid international developments, not only in the field of DSS but also in many other computer oriented fields. It behoves researchers and practitioners alike to periodically assess where the field stands and where it is heading. Accordingly, this chapter will attempt to identify problems within the DSS field and to suggest from which direction assistance might be forthcoming. Problems internal to DSSs are covered first, followed by a discussion of some problems in the DSS environment that needs addressing.

The shortcomings identified and discussed in this chapter are *firstly* shortcomings that are not inherent to DSS but which are the result of the diversity of interpretations given to DSS concepts by researchers and practitioners. For example, a preoccupation with support could lead to the design and development of management, executive, personal or mind support systems instead of decision support systems (Stabell, 1983:223). *Secondly*, it is argued that the range of support provided could and should be broadened to cater for the needs of a changing business environment. For

example, DSS has a definite role to play in supporting company executives in their search for strategic uses of computers to give their companies a competitive edge (EDP Analyzer, April, 1986).

3.2 SHORTCOMINGS AND LIMITATIONS OF DSS.

The detailed discussion of DSS shortcomings and limitations that follows identifies a number of areas that are in need of research. The problem areas are:

- * Passiveness.
- * Limited support.
- * Inflexibility.
- * User interface problems.

3.2.1 PASSIVENESS.

Undoubtedly there exist many DSSs that function efficiently and in doing so, are increasing the effectiveness of decision-makers. An example is the very successful Geodata Analysis and Display System (GADS) which has been used for different problem areas such as urban planning, customer engineer territory planning and placement of police personnel on beats (Sprague and Carlson, 1982:45-58). PMS (Portfolio Management System), BRANDAID and IMS (Interactive Marketing System) are further examples (Keen, 1981:2). The question, however, is whether some of these systems could not perform equally well (or even better) *with less demands on the human user in terms of time and intellectual and judgemental effort.* A hardware parallel would be the

computer terminal which, when introduced, greatly improved the productivity of information workers. Gradually these "dumb" terminals were given more capabilities until eventually they could be regarded as "smart" terminals boasting local processing power which enables them to function in collaboration with a mainframe or independently.

Three possible areas in which DSS could be given a more active role to play are (i) special circumstances, (ii) system transparency and (iii) user and machine learning.

(i) SPECIAL CIRCUMSTANCES.

A corner-stone upon which the majority of DSSs are based, is the seminal work by Keen and Scott Morton (1979) in which it is proposed that a DSS should "support, rather than replace, managerial judgement." It is argued here that if this guideline is followed too closely DSSs will be forced into the role of passive assistants in the decision-making process (keeping a dumb terminal dumb). They will blindly follow a fixed set of instructions and will have no way of adapting to changing circumstances (Sowa, 1984:278). Knowledge-based DSSs, on the other hand, will apply their knowledge acquired from human experts in the field in novel ways in different circumstances. Ginzberg and Stohr (1982:12), while underscoring support of the decision-maker as the central issue in DSS, also suggest that DSSs could be given a more active role in the decision-making process by stating in their definition of a DSS that support must be

provided "in situations where it is not possible *or desirable* to have an automated system perform the entire decision process."

A situation in which it might be desirable to automate decision-making is when the decision made by one individual serves as the starting point for another decision-maker. Hogue and Watson (1985:208) refer to this as a sequential interdependent decision [a term borrowed from Hackathorn and Keen (1981)] and report that in their survey 33 percent of the respondent companies utilized DSSs in this way. When a specific decision-maker in the "chain" frequently is not available, an intelligent decision support system could generate an acceptable local decision thereby allowing the decision-making process to continue thus effectively supporting the final decision-maker.

Contributions by researchers in fields such as natural language processing (NLP), logic programming and knowledge representation are making it increasingly easy for DSS builders to create systems that mimic the problem-solving activities of experts. Languages, tools and systems which make the automatic solving of fairly unstructured problems possible are becoming commercially available. For example, The Intelligent Machine Model (TIMM) is a system building tool that automatically creates problem-solving procedures from a set of representative examples (Harmon and King, 1985:119).

However, a word of warning is sounded at this stage: automating decision-making could have behavioural implications such as fear of employee displacement, legal implications such as uncertainty about the accountability for the consequences of a decision, and economic implications such as loss of revenue due to using a decision model that is not applicable to the situation.

(ii) **SYSTEM TRANSPARENCY.**

Many users feel uncomfortable with a support system when it is not transparent; that is, there should be compatibility between the user's model of the problem and that of the system and the user should understand the solution (Watson and Marret, 1979:127).

One way of achieving these objectives is to build interactive DSSs, a fact that is reflected in many definitions of DSS [(Sprague et al., 1980:4); (Scott Morton, 1977)]. Interaction, however, should be an intelligent two-way process since it is in essence a communication process where the one actor guides the reasoning and/or actions of the other towards the objective and vice versa. Ideally the DSS should "take an active role in leading the decision-maker to a problem solution" (Bennet, 1983:43). That is, it should actively support progress towards the goal by, for example, displaying preparedness to accept a wide spectrum of directives.

Conversely, if the number of options open to the user is limited by the system, the communication channel is narrowed. Therefore, if the system prompts the user to make a choice between, say, two alternative problem-solving techniques, the user should be allowed to respond with:

- (a) a choice between the two; or
- (b) a request for a list of all relevant problem-solving techniques; or
- (c) an instruction to use another specified technique; or
- (d) a request for an explanation of the logic that directed the system to present the two specific alternatives; or
- (e) an instruction to the system to make an independent choice and then justify its choice to the user.

Finin, Joshi and Webber (1986:936) propagate the same principle when they talk about "enlarging the range of interactions that responsible users need to be able to have with expert and knowledge based systems to ensure that the advice the user receives is appropriate and is understood correctly". According to them the user should be allowed to (a) volunteer additional information, (b) offer facts from which an answer can be deduced and (c) defer an answer to force the system to pursue lines of reasoning which do not require the requested information. The suggestions above as well as those by Finin et al.(1986) are in line with the DSS concept that the human user should retain control over the decision-making process [(Ford,1985:24); (Kowalski, 1984:95)].

(iii) USER AND MACHINE LEARNING.

The last two items on the list of possible user actions outlined in subsection (ii) by the author, give new meaning to the concept that DSSs should improve the effectiveness of decision-makers. Current DSSs aim to achieve this objective by utilizing the immense speed of the computer to perform for example the large amount of arithmetical calculations required to solve sets of linear/non-linear equations or to search through a vast data base of facts for a specific unit of information. The net result of this is more alternatives for the decision-maker to consider and/or faster decision-making as such. But no or little contribution is made towards actively increasing the user's or the system's level of knowledge about the problem domain and the mechanics of decision-making. This need is clearly reflected in the following:

"The focus of the (DSS) human/machine interface is on learning, creativity, and evaluation rather than on replacement, automation, and routine procedure."

(Thierauf, 1982:64)

By explaining why the user is requested to supply a specific unit of information

- (a) the system becomes transparent to the user;
- (b) the user can verify that the system is "reasoning" along the same lines as the user; and
- (c) in the case of an inexperienced user, the knowledge

level of the user about the problem domain is raised.

One advantage of a system that actively participates in the decision-making process by, for example, "taking over" some of the decisions to be made and then explaining why a certain decision was made, is that it gives the user insight into the problem-solving techniques normally applied by the expert decision-maker who assisted in the construction of the DSS. Such systems will improve the effectiveness of the decision-maker by not only assisting the user to overcome his/her limited cognitive capabilities but also by raising the user's knowledge level. According to Bennet (1983:44) the quality of decisions will also be improved.

3.2.2 LIMITED SUPPORT.

In this section the focus is on:

- (i) the support that DSSs provide for the different phases of the decision-making process;
- (ii) the support for different decision-making models;
- (iii) decision-making aids; and
- (iv) the results of concentrating on decisions instead of the decision-maker.

(i) SUPPORT IN TERMS OF THE PHASES OF THE DECISION-MAKING PROCESS.

In their list of six performance objectives of a DSS, Sprague, et al.(1982:28) mention that a DSS should support

all phases of the decision-making process, a sentiment shared by Hogue et al.(1985:207). In the context of Simon's model of decision-making (Simon,1960:1), it means support for intelligence, design and choice. However, in his survey of 56 systems with DSS characteristics, Alter (1980) found that he was able to divide them into two general categories: data-oriented and model-oriented. The data-oriented systems primarily support intelligence activities, but not design or choice, and the model-oriented systems support design and/or choice assuming that the previous activities have been completed.

In a more recent study of DSS use, Hogue and Watson (1985) interviewed the users of 18 DSSs in an area well-known for its high concentrations of corporate level decision-making. They found that all the systems supported the intelligence phase but only 33 percent provided assistance in the selection of the alternative to be implemented. The systems that did support the choice phase tended to be those with built-in optimization routines. As far as the design phase is concerned, the systems were evaluated on two related yet different capabilities:

- (a) the ability to analyse existing alternatives (answer what-if questions); and
- (b) the ability to suggest new alternatives (problem goal-seeking).

All 18 DSSs supported what-if questions but only 60 percent could perform goal-seeking, a result that is consistent with

the Sprague and Carlson (1982:30) observation that "there has been no substantial support for the design phase".

Although the study by Hogue, et al.(1985) cannot be regarded as representative, it does seem to imply that since 1980 some progress has been made in supporting the first two phases of decision-making. However, Reitman (1982:157) claims that despite the fact that many decision problems involve interactions among intelligent agencies or organizations, current DSSs cannot deal with these directly. Instead, they employ numerical models to replace the interacting intelligences, in the hope that these models will approximate the consequences of the interactions as the system moves through time.

These systems, therefore, cannot make use of knowledge about goals, resources, options and constraints in an interactive framework and are unable to selectively generate good alternatives. Consequently, even with the best current DSSs, it remains up to the user to specify explicitly each of the alternatives to be considered. This seems to be a promising area for the application of Artificial Intelligence techniques.

The fact that there are "decision support systems" in use that do not have goal-seeking capabilities or do not support the decision-maker during the choice phase, is a sad state of affairs. A few possible contributing factors are

orientation of system designers/developers (data processing vs. operations research), use of a specific systems development life cycle (traditional vs. prototype approach) or lack of user-involvement in the development process. Regardless of the reasons why, the problem of limited support needs addressing and as is explained in the next chapter, the field of ES looks promising for assistance in this respect.

(ii) SUPPORT OF DIFFERENT DECISION-MAKING MODELS.

Huber (1981:1) came to the conclusion that the DSSs of the late seventies/early eighties were almost invariably designed to function in rational, or rationalized, decision-making environments despite the fact that many organizational environments, such as political, are more accurately portrayed with models other than the rational model.

According to Bahl and Hunt (1985:83), it should be remembered that "models are templates imposed on situations". A decision-maker's handling of a situation is therefore limited by the model used to define or describe it. Since decision-makers are often not faced by a choice situation per se but by events that call for evaluation and interpretation (Dery, 1983), the availability of alternative models would invite alternative perceptions and actions on the part of the decision-maker.

Unfortunately, in practice the use of one decision-making model tends to foreclose the use of others, even others that

might be preferable on some acceptable criterion [(Bahl, et al., 1985:86); (Sprague, et al., 1982:107)]. To be truly supportive, a DSS should be able to solve a problem using any one of a number of decision-making theories. A clear advantage of such an approach would be the wider acceptance and use of DSSs (Carlson, 1983:18) since they will more closely fit the cognitive styles of the different users.

(iii) PROVISION OF DECISION AIDS.

In their analysis of decision-makers, Sprague, et al. (1982:104,110) quote Newell and Simon (1972) as saying that decision-makers need memory aids of which the following are examples:

- (a) a data base;
- (b) views of the data base;
- (c) workspaces for preserving intermediate results;
- (d) libraries for saving workspaces; and
- (e) profiles to store default and status data.

They then state that although early or traditional DSSs do provide long-term memory aids, such as a data base, short-term aids are not provided. The equivalent of scratch paper and staff reminders to which decision-makers are accustomed usually are not available in most DSSs. Furthermore, additional memory requirements are imposed on the user such as learning (and remembering) a new command language (Sprague, et al., 1982:107). If a DSS functions in an automated office environment many of these facilities are available and easily accessible. If not, aids such as

electronic mail, a word processor and pop-up electronic diaries and calculators should seriously be considered for integration in the DSS.

(iv) FOCUS OF DSS SUPPORT.

Bahl and Hunt (1985:86) have brought an old controversial issue back into the limelight by stating in an article that it seems to be part of the conventional wisdom that large areas of managerial decision-making are essentially inaccessible to a DSS. They feel that if one insists that DSS exist to support decisions, then DSS developers may well have nothing to contribute in so-called "unstructured" or "ill-structured" cases. It is conceivable that there are those who will differ from Bahl and Hunt on the point that the basic purpose of DSSs is to support a decision-maker who is a distinctive personality playing out an institutional role. They will design systems which will be *imposed* on decision-makers giving them information to which they are not "psychologically attuned" to (Mason, et al., 1981:478). This prescriptive approach leads to limited support in a limited number of decision situations.

If, however, one shifts from a preoccupation with decisions to an emphasis on the decision-maker, then the scope of possible DSS support widens considerably. An area in which decision-makers could receive more consideration is that of training. For many decisions, according to Ackoff and Sasieni (1968:441), information is not enough; instruction

on how to make decisions is also required. Hammond (1980) suggests that it may not be sufficient to provide decision aids unless explicit attention is given to how these aids support effective learning (Henderson and Schilling, 1985:157). The ideal "explicit attention" is, in the author's opinion, an *integrated training component*. Without such a component DSSs are limited support systems because:

- (a) although the adaptive (prototyping) approach does provide some opportunity for learning it excludes those not participating in the development of the system from this experience; and
- (b) it is conceivable that in any sizeable system there will be a number of seldomly-used functions in which all users will need occasional "refreshment courses".

The results of empirical studies conducted by Fuerst (1979), Welsch (1981) and Meador, Guyote and Keen (1984) indicated that training as a form of user support should be a primary concern of DSS since it affects DSS success and use.

3.2.3 INFLEXIBILITY.

According to EDP Analyzer (April, 1986) there exists in most organizations a need to react to change more quickly, there is no longer time to fully develop all systems using structured methodologies and users are becoming more sophisticated and expect to receive systems tailored to their needs. What follows is an evaluation of traditional DSS in terms of the above identified needs and expectations.

(i) INABILITY TO ADAPT.

This thesis is concerned with those DSSs that are developed to support decision-makers in modern organizations. Since these organizations function within an environment which is constantly changing (EDP Analyzer, April 1986), it follows that either the DSS or the decision-maker (or both) must adapt over time to be effective in the decision-making process.

Alter (1980) has found that one type of problem experienced by DSS users was that the systems do not really address the decision problems which arise because of the inability to adapt to changes in the problem domain. This inflexibility is caused mainly by the use of conventional programming languages to program decision-making processes. The resulting code is often complex and according to Nilsson (in Shen (1981:67) and Bonzcek, Holsapple and Whinston (1981:269), knowledge changes usually require extensive modifications to the various existing programs, data structures and subroutines. This is caused by the fact that the logic and control components (the "what" and the "how") of a program written in a conventional programming language are not separated. The user of a DSS is therefore forced to live with a *growing inadequacy* on the part of the DSS to support him/her in decision-making situations.

(ii) RIGIDITY OF THE MODELLING COMPONENT.

According to Elam and Henderson (1983:111) most DSSs assume a predetermined underlying structure for a decision. They are designed and implemented for decisions that relate to a specific problem and therefore center around a single, predetermined model that a decision-maker can use to explore various problem characteristics and solutions. The fact that semi- and ill-structured problems are not covered by a single decision model calls for flexibility in the model-building capabilities of a DSS. However, the three shortcomings of corporate models most frequently cited by users are inflexibility of the model, poor documentation and excessive input requirements (Bonczek, et al., 1981:263).

One major problem with traditional DSSs is the fact that they suffer from an inadequacy to develop an integrated model to handle a realistic set of interrelated decisions. The solution is normally a collection of separate models, each dealing with a distinct part of the problem.

Communication between these related models is left to the user as a manual and intellectual process (Sprague, et al., 1982:36). Furthermore, in the exceptional case where a DSS employs application-specific metaknowledge (knowledge about knowledge) in order to determine which empirical knowledge should be interfaced with what procedural knowledge, it is embedded in the processing component of the system (Bonczek et al., 1984:150). This contributes towards system

inflexibility. Possible ways of improving the situation is to:

- (i) provide an easy, flexible way of linking different models and/or model building blocks to form new models; and
- (ii) incorporate in a DSS a knowledge-based model management system capable of automatically performing many of the tedious and time-consuming tasks currently expected of the user.

Such systems will be better equipped to cope with the changing decision-making situations with which the modern decision-maker is confronted.

(iii) INAPPROPRIATE DESIGN AND DEVELOPMENT METHODOLOGY.

The inflexibility of DSSs outlined above also extends to the DSS development process. For years users have had to listen to EDP and MIS department managers explaining how difficult it would be to modify an existing system (inflexible systems design) or how long they would have to wait for a new system because of the application backlog.

Furthermore, in an effort to protect themselves against "never-ending" projects, systems people have adopted a rigid systems development procedure which forces a system requester to fully specify his/her needs before system development commences. Many problems faced by decision-makers, however, can not be fully specified in advance (Bennett, 1983:48). Others change over time necessitating

fast modification of decision support systems [(Hurst, Ness, Thomas and Gambino,1983:123); (Moore and Chang,1983:185)].

Hogue, et al.(1984) found that less than one third of the DSSs covered in their study were developed in less than one month, another third took from one month to one year and the rest took more than one year. The fact that such a large percentage of systems required more than twelve months to be developed could be indicative of the application of a typical transaction processing systems development technique or the traditional MIS systems development life cycle (SDLC) approach. In this respect Moore and Chang (1983:189-194) use the Procurement Decision Support System (PDSS) developed at the Eastern Manufacturing Company to illustrate how the use of an inappropriate design and development methodology could lead to the migration of the projected DSS towards a structured administrative control system, that is, an MIS.

Many DSS researchers advocate the use of prototyping in some form or another for DSS development. The two best-known examples are probably the iterative design and development approach as advocated by Sprague, et al.(1982:16) and the adaptive development strategy of Keen (1980a). But unless analysts and programmers are supplied with easy-to-use DSS specific tools, they will "fall back on the very procedures that have proven successful in MIS design, applying them to unstructured planning activities with which they are inherently incompatible" (Moore and Chang,1983:184). This

topic will be pursued further when the techniques of ES that are applicable to DSS design, are discussed.

Finally, Holmes (1985:6) calls the business world "a seething stew of change" and states that the *highly structured DSSs currently being developed* are too slow and inflexible to meet the realities of the business world.

3.2.4 USER INTERFACE PROBLEMS.

The point of contact between user and system is the user interface. To the user, this interface *is* the system [Stabell (1983:224) referring to Keen (1976)] with the result that problems experienced at this level could lead to total rejection of a possibly superb DSS. Instead of covering all problem areas the following three topics are discussed in some detail:

- * Dialog style.
- * System availability and response time.
- * Presentation formats.

(i) DIALOG STYLE.

A generally accepted objective of DSS is that the man-machine interface should be user-friendly. This, however, is a difficult objective to meet since a dialog style which is easy for one class of user, might be regarded as totally user-unfriendly by another class (Sprague, et al., 1982:207). The way DSS designers usually get out of this dilemma is to provide two dialog styles, one in the form of a cryptic

command language for the experienced user and the other a quasi-natural language (or question-answer or menu or fill-in-the-blanks) interface for the inexperienced user.

Neglecting the user in this way can have far reaching effects because "efficiency in the use of system resources can be ineffective if a system is not designed to match the needs and abilities of its users" (Li,1984:1). Since user performance deserves as much attention from system designers as does system performance, the ideal DSS should make provision for the preferences, skills and task requirements of all classes of users by including not only two but multiple dialog styles in the man-machine interface sub-system.

Although many easy-to-use query languages exist (for example, Structured Query Language or SQL and Query By Example or QBE) they could still be termed user-unfriendly. One reason for this is given by Sowa (1984:287) who points out that users are required to be knowledgeable about implementation issues such as the name of the file to be queried. Users should only be concerned with their own views of the world and the system should apply its knowledge of the subject matter and database organization to determine what fields and files to access.

Recent developments in the microcomputer world has led to a suggestion in EDP Analyzer (July 1984:8) that DSS designers should learn from the success story of Apple's Macintosh

computer. "Users will like the ability to quickly move the cursor (via a "mouse"), call up a menu, select an item on the menu and get the results - all in two to three seconds." The icon-based dialog style would seem to be equally acceptable to inexperienced as well as experienced users.

(ii) SYSTEM AVAILABILITY AND RESPONSE TIME.

Two important behavioural issues related to the acceptance and use of DSSs are system availability and response time. Personal practical experience on large main frame computers has indicated that poor DBMS performance due to lack of main memory pages, data transmission errors due to low quality telecommunication services and overloaded I/O channels frequently lead to totally inadequate response times. Add to this operating system, DBMS or network control program malfunction, which could make the DSS inaccessible, and the user is left stranded with no support at all (Alter, 1980). In the author's opinion system availability of less than 99.9 percent uptime and a response time slower than 3 seconds are unacceptable. This sentiment is shared by Bonczek et al. (1984:170) who describe a micro-based system that actually achieved this level of performance.

Many data processing professionals, including DSS designers and developers, still view microcomputers primarily as spreadsheet machines with some useful word-processing capabilities and some very limited, simple file-handling facilities (Bonczek, et al., 1984:169). Although currently

available desktop computers do fall short of the memory and computational powers needed to satisfy the requirements of a large-scale DSS (Reitman et al., 1985:166):

- (a) a number of highly successful statistical and modelling mainframe software packages such as Statistical Analysis System (SAS), Statistical Package for Social Sciences (SPSS) and Interactive Financial Planning System (IFPS) have already been made available for use on micro's;
- (b) there exists DBMS's in the micro world which compares favourably with their mainframe counterparts such as Cullinet's IDMS. (MDBS KnowledgeMan, short for Knowledge Manager, is one such a tool that supports postrelational semantic structuring, extensive integrity guarantees, encryption and a host of performance controls including transaction logging and a non-procedural query interface (Bonczek, et al., 1984:162,170)); and
- (c) competition and user naivety are forcing microcomputer software developers to produce increasingly more user-friendly systems.

With all three basic components of a DSS (user interface, DBMS and modelling facilities) commercially available, micro's should be seriously considered as alternative delivery machines for small to medium-sized DSSs.

(iii) PRESENTATION FORMATS.

Early DSSs do not provide decision-makers with familiar representations which support conceptualization (Sprague et al., 1982:107). Despite the fact that decision-makers

constantly use phrases such as "bottom line", "payoff curve" and so on, indicating conceptualization through graphical representations, the most-used presentation form is still the two-dimensional table.

A possible reason for this state of affairs is the fact that it is currently still very much a controversial issue whether graphics and colour contribute meaningfully towards enhancing decision-maker effectiveness. According to Benbasat, Dexter and Todd (1986:1094) academic surveys of the field by DeSanctis (1984), Ghani (1981) and Ives (1982) indicate that:

- (a) the dramatic claims made about the benefits of colour and graphics in the professional literature appear to be somewhat exaggerated and are not well substantiated by empirical evidence;
- (b) research findings are inconclusive and often in conflict with each other; and as a result
- (c) there does not seem to be a well-grounded empirical or theoretical base in the information systems and administrative sciences fields to guide practitioners who are designing presentation formats.

The empirical study undertaken by Benbasat and Dexter (1986:59-83) indicated that:

- (a) the use of colour only led to improvement in decision-making when time was a critical factor; and
- (b) colour and graphics have a role to play in conditions

where cognitive strain is likely to occur. (Cognitive strain is defined by Benbasat and Dexter (1986:60) as the breakdown of the decision-maker's cognitive processes, occurring when he or she suffers from information overload. Overload is the point at which information processing demands exceed the information capacity of the individual).

It could be concluded that:

- (a) personal characteristics of the user such as cognitive style, intellectual ability and physiological make-up; as well as
- (b) characteristics of the environment including the task of decision-making, and the characteristics of the specific DSS in use should receive special attention in research projects addressing this issue [(Mason and Mitroff, 1973); (Bariff and Ginzberg, 1980); (Benbasat and Taylor, 1982); (Lucas, 1981)].

DSS designers should furthermore take note of the popularity of those micro-based spreadsheet packages which include flexible, integrated graphics capabilities. The relative ease with which users can develop report formats to suit their exact needs and to store them for later use, should be an important design consideration (Reitman et al., 1985:175).

3.3 OTHER PRESSING REASONS.

3.3.1 KNOWLEDGE WORKER PRODUCTIVITY.

A factor that clearly influences the productivity of any country's work force is the shift from an agrarian to an industrial society and then onwards to an information and service society as is experienced in the USA. Whereas technology has been, and continues to be, the key to enhancing the productivity of the blue collar worker, it is felt that technology is not being applied to the white collar sector on a large enough scale to have a significant impact (Borko,1982:203). Knowledge workers, that is, the subclass of white collar workers that exert the greatest influence on the decision-making process in an organization, need technologically up-to-date decision support to improve their productivity.

Without any doubt, decision-makers of the near future will find themselves more and more in situations where the tasks for which they are responsible, will entail the gathering, understanding and evaluation of an overabundance of information [(Hammer,1982); (Hayes-Roth,1984:28)]. In addition to possibly overwhelming the human's capability to react in a timely and effective manner, this excess amount of information will often be incomplete and erroneous (Hong,1983:18), affecting the productivity of decision-makers negatively.

DSSs displaying the limitations and shortcomings outlined in the previous section, will obviously not meet future user

requirements. What is needed, is a little "technology push": DSSs incorporating state-of-the-art AI and ES techniques, should be developed to assist the user in counter-acting the effects of information overload on productivity. A discussion of how white collar worker productivity could be increased through the application of ES tools and techniques, is deferred to the following chapter.

Considering that:

- (a) learning to use one DSS does not help one to use any other DSS (implying costly formal retraining whenever a new DSS is implemented) (Andriole, 1985:12); and
 - (b) the cost of knowledge workers is significantly greater than the cost of clerical personnel (implying that even a small increase in knowledge worker productivity can result in significant savings) (Borko, 1983:205),
- it makes sense to standardize the user-interfaces of all DSSs within an organization (without limiting the number of different interfaces made available on any specific system!) to facilitate ease of use and ease of training. Here again, microcomputer software developers have set the trend: every successful software package has an extensive online help facility, listing and explaining how to use all the available commands.

By applying the tools and techniques of ES, knowledge-based CAI and NLP, system designers and developers can produce DSSs that will:

- (a) eliminate the need for costly formal training sessions by allowing users to choose between productive use of the system and student-paced training; and
- (b) encourage users to explore the data base contents and available DSS features by allowing free-form interaction.

Since both (a) and (b) above are aimed at increasing the knowledge level of the worker, improved productivity could follow. This topic is pursued further in chapter 5 under the topic "Characteristics of Expert Decision Support Systems".

3.3.2 EMPLOYEE TURNAROUND AND RELATED TOPICS.

A shortage of sufficiently knowledgeable professionals generally gives rise to high employee turnaround, forcing companies to sometimes pay unrealistically high salaries in an effort to keep their experts from seeking "greener pastures". Furthermore, when a company loses an expert decision-maker through death, retirement or resignation, it is difficult to find a suitable replacement. To remain competitive a company must constantly seek ways and means to curb spiraling costs, the most important of which could well be employee remuneration.

The realization that the data of an organization is a valuable resource that should be managed effectively, has led to the development and general acceptance of data base management systems. Likewise, the problem-solving skills

developed by organizational employees during their time of employment, should be regarded as a valuable company asset. In many instances the skills referred to here are directly related to considerable company investment in employee training. The different forms of employer-backed education, with the purpose of developing and refining an employee's knowledge and problem-solving skills, are costly and organizations should attempt to "institutionalize" this knowledge. By capturing the expertise of their employees in one or more computerized "knowledge bases", a company can, to some extent, retain the skills of an employee that leaves the service of the company.

Company-provided in-house training is costly mainly because it implies that the trainees as well as the expert(s) conducting training courses are "unproductive" for the duration of a course. A possible solution is to make the expertise of the most proficient employees available, in computerized form, to either serve as an aid to the employee who fills in for the expert while the latter is conducting a training session or as the basis of a computer-based training course. As a side benefit the level of expertness of the experts within a company could be raised since the process of encoding their knowledge will force them to critically examine their problem-solving techniques.

3.4 CONCLUSION.

It could be said that DSSs which are developed using

traditional concepts and development methodologies display a certain amount of passiveness in their support of decision-makers, provide only limited support, are inflexible and therefore insensitive to changes in their environment and are not always easy to use. Some scope for enhancement exists in areas such as user-system interaction, model-building and the tutoring capabilities of DSSs. System builders should endeavour to support more phases of the decision-making process while at the same time taking note of the fact that it could be advantageous to incorporate multiple decision theories in a single DSS.

Two of the more important organizational problem areas where DSSs can make a more meaningful contribution, are white collar worker productivity and employee turnaround. A promising direction that needs investigation is that of intelligent, knowledge-based tutoring which could relieve scarce expert problem-solvers from their time-consuming training responsibilities, freeing them to perform the tasks they were trained (and being paid) for.

Possible sources of solutions for the problems that have been identified, are the field of Artificial Intelligence and related fields of study, such as ES and NLP. The next chapter is therefore devoted to a survey of current practices in these fields and an evaluation of the applicability thereof to problems faced by DSS builders.

CHAPTER 4.

THE PROMISE OF EXPERT SYSTEMS.

4.1 INTRODUCTION.

The past ten years have witnessed the design and successful implementation of a number of computer-based systems that simulate human experts by drawing on large amounts of domain-specific knowledge and problem-solving heuristics (rules of thumb). The applied AI research that has led to the development of these systems, aptly named Expert Systems (ES), has provided management information practitioners with the necessary technology to create frameworks for capturing, structuring and using expert knowledge in management systems. Of particular interest to this thesis is the extent to which ES technology can be applied in building computer-based systems that support the organizational decision-making process in a more efficient way than traditional DSSs while at the same time increasing the effectiveness of the decision-maker.

To understand why ES technology is particularly suited for problem-solving situations in which alternatives are many, data incomplete, and conclusions likely to be expressed as unquantifiable likelihoods - "the very circumstances in which most humans make their most important decisions most of the time" (Alexander, 1982:142) - the architecture and

characteristics of ESs are discussed in the following sections. This is followed by a comparison of ES and DSS to illustrate the close relationship between the two technologies. Finally, those features of ESs that can contribute towards solving some of the problems experienced by users of current DSSs, are identified.

4.2 OVERVIEW OF MAJOR ES COMPONENTS.

Judging by the wide variety of conceptual views of ES presented to the reader of ES literature [Hayes-Roth, Waterman and Lenat, (1983:17); Frenkel (1985:580); D'Ambrosio (1985:373); Oxman (1985:37); Turner (1985:12); Ford (1985:24); Sowa (1984:279); Hong (1983:15)], it seems as if there is presently no universally accepted architectural representation of the structure of ES. Figure 4.1 represents the author's view of the more complex type of ES and indicates the main components as:

- (i) User interface;
- (ii) Inference engine;
- (iii) Knowledge base;
- (iv) Global working space;
- (v) External device interface; and
- (vi) Explanation subsystem.

(i) USER INTERFACE.

This is a computer program that:

- * allows a user to ask questions;

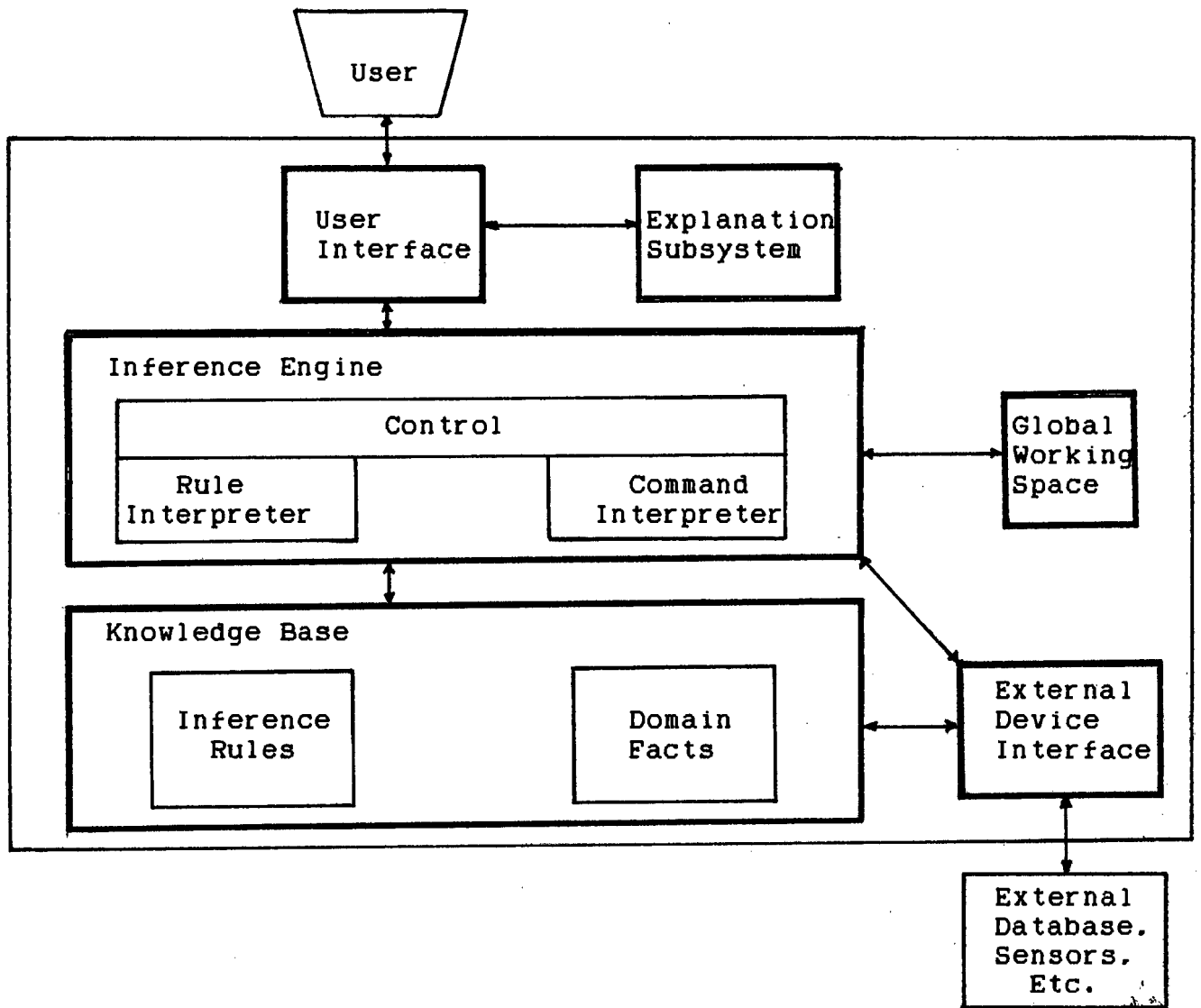


Figure 4.1

ES Architecture

Compiled from various sources.

- * prompts the user for relevant information;
- * provides responses to user requests; and
- * supports any other communication that would be necessary between the ES and the user [(Oxman, 1985:37); (Buchanan and Duda, 1983:173)].

It is the mechanism through which the user accesses the system's knowledge to help him/her solve problems (Kidd, 1985:16). In many instances this communication between user and system is in natural language or an approximation thereof, for example, "canned" or pseudo English [(Hayes-Roth, Waterman and Lenat, 1983:16); (Jones, 1985:86)].

(ii) INFERENCE ENGINE.

The inference engine is responsible for the processing of user requests and commands. Its objective is to simulate the deductive thought processes of a professional problem-solver. To succeed, it interfaces directly with, and controls all other components of the system (Buchanan, et al., 1983:173). It provides the system with its reasoning power and consists of three parts: a control component, a rule interpreter and a command interpreter.

The *control* component can be thought of as a kind of supervisor. It analyses user input in order to determine the problem solving strategy which the system must follow to service a user request. That is, it decides on the sequence in which rules must be "fired" (executed) by the rule interpreter [(Harmon and King, 1985:53-54); Hayes-Roth, et

al., 1983:18)].

The *rule interpreter* is that part of the inference engine that attempts to reach some conclusion based on (a) information provided by the user, (b) the rules and facts contained within the knowledge base and (c) intermediate results temporarily stored in the global workspace (if any) [Oxman (1985:37); Hayes-Roth, et al. (1983:18)].

The *command interpreter* services user and system commands. For example, user commands such as "How" and "Why" are interpreted as requests to activate the explanation subsystem. System components, such as the rule interpreter, make use of the services provided by the command interpreter when, for example, requesting extraction of data from external devices (Li, 1984:128).

(iii) KNOWLEDGE BASE.

As illustrated in Figure 4.1, the knowledge base contains two separate "databases": a set of domain facts and a collection of inference rules (Hayes-Roth, et al., 1983:19). The domain facts database is composed of simple facts in the form of descriptions that identify and differentiate objects and classes of objects in the domain (Hayes-Roth, Waterman and Lenat, 1983:12). For example, the fact "Abel is a male person" is presented in a PROLOG database as:

```
male(Abel).
```

In addition, it contains descriptions that express the dependencies and associations between items in the knowledge base Hayes-Roth, et al., 1983:12). For example, the relationship "Abel is the father of Mary" is encoded as: father(Abel,Mary) in PROLOG (Davis,1985:54).

The set of inference rules on the other hand, are the procedures which specify the operations to be performed during problem-solving [(Oxman,1985:36); (Hayes-Roth, et al., 1983:12)]. For example:

- * An accounting item is a current asset if:
 - the item is an asset
 - and: the item is liquid within one year.

To solve the problem whether a specific item is a current asset, the rule above specifies that the following operations should be performed:

- (a) Prove that the item is an asset.
- (b) Once (a) has been proven, prove that the item will be liquid within one year.

(iv) GLOBAL WORKING SPACE.

New facts derived during rule processing can (a) either be entered automatically into the knowledge base or (b) they are stored in the global workspace together with other intermediate results, hypotheses and decisions (Hayes-Roth, Waterman and Lenat, 1983:16) to form a kind of situation model (Simons, 1984:180). In the case of (b) the user is

given the option to erase newly-derived facts from memory or to save them in the knowledge base.

(v) EXTERNAL DEVICE INTERFACE.

For the sake of efficiency ESs keep their knowledge bases in primary storage. In the case of very large knowledge bases, however, this might not be possible. Sometimes the need also arises to consult existing external databases. Provision is therefore made via the external device interface, to access external storage devices and to manage databases on these devices [(Sowa, 1984:319); (Jarke and Vassiliou, 1984)].

(vi) EXPLANATION SUBSYSTEM.

The function of this subsystem is to explain the actions of the system to the user. It answers questions about why some conclusion was reached or why some alternative was rejected [(Hayes-Roth, et al., 1983:18); (Gevarter, 1983:39)].

By combining the components outlined above in one system, a synergistic effect is achieved. The resulting system displays problem-solving characteristics approximating those of human decision-makers [(Buchanan and Duda, 1983:172); (Hayes-Roth, 1984:29)]. These characteristics are examined more closely in the next section.

4.3 CHARACTERISTICS OF EXPERT SYSTEMS.

Many researchers in the field of ES tend to avoid giving a precise definition of an ES and rather focus on the characteristics they would like these systems to display. For example, Feigenbaum and McCorduck (1983:31-50) state that an ES is a system that:

- (i) has expert rules and avoids blind search;
- (ii) performs well;
- (iii) reasons by manipulating symbols;
- (iv) grasps fundamental domain principles;
- (v) has complete weaker reasoning methods to fall back on when expert rules fail;
- (vi) deals with difficult problems in a complex domain;
- (vii) can take a problem description in lay terms and convert it into an internal representation appropriate for processing with its expert rules; and
- (viii) can reason about its own knowledge especially to reconstruct inference paths rationally.

Some experimental systems display many of these "desirable" characteristics but commercially available ESs generally include only a subset of these features. The following discussion concentrates on those characteristics displayed by most ESs and concludes with a list of enhancements and additional features that ESs of the near future will exhibit (given sustained research interest and ample funding). No attempt is made at linking ES features with problems faced

by DSS designers, developers or users. The main objective is to set the scene for a comparison between ESs and DSSs.

GENERAL ES CHARACTERISTICS.

From the many descriptions of experimental and commercially available ESs in the ES literature it was possible to determine that, in general, ESs:

- (1) make use of a variety of knowledge;
- (2) make use of specially tailored logical data structures;
- (3) separate descriptive and problem-solving knowledge;
- (4) employ heuristic problem-solving techniques; and
- (5) can explain and justify their lines of reasoning.

A short discussion of each of these characteristics follows.

(1) ESs MAKE USE OF A VARIETY OF KNOWLEDGE.

There is agreement among researchers that the power of an ES is based on the amount, quality and type of knowledge available to it [(Buchanan and Duda, 1983:184); (Gevarter, 1983:39); (McDermot, 1984:12); (Lenat, 1983:247); (Hayes-Roth et al., 1983:3,7)]. The types most commonly found in ESs are descriptive or factual, problem-solving or procedural and task-specific or heuristic knowledge (Bonnet, 1985:145). Under *descriptive knowledge* is meant simple domain facts as well as "facts" such as causal relations and associations between items in the domain. *Problem-solving knowledge* are the rules or procedures that are applied to the domain facts

in order to reach a solution to a problem. The *task-specific knowledge* is "the model of the world associated with a specific problem, its status, and its history" (Gevarter, 1983:39) and is stored in the global database (global workspace in Figure 4.1).

Descriptive and problem-solving knowledge are further classified as:

- (i) Public knowledge;
- (ii) Private knowledge; or
- (iii) Metaknowledge.

Public or surface knowledge are the definitions and theories which are found in published literature (Alexander, 1984:102). Because of its accessibility it is the most commonly found knowledge in ESs. *Private or deep* knowledge are the rules of thumb (heuristics) which represent genuine understanding based on scientific principles or human perceptions of cause and effect (Alexander, 1984:102). It is subjective, ill-documented and judgemental and has proven difficult to extract from human experts. At the next deeper level lies *metaknowledge* which is compiled experiential knowledge such as the records kept by a human problem-solver of the success and failure rate of rules used (Lenat, Davis, Doyle, Genesereth, Goldstein and Schrobe, 1983:235). These statistics guide him unconsciously in his choice of rules to try in new situations. Any specific ES will only have a limited number of meta-rules mainly due to the fact that experts are not always aware of the existence of such rules.

(2) ESSs MAKE USE OF SPECIALLY TAILORED LOGICAL DATA STRUCTURES.

For each type of knowledge identified in (1) above there exists a data structure or knowledge representation scheme which optimizes its utilization. Example schemes include predicate calculus, scripts and semantic inheritance networks. Although some researchers advocate the use of a single scheme [(Kowalski, 1979, 1984); (Elam and Henderson, 1983)] others feel that multilevel ESSs, that is, systems that use different schemes for different levels of knowledge, will be more efficient [(Wiederhold, 1984:71); (Sloman, 1985); (Michaelson, Michie and Boulanger, 1985:310)].

(3) ESSs SEPARATE DESCRIPTIVE AND PROBLEM-SOLVING KNOWLEDGE.

In computer systems developed along conventional lines as well as in some early ESSs, descriptive knowledge (data) and the procedures that manipulate it are inextricably intertwined. The modern trend in ES is to represent general domain principles as inference rules and to use separately stored descriptive knowledge (domain facts) to instantiate these principles in specific situations (Buchanan, 1985:30). Instantiation is the process of giving meaning to a general rule by substituting variables with data from the factual database (see Kowalski, 1979:15). The procedural knowledge (inference engine) needed to perform this instantiation, is also a completely separate module within the ES. The main advantage of this approach is the relative

ease with which the system can be expanded or updated.

(4) ESs EMPLOY HEURISTIC PROBLEM-SOLVING TECHNIQUES.

ESs differ substantially from conventional computer programs because their tasks normally have no algorithmic solutions and because often they must make conclusions based on incomplete or uncertain information (Gevarter, 1983:39). Instead of using optimization or formal reasoning techniques, results or decisions are determined by rules of thumb and judgemental reasoning.

(5) ESs CAN EXPLAIN AND JUSTIFY THEIR LINES OF REASONING.

The ability of expert systems to provide some sort of explanation of their line of reasoning is one of the most important and powerful ideas to emerge from such systems (Brachman, Amarel, Engelman, Engelmores, Feigenbaum and Wilkins, 1983:42). Three levels of sophistication can be identified in current ESs:

- (i) At the lowest level there are those systems in which the system builder annotates the system during creation with text strings justifying the system's methods. These text strings are then displayed during execution in response to queries. There is no way to ensure that the justifications provided, accurately reflect what the code actually does (Swartout, 1985:96).
- (ii) At the next level, explanation is associated with the listing of those rules that "fired" during the course

of a problem-solving session, in some instances with a little bit of syntactic doctoring (Kidd, 1985:17). This form of explanation is more reliable but of more use (more intelligible) to the system debugger than to the novice user.

- (iii) At the highest level, use is made of the system's self-knowledge (that is, knowledge about its own operation and structure) and of fundamental domain principles to explain and justify the system's actions. Currently the use of self-knowledge is somewhat simplistic, providing only basic explanation and justification capabilities [(Hayes-Roth, Waterman and Lenat, 1983:28); (Lenat et al., 1983:231-233)].

ES FEATURES IN THE PIPELINE.

- (1) The ability to communicate with ESs in full natural language including verbal communication through discrete and (possibly) continuous speech [Biermann, Rodman, Rubin and Heidlage (1985); Bonnet (1985:53-59); Simons (1984:162-166)].
- (2) Faster performance through parallel processing (Pountain, 1985).
- (3) Improved knowledge representation that will facilitate automatic knowledge expansion using all the different learning paradigms (Michalski, Carbonell and Mitchell, 1983).
- (4) Explanation subsystems which are expert systems in their own right, reasoning from domain principles about the

actions of the application expert system [Hayes-Roth, et.al.(1983:28); Lenat, et.al. (1983:231-233); Brachman, et.al.(1983:48-49)].

When considering the objectives of DSS (Chapter 2) in the light of the characteristics of ESs outlined above, it becomes obvious that ES technology has much to offer to the field of DSS. In the following section attention is drawn to the fact that although DSS and ES differ in some respects, they also have much in common.

4.4 SIMILARITIES/DIFFERENCES BETWEEN ES AND DSS.

Since DSS and ES are both concerned with the reaching of conclusions based on a given set of facts, questions such as: "What are the similarities/differences?" and "Why two separate fields of study?" arise. This section is an attempt to answer some of those questions.

4.4.1 SIMILARITIES.

(i) COMMON OBJECTIVE.

The fundamental goal of DSS and ES is basically the same: they seek to improve the quality of decisions made (Ford, 1985:24). DSSs achieve this goal by providing quick and easy access to data and models which are relevant and applicable to the specific decision. ESs simply go one step further by also providing access to the problem-solving skills of an expert in the field. That is, ESs actually

provide the user with a conclusion (decision) that is more likely to be correct than the conclusion many users would have reached.

(ii) SEPARATION OF KNOWLEDGE TYPES.

DSS and ES developers agree that procedural and non-procedural knowledge should be separated. The following citations serve to stress the point.

Sowa (1984:279) on ES:

"Knowledge-based systems take the ultimate step of separating the computational steps from the control flow and putting them in nonprocedural tables."

Bonczek, et.al.(1984:157) on DSS:

"A generalized problem processor is a single invariant mechanism whose code does not change, regardless of the problem domain in which it is being used."

A comparison of Figure 4.2 and Figure 4.3 further highlights the similarity between the structure of a DSS and that of an ES. Both have the following components in common:

- (a) a language system that analyses inputs and generates outputs;
- (b) a processing component which controls the actions of the system and which does not change when a new piece of knowledge is added to the system; and
- (c) a store of domain-specific knowledge.

(iii) AGREEMENT ON METHODOLOGY.

Both DSS and ES are generally designed and developed using an iterative or prototyping approach [Sprague and Carlson (1982:137-155); (Buchanan et al.,1983)]. Keen (1980b) suggests that DSSs are relevant to situations in which a final system can only be developed through an adaptive process of learning and evolution, while Buchanan, et.al.(1983:160-166) recommend the development of a Mark-I ES from which experience should be gained before "throwing it away" in favour of the Mark-II model.

(iv) RELIANCE ON OTHER DISCIPLINES.

The builders of both types of systems draw heavily on the knowledge possessed by experts in other disciplines. In ES the knowledge engineer (ES builder) is largely dependent on some expert in the task domain to obtain problem-solving heuristics to augment text book facts [(Wiederhold, 1984:63); (Gevarter,1983:39)]. In DSS the builder relies on OR/MS experts to provide tools and techniques (for example, models) with which to support the decision-making process [(Holmes,1985:6); (Bonczek, et al. 1984:152)].

4.4.2 DIFFERENCES BETWEEN DSS AND ES.

(i) TRADITIONAL vs. LOGIC PROGRAMMING.

The modelling tasks required of traditional DSSs are understood well enough that they can be specified in mathematical formulae and procedurally programmed using COBOL or FORTRAN. DSSs have been incapable of modelling

expert decisions, because such decisions are difficult to express in the form of hard and fast rules required by procedural programming languages which are rich in syntax but weak in semantics (Simons, 1984:181). Expert systems, on the other hand, are rich in semantics and employ unique programming techniques to model expert decisions. Expert knowledge is captured in, for example, a set of if-then rules which are processed in strict order of deductive inference. Furthermore, these rules allow conclusions to be reached with less than complete certainty, thereby modelling the uncertainty which is often present in expert judgements (Simons, 1984:182,185).

(ii) KNOWLEDGE REFINEMENT.

ESs have the ability (self-knowledge) to expand and refine their knowledge bases by recording the outcome of various lines of reasoning (conclusions) in the global database [(Ford, 1985:24); (Hayes-Roth, Waterman and Lenat, 1983:28)]. This knowledge can then be transferred to the domain facts database as permanent entries, implying a "learning" from experience [(Michalski, Carbonell and Mitchell, 1983); (Starrs, 1985:38)]. DSSs do not have this capability.

(iii) SELF-KNOWLEDGE AND EXPLANATION.

ESs have the ability to "explain" their lines of reasoning (Buchanan, 1985:31). One way this is achieved, is by keeping record of the rules which are used during a

session. These records constitute a form of self-knowledge: the system "knows" what it has done. The system can also tell where it is heading by displaying the rule(s) which logically follows the current one. DSSs developed along traditional lines cannot explain their actions to the user.

(iv) TARGET USER GROUP.

Currently the intended user groups of DSSs and ESs differ. The user of a particular DSS is often the decision-maker who helped design the system. ESs on the other hand, tend to be developed for use by anyone in a specific field who needs to draw on the knowledge of some expert(s) in the field who assisted in the development of the ES (Ford, 1985:25). It is, however, foreseen that as ES technology matures, more users will request the development of such systems, and then assist in the creation thereof.

The similarities pointed out above, suggest that ES and DSS are closely related disciplines. The differences on the other hand, are mainly caused by the unique capabilities of ESs. Since DSSs are weak in these areas, it stands to reason that DSS (and the user) will benefit from a merging of DSS and ES. The benefits that can be expected, are discussed in the following section.

4.5 APPLICABILITY OF ES TECHNOLOGY TO PROBLEMS OF DSS.

This section highlights those features of ESs that can contribute meaningfully towards overcoming some of the shortcomings and limitations of DSSs that were outlined in Chapter 3. It concentrates on one problem area at a time, choosing applicable ES tools and techniques, and then justifying the choices made by discussing the advantages that will accrue upon implementation of the suggestions.

4.5.1 COUNTERACTING DSS PASSIVENESS.

One way of solving this problem would be to increase the amount of communication between user and system while simultaneously improving the quality thereof.

(i) INCREASING THE *AMOUNT* OF COMMUNICATION.

A DSS that discusses with its user the important features of a problem, its relationship to other problems, and the reasons why some strategies may be effective and others ineffective (Blanning, 1985:158), could well be termed an active DSS. Such discussions could lead to a broadening of the user's domain knowledge, deeper insight in the intricacies of the decision-making process and a subsequent improvement in the decision-making capabilities of the user. To achieve this objective:

(a) Communication between user and system should

preferably take place in natural language. This does

not necessarily imply lengthy input and output sequences but merely suggests that because users are generally not computer scientists, they would in many instances be unable to follow a formal representation of, say, the system's reasoning (McKeown, 1986:962).

(b) The DSS should be augmented with a domain-specific knowledge-base which must include not only general domain principles but also heuristic problem-solving rules. This combination of domain knowledge and rules prescribing how and when to use the knowledge, will allow a DSS to reason and then converse intelligently about domain-related matters. Bonczek, Holsapple and Whinston (1981a) provide useful guidelines in this regard.

(c) An explanation facility similar to those found in ESs should be added to DSSs. It is conceivable that some decision-makers will use a knowledge-based DSS to confirm a conclusion already reached. In the case where the system comes up with a different conclusion, such users need to be able to understand both how and why the system made its decisions in order to decide whether to accept or reject the system's advice (McKeown, 1985:89).

(ii) IMPROVING THE *QUALITY* OF COMMUNICATION.

In this respect attention should be given to the quality of the domain facts and encoded expertise, the ways in which they are processed and the level of self-knowledge

built into the system.

(a) Quality of facts and rules.

Improvement of the quality of communication will come as a fringe benefit when DSS builders start using knowledge engineering techniques to extract application-specific problem-solving expertise from experienced decision-makers. During the knowledge acquisition phase of developing an "expert" DSS, the expert problem-solvers will be forced to consider critically, and then to formalize, the manner in which they think about a problem (Starrs, 1985:40). This could lead to new or improved approaches towards problem-solving in the applicable domain which will be conveyed to decision-makers when they use the system thereby adding a training dimension to communication.

(b) Improved processing.

Logical deduction and judgemental reasoning, two techniques commonly applied in knowledge-based systems, are ideal candidates to further improve the quality of communication. Individual bits of information are related to general rules and lines of reasoning, enabling the system to decide what questions it really needs to ask in order to arrive at conclusions and/or recommendations ("The Economist", Sept. 1981:95). This elimination of "noise" (trivial questions with obvious answers which the system can logically deduct) will raise the intellectual level at which communication

takes place.

Since DSSs are systems in which man and machine work together to solve problems, it is suggested that designers that intend including judgemental rules in future knowledge-based DSSs, should not underestimate the value of human intuition and gut-feeling. They should allow the user to override any assumption or decision on the part of the system by presenting the user with a wide range of choices at each major decision point within the system (This topic was covered extensively in section 3.2.1).

(c) Adding self-knowledge to DSSs.

Knowledge-based systems can translate user queries into database terms, using application knowledge to rephrase the queries into better queries for processing. This ES feature will enhance communication in a way transparent to the user. The following example illustrates how this could be achieved:

QUERY BY USER: "How much oil did India import in
1982?"

STRAIGHT TRANSLATION (resulting in an inefficient
database query):

```
FOR ALL bill_of_lading_documents
    WHERE country = "INDIA"
    AND cargo_type = "OIL"
    SUM (quantity_unloaded).
```

TRANSFORMED TRANSLATION (resulting in a more efficient query):

FOR ALL ships

WHERE ship_type = "TANKER"

FOR ALL trips

WHERE dest_port_country="INDIA"

SUM (quantity_unloaded).

(Wiederhold, 1984:67)

The transformed query is more efficient since the application-specific knowledge that all Indian oil imports are by tanker, is used to access the much smaller "ships" file. Further efficiencies may accrue because the "ship-type" attribute, here having the stable value Tanker, is probably indexed, whereas the larger "bill-of-lading-document" file, with less stable attributes, may not be indexed. More efficient database queries, which obviously will improve the efficiency of DSSs, are made possible by the fact that the system actively participates in the query formulation process, drawing on its domain-specific knowledge-base as it does so.

4.5.2 TOWARDS SOLVING THE PROBLEM OF LIMITED SUPPORT.

In this subsection it is argued that ES technology can provide DSSs with enhanced abilities to:

- * support users in building useful models;

- * provide users with a wide range of support;
- * perform goal-seeking; and
- * support users of different cognitive styles.

(i) MODEL BUILDING.

The ability of ESs to integrate and fuse large volumes of information from a complex real-world environment in real-time (something which, according to Hong (1983:14), humans are incapable of), can assist the decision-maker in developing and evaluating models that more closely resemble real-world situations. An example of such expert modelling systems is REVEAL (by Decision Products Services, Inc.) in which forecasting and scenario planning models can be developed which take into account a measure of uncertainty - and therefore provide a more realistic view (model) of the world (Vince, 1985:22).

(ii) EXTENDED RANGE OF SUPPORT.

The unique programming techniques employed by ES developers to model expert decisions, for example production rules, offer two major advantages over those used for DSSs. *Firstly*, since production rules are independent of each other, program revision and/or modification is made very easy (Buchanan, 1985:30). And *secondly*, it provides an ES with the capability of explaining its line of reasoning by displaying the rules used to reach a conclusion [(Kowalski, 1984:102); (Michaelsen and Michie, 1983:240)]. The application of these techniques during DSS development, could

lead to systems that provide a wider range of user support such as the following:

- (a) By displaying its reasoning strategies, the system supports the user's need to verify that there is compatibility, at the cognitive level, between the system's model of the solution and the user's model (Watson and Maret, 1979).
- (b) By adding an explanation dimension to DSSs, ES programming techniques provide DSSs with the ability to support conflict resolution in group decision-making situations (Rathwell and Burns, 1985:256).
- (c) By facilitating ease of system development and modification, ES programming techniques provide users with the much needed support of faster DSS development. In addition, existing systems can be technically supported by IS personnel for longer because of the ease with which modifications can be made (Buchanan, 1985:30).

(iii) **GOAL-SEEKING.**

ES technology, when applied to DSSs, will lead to the development of systems that support goal-seeking. The three most commonly used ES control strategies are : (1) data driven (forward-chaining), (2) goal driven (backward-chaining), and (3) a mixture of data and control driven strategies [Harmon and King (1985:55); Clancey (1986:39-40)]. The second strategy above is of interest at this stage and is discussed in some detail below.

Backward-chaining starts by transforming the query posed to it into an hypothesis (for example, "Should we relocate the factory?" becomes "We should relocate the factory.") and then focuses its efforts on finding substantiating evidence for the hypothesis. If it succeeds in finding such evidence, the problem is regarded as solved and the hypothesis is presented to the user as an affirmative answer to the original query. If insufficient supporting evidence is found, the problem is divided into sub-problems (Kowalski, 1984:95), which, when solved, will imply the solution of the original problem. For example:

Hypothesis 2a(i): We have a suitable site available AND

Hypothesis 2a(ii): We have sufficient funds available to
build a new factory.

OR

Hypothesis 2b: We have the funds to buy a suitable new
factory.

If facts in the knowledge base indicate the availability of a suitable site and sufficient funds, the user is informed that s/he should relocate the factory.

The search for a solution continues in this fashion, working systematically from the original problem, until a sub-problem is encountered which can not be further subdivided and the knowledge base contains no evidence to support the hypothesis. At this point the system turns to the user and

asks for the relevant facts (Harmon and King, 1985:54). If the user cannot supply the needed information, then this line of reasoning is dropped in favour of other possible lines of reasoning.

It should be noted that at as soon as the system finds the solutions to all sub-problems defining the original problem, it presents the user with *a* solution to the problem. The user can accept the solution, or instruct the system to carry on with the search process in order to obtain *alternative* solutions (Harmon and King, 1985:55).

Using this control strategy, DSS developers could produce systems that make extensive use of knowledge about the goals of the organization they serve, the resources available to achieve those goals, and so forth. Furthermore, these systems will provide decision-makers with *all* the alternatives in any given decision-making situation.

(iv) SUPPORTING DIFFERENT COGNITIVE STYLES.

It was suggested in Chapter 2 that DSS should support the different cognitive styles of users as well as various models of decision-making. It is now hypothesized that if all three ES control strategies mentioned in the previous paragraph are incorporated in a single multi-user DSS, the system will be capable of performing its primary function, namely *decision-maker support*.

The rational decision-maker will be provided with all the alternative solutions to a problem (goal driven approach) from which to choose the "best" one, while the satisficing user will be accommodated by the mixed approach. The user will be given the freedom to choose whether s/he wants to steer the system by volunteering input at any given point in time (data driven), whether the system should carry on independently and only ask for information if it cannot find it in the knowledge base (goal driven), or whether the system should use information volunteered by the user to determine a goal and then query the user for more information while working on the goal.

4.5.3 MAKING DSSs MORE FLEXIBLE.

DSSs can be made more flexible by applying ES technology either to the procedural component or to the modelling component or to both.

(i) ENHANCEMENTS THROUGH PROGRAMMING.

In this area DSSs stand to gain if fifth generation (AI) languages and knowledge representation techniques combined with prototyping are used to develop systems.

(a) Using logic programming.

Logic programming, using for example PROLOG, will enhance the flexibility and adaptability of DSSs. This is mainly due to the following:

(1) PROLOG provides the system developer with the

ability to separate the control components of a program from the specification of the logic (Davis, 1985:53). This separation facilitates ease of change which implies fast modifications (Coelho, 1983) resulting in more flexible systems.

- (2) A PROLOG program has the ability to expand and refine its knowledge base automatically by recording the values of various lines of reasoning and conclusions [Section 4.2(iv) and Cohen (1985:1319)]. This ability to learn from experience makes the system adaptable to changes in the environment. For a more detailed discussion on the different learning paradigms that could be employed to enhance DSSs, reader is referred to the book on machine learning by Michalski, Carbonell and Mitchell (1983).

(b) Logic programming plus prototyping.

Kowalski (1984:94) points out that, although prototyping is gaining field as an alternative to the traditional life cycle approach, "the language used to implement the prototype, is in too many instances one that was designed for the final stages of software development". This makes the development process, and therefore possibly also the developing system, inflexible.

The declarative form in which knowledge is expressed

in languages such as PROLOG, assists in eliminating misunderstandings between user and developer at the earliest possible time in the development cycle. Since logic-based programming is an executable analysis of the user requirements (Kowalski, 1984:93) the full benefits of prototyping, including early elimination of confusion and flexibility during the development phase, will reach the user if it is combined with prototyping.

(c) Development environments.

A more comprehensive solution to the problem, however, would be the creation of a flexible DSS development environment with prototyping a fundamental component thereof. Bonczek, et al. (1983) propose such an environment in their discussion of a decision support system development system (DSSDS). They hypothesize that many new DSSs will be needed in future and it therefore makes sense to focus attention on building a facility that will enable system builders to develop the needed DSSs with greater efficiency than is possible with traditional methods.

Since the problem space for a DSS is continually changing, modifications and extensions of a DSS should be regarded as the norm rather than as an exception (Bonczek, et al., 1983:423). It is further hypothesized that if the DSS development environment displays

the following characteristics, then the DSSs created within that environment will be successful (with flexibility and adaptability the criteria for success):

- (1) Ability to produce DSSs quickly.
- (2) Ability to produce systems with inherent features of modifiability and extensibility.
- (3) Availability of tools that will ensure modifiability and extensibility of produced DSSs.

(ii) IMPROVING FLEXIBILITY BY IMPROVING MODELLING.

There seems to be general agreement amongst researchers that:

- * decision models, like data, are an important information resource that should be managed effectively;
- * a discipline of model management should be developed to help DSS users make more effective use of models; and
- * model management systems should be developed to implement important model management concepts (Blanning, 1983a:397); (Elam, 1980); (Konsynski, 1981).

However, when it comes to implementation methods to achieve these objectives, two schools of thought are clearly discernible. *Firstly*, there are those researchers that merely attempt to apply DBMS principles to model management and by doing so, are achieving objectives 1 and 3 above.

The *second* group also attempts to achieve the objective of assisting users to make more effective use of models by applying AI tools and techniques to model management.

(a) The Relational school.

The work of Blanning (1981,1982,1983a) serves as an example of the first approach, since he draws solely on the relational framework for data management in his efforts to create a model management system (MMS). He characterizes a model as a subset of the Cartesian product of domains corresponding to its input and output attributes. It is suggested that a model bank may be viewed as a set of virtual relations with input and output attributes and functional dependencies between them, just as a database may be viewed as a set of relations with key and content attributes (Blanning,1983:397). As a result, model execution can be considered in terms of the relational "selection" operation and a "join" operation could lead to the utilization of one model's output as input to another model (Bonczek, Holsapple and Whinston,1984:166).

The main advantages of such a relational framework in model management are its simplicity and its close relationship with the well-known relational data model. Since no reference could be found to an experimental or operational implementation of such a model management system, its contribution towards more effective and flexible use of models cannot be reported on.

(b) The ES approach.

The literature surveyed, indicate that the development of effective model management systems that provide users with a considerable degree of flexibility with regards to model usage, is being made possible by ES technology. The work of Bonczek, et al.(1981a) will be used here as representative of this approach to building model management system.

Present day DSSs require from the user either a procedural specification of how a model is to be constructed or the name of a predefined model (Bonczek, et al.,1981b:269). The effect of the first requirement is to make the system difficult to use for the novice user while the second makes the system inflexible. It is claimed that the use of predicate calculus will:

- (1) allow the user to specify his/her needs in a non-procedural manner (Bonczek, et al.,1981b:269) thus making the system easy to use and user-friendly; and
- (2) enable the automatic formulation of a model from a pool of model building blocks and from fragments of metaknowledge pertaining to model usage (Bonczek, et al.,1984:167).

The degree of modelling flexibility offered by a DSS with this capability is a function of the model building knowledge available to the DSS [(Bonczek, et al.,1981b:269); (Dolk and Konsynski,1984:621)]. It is suggested that this

application-specific modelling knowledge be extracted from an OR/MS expert and stored, along with the pool of model building blocks, in predicate calculus form using an extended-network database management facility (Bonczek, et al., 1984:167).

When a request for information implies the application of a model, the system:

- (1) converts the query into well-formed formulas, that is, legitimate predicate calculus expressions;
- (2) invokes the resolution technique (Robinson, 1965) which attempts to "unify" the query formulas with one or more formulas in the knowledge base; and then
- (3) performs a series of database retrieval requests and model procedure invocations to produce the desired results (Dolk and Konsynski, 1984:620).

From the above it could be logically deduced that this approach to modelling will contribute meaningfully towards overcoming some of the inflexibility of DSS outlined in section 3.2.3.

4.5.4 ADDRESSING THE PROBLEMS OF BEHAVIOURAL NATURE.

By striving to improve system reliability, system availability and the man-machine interface in addition to setting the user at ease DSS designers could contribute towards overcoming user resistance. It is hypothesized that DSSs enhanced with AI techniques will satisfy more user

requirements than traditional systems.

(i) IMPROVED MAN-MACHINE INTERFACES.

While previously discussing ways and means of providing DSSs with a more active role in the decision-making process, it was argued that they should be made easier to use by the inclusion of a natural language interface (section 4.5.1). Although "the area of increasingly natural and facile interface languages is one of intense investigation" (Bonczek et al., 1981a:57), the current state of the art in natural language processing is such that the inclusion of a natural language system in the dialogue component of a DSS could have negative effects on the efficiency of the system. This can be ascribed mainly to the ambiguity of, for example, the English language which generally leads to prompting questions by the system in an effort to "understand" the user (Sowa, 1984:288). That is, a certain amount of delay in solving problems is introduced.

Looking at this newly-introduced problem from the novice user's point of view, the advantage of increased system accessibility offered by a (even rudimentary) natural language interface (Finin, Joshi and Webber, 1986:921), could outweigh the disadvantages of a prolonged "conversation" with the system. The more experienced user, on the other hand, might find such conversations tedious (Stewart, 1985:68).

However, since many natural language systems are knowledge-based systems in their own right, capable of learning from experience and of drawing inferences and making decisions in response to free-form input, they could be "trained" by users to understand personalized commands. CLOUT, a natural language companion to Microrim's "R:base" series and other database managers, for example, regards the following two queries as the same question: "Show me a list of employees and when they were hired." and "Employees when hired?" (Stewart, 1985:68). The novice user is free to phrase questions in full natural language, whereas the more experienced user can pose fairly cryptic queries and still receive the correct response from the system.

(ii) SETTING THE USER AT EASE.

In addition to making computer-based systems in general and DSSs in particular, more acceptable to a large group of potential users, natural language user-interfaces "will revolutionize the use of database management and decision support systems by extending their capabilities and by increasing access to what was previously the exclusive preserve of programmers and systems analysts." [(Andriole, 1985:12); (Finin et al., 1986:921)]. In other words, by providing users with a natural language interface to the DSS/DBMS, Andriole predicts that they will feel confident enough to start exploring the contents of the database and in doing so, will become more efficient users of the system.

This correlates with a conclusion reached by Malhotra (1975) that many users, particularly the naive and infrequent ones, need to ask questions to familiarize themselves with the database before asking specific questions to support them in their decision-making task. McKeown (1986:961) also points out that for many users the frequency of system usage is such that it does not warrant the time needed to learn a formal language.

(iii) OVERCOMING USER RESISTANCE.

Although prototyping was discussed in some depth in the section on DSSs inflexibility, its effectiveness in alleviating certain behavioural problems, warrants a revisit at this stage. In a research project comparing the prototyping approach with the more traditional life cycle approach, Alavi (1984:562) found that by following the prototyping approach, more favourable user attitudes toward the design process and the developed system could be achieved.

Furthermore, prototyping increased the actual utilization of the system by users and system performance (as measured in terms of user satisfaction with the output and its perceived accuracy and helpfulness) was rated higher by users of prototyped systems than by users of systems developed by the life cycle approach. The application of ES technology, such as logic-based programming, to prototyping will serve to strengthen these user attitudes, mainly because:

- (a) it is an executable analysis of user requirements;
- (b) it can assist the conventional software development life cycle at the earliest possible stage;
- (c) it allows user requirements to be analysed and executed before a functional specification, design or program is derived; and
- (d) it leads to elimination of misunderstandings at an early stage (Kowalski, 1984:94).

(iv) IMPROVED RELIABILITY AND AVAILABILITY.

It is a fairly general misconception that ESs are inevitably large programs requiring huge computer memories and fast processors. Although this might be true about the majority of currently commercially available ESs, the possibility exists to implement useful systems on microcomputers. For example, PUFF, an ES that can diagnose lung diseases, was originally implemented in EMYCIN on a mainframe. It was rewritten in BASIC and instead of requiring 4 megabytes of storage, it now only needs 64K bytes (Simons, 1984:206).

Furthermore, micros with main memories of 4 megabytes plus, very large storage capacities (100+ Mb), and increasingly faster processors are appearing in the marketplace. Builders of knowledge-based DSSs should therefore continuously evaluate microcomputers for the implementation of systems in order to capitalize on the high availability offered by such systems.

4.5.5 IMPROVING PRODUCTIVITY.

It is suggested that ES technology can contribute towards improving user productivity through the creation of "expert" DSSs which:

- * are made available to the user faster than traditional DSSs;
- * remain usable that much longer;
- * make needed expertise available around the clock; and
- * can be used as intelligent tutors.

(i) SYSTEM AVAILABILITY.

One way to improve the productivity of knowledge workers, such as decision-makers, is to increase the time during which an efficient DSS or an expert to assist them, is at their disposal. ES technology can contribute towards the realization of this objective in three ways:

- (a) Friendly and easy-to-use knowledge-based development environments reduce the time-span from user requirement specification to system implementation (Bonczek et al., 1984:165). That is, users get their support sooner than with traditional development processes.
- (b) Ease of change, due to the use of logic-based programming languages, leads to faster modifications to DSSs when the environment or the user requirements undergo changes. Higher productivity is further achieved by the fact that DSSs remain "current" that

much longer before a major redesign is needed.

- (c) By making the expertise of a highly-skilled worker available to all those workers who rely on his/her judgment and knowledge in the execution of their tasks, a knowledge-based DSS will cut down on time lost due to unavailability of the expert. However, since productivity gains will be a function of many factors, such as degree of dependence on an expert (Johnson, 1984:82), availability of the expert, geographical distribution of the organization and so on, a cost-benefit study is essential before the decision can be taken to enhance a DSS with an expert's expertise.

(ii) TRAINING.

Another reason why ES technology can be regarded as a productivity enhancement tool, is the fact that it contributes towards raising the knowledge level of its users. This is mainly achieved in two ways:

- (a) Extremely user-friendly interfaces are possible which encourages exploration of both database and knowledge-base contents as well as system capabilities (Andriole, 1985:13). This could lead to more efficient usage of available resources by the user community.
- (b) The provision of an explanation facility which allows users to ask why and how a specific conclusion was reached, making the knowledge and experience of the expert who assisted in the construction of the system,

available to the users (Michaelsen and Mitchie, 1983:246). This represents a form of informal training which could only lead to better qualified workers and therefore higher productivity.

AN EXAMPLE OF INCREASED PRODUCTIVITY.

ADVISER, an ICL expert system product, is a high-level language which can capture qualitative experience and then reason with it as if it were hard data. The expertise of an ICL employee responsible for the sizing of transaction processing systems, was captured using REVEAL. The job of sizing, which previously took up to 3 weeks to complete, now only takes about 40 minutes (Vince, 1985:21).

4.6 CONCLUSION.

The major components and characteristics of ES were reviewed in this chapter. This was followed by a comparison of ES and DSS to illustrate the close relationship between the two technologies. Considering that they do not differ on fundamental issues, it is suggested that it is in the interest of the user community that the two should merge.

It was further pointed out that DSSs could become more active participants in the decision-making process, provide users with enhanced support and exhibit more flexibility if they were to be designed and developed along ES lines. Additional benefits to be gained are improved productivity and reduced user resistance.

Before identifying the specific components and architectural structure of such enhanced DSSs the characteristic features of and users' attitude towards intelligent knowledge-based DSSs will be investigated in the following chapter. Possible impacts on the traditional role of decision-makers will also be pointed out.

CHAPTER 5.

EDSS: THE MERGING OF ES AND DSS TECHNOLOGIES.

5.1 INTRODUCTION.

The single most important driving force behind the development of EDSSs is, in author's opinion, demand pull. This opinion is based on the following two characteristics of modern-day society:

- * the availability of small and inexpensive, yet extremely powerful computers; and
- * a changing attitude towards computers.

The so-called microcomputer revolution and the advent of personal computers, have led to the introduction of Computer Studies at primary and secondary schools while at tertiary level there is a tendency in many faculties to include at least a short course in computer appreciation in their curricula. The result is that many entrants to the labour market now view computers as aids to increase their productivity and creativity. For them the computer, like the telephone on their desk, is part of the office furniture.

Over the past two decades middle management and staff assistants to upper management in many organizations have been exposed to, and consequently became used to, the main-frame and/or microcomputers. As these workers get promoted

to higher level positions, they (will) expect assistance from the computer in solving the problems facing them in their new organizational roles. Since moving up the ladder goes hand in hand with a decrease in the structuredness of problems, many of these newly-appointed managers (will) demand smart systems to assist them in the execution of their daily tasks.

In this chapter the emergence of a new generation of intelligent DSSs, called Expert Decision Support Systems (EDSSs) which will be especially equipped to meet decision-makers' demands, is investigated.

5.2 CHARACTERISTICS OF EDSS.

Embedded in the term EDSS are two other well-known terms, DSS and ES. This is no coincidence. An EDSS is a DSS which has been raised from the information processing level to the knowledge processing level through the application of ES technology. An EDSS is therefore primarily a DSS but could, with certain reservations, also be regarded as representative of the class of systems known as ESs.

As was seen in chapter 2, the term DSS has led to serious definitional problems: any system that supports a decision (in any way) could be regarded as a DSS. The problem was solved by giving a fairly concise definition and then fleshing out the definition by means of the essential

characteristics that should be exhibited by such a system. This approach will also be followed in this section. As was suggested in chapter 1, an EDSS could be defined in the following way:

"An EDSS is an interactive computer-based information processing system that draws on a large body of modelling, domain-specific and general problem-solving knowledge to help decision-makers solve ill- and un-structured problems. It is (i) heuristic, (ii) trans-parent and (iii) flexible."

It is hypothesized that this new generation of "smart" decision support systems will display the following characteristics in addition to those typically found in traditional DSSs:

- Intelligence approximating that of humans;
- Natural performance;
- Expertness at problem-solving;
- Powerful reasoning capabilities;
- Dynamic modelling capabilities; and
- Tutoring capabilities.

(a) HUMAN-LIKE INTELLIGENCE.

An attempt to define human intelligence lies outside the scope of this research. Instead, two of the criteria by which it is judged, are discussed below to illustrate that EDSSs could in fact approach human intelligence. The two

measures are: (i) the ability to abstract new knowledge from existing knowledge and (ii) the ability to reason by analogy (that is, the ability to learn from experience in order to improve future performance). It is argued here that the synergistic effect of the last three qualifiers in the definition of EDSS in combination with the large body of knowledge referred to, is to give these systems at least the above mentioned intelligence-defining properties.

(i) THE ABILITY TO MAKE ABSTRACTIONS.

Heuristics, described as compiled hindsight, nuggets of wisdom and contingent pieces of guidance knowledge by Lenat (1983:262,277), were used as judgemental criteria in an ES named AM (short for A Mathematician) to evaluate the worth and effectiveness of problem-solving rules, other heuristics and metaknowledge. It consequently lead to the discovery by the system of new domain facts and new heuristics as well as the modification of existing heuristics (Lenat,1983:243-306). EDSSs, as decision support systems developed along expert system lines, will apply heuristics in similar fashion to the various kinds of knowledge at their disposal. It can therefore be assumed that they too, over time, will grow in knowledge (new facts) (Starrs,1985:38) while simultaneously refining their reasoning powers (new and modified heuristics).

(ii) ANALOGICAL REASONING.

"Being able to learn from experience and apply that knowledge in relevant situations is an important step towards actual intelligence."

[Schank and Hunter (1985:152)]

Any system displaying the ability to transform the solution sequence of a previously encountered problem to satisfy the criteria of a new, related problem, could be termed intelligent, flexible and adaptive. By utilizing logical data structures developed especially for this purpose, such as Schank's (1982:83) memory organizing packets (MOP's), to maintain a well-organized "dynamic memory" of past experiences, EDSSs will be:

- * *reminded* of the existence of a problem that bears a strong similarity to the present problem;
- * able to *recall* the solution to the previous problem;
- and
- * able to *modify* the solution to meet the demands of the current problem.

The above assumption is made on the basis of the success achieved by Lebowitz (1980) with his Integrated Partial Parser (IPP) and by Kolodner (1980) with her CYRUS system. Both systems use MOP's as the basic knowledge representation scheme coupled with generalizations (higher level MOP's to facilitate reminding and retrieval) in their efforts to learn about the world by reading stories from newspapers and the UPI news wire.

For further reading on the topic of intelligent learning systems, a reader is referred to research by Carbonell (1983:137-159), Mitchell, Utgoff and Banerji (1983:163-189) and Hayes-Roth (1983:221-238).

(b) NATURAL PERFORMANCE.

The term "natural performance" is used here in the sense of "performing in a fashion not radically different from humans". For example, EDSSs reason with inexact and incomplete data to reach conclusions with less than certainty, just like humans. They also communicate with users in natural language when requested to explain their lines of reasoning or to justify their conclusions. Natural language is also used to paraphrase, that is, to restate a text so as to convey its meaning in another, perhaps, clearer way in an attempt to ensure reliable conversation (Finin, Joshi and Webber, 1986:921). Furthermore, heuristics and metaknowledge in combination with efficient knowledge representation schemes, enable EDSSs to perform at speeds fast enough to maintain the user's thought processes, thereby making the user as "comfortable" with the system as with a fellow human being.

(c) EXPERTNESS AT SOLVING DOMAIN-SPECIFIC PROBLEMS.

Expertness in the context of problem-solving is defined as the attainment of the same high levels of performance that a human expert normally achieves in the performance of his/her tasks and implies producing high quality results in

minimal time. Since:

- (i) expertness at problem-solving is without doubt directly related to the knowledge level of the problem-solver;
- (ii) EDSSs use a large body of knowledge including, inter alia:
 - Factual/public/surface knowledge
 - Heuristic/strategic/deep knowledge
 - Metaknowledge

and (iii) sophisticated reasoning techniques are applied to the above,

it is expected that EDSSs will outperform most humans at problem-solving within their respective domains. This is not a surprising conclusion if it is kept in mind that EDSSs are DSSs enhanced with ES technology and that ESs have already reached expert levels of performance, exceeding in some cases human performance [(Gevarter, 1983:39); (Starrs, 1985:38); (Brown, 1985:35); (Sowa, 1984:280); (Hayes-Roth, 1984:29)].

(d) POWERFUL REASONING.

Unlike traditional DSSs, EDSSs will have the ability to select the most appropriate problem-solving strategy for the prevailing situation. This is made possible, for example, by the application of metaknowledge to modify procedural knowledge such as production rules or by "short-circuiting" the normal procedures when something about the problem reminds the system of a similar problem that has been solved

previously. In chapters 6 and 7 it is suggested that this short-circuiting should be fully exploited where scenario management is concerned. EDSSs will also accept directives from the user, as to whether backward chaining, forward chaining or a combination of the two should be used.

(e) DYNAMIC MODELLING.

EDSS draws on the vast volume of research that has been done in the area of intelligent model management systems [(Wang and Courtney, 1984); (Dolk and Konsynski, 1984); (Elam and Henderson, 1983); (Blanning, 1983a;1985); (Vince, 1985); (Bonczek, Holsapple and Whinston, 1981a; 1981b; 1982; 1983; 1984)]. Flexibility is the keyword and users will be able to:

- (i) give a procedural specification of how a model should be constructed; or
- (ii) invoke a predefined model by name; or
- (iii) state what information is required and then leave it to the modelling component of the EDSS to dynamically construct applicable and optimized models from a pool of model building blocks, modelling heuristics and metaknowledge acquired from OR/MS experts (Bonczek et al., 1981b:268).

A reader is referred to section 4.5.3(ii) as well as to Bonczek et al.(1981a:358-376) for an example of how modelling knowledge could be operationalized in terms of predicate calculus.

(f) TUTORING CAPABILITIES.

EDSSs will easily be identified by the presence of a knowledge-based training module as one of the major components. It should be noted that this proposed tutoring component is in addition to the explanation facility offered during normal system usage.

While EDSS is primarily aimed at supporting decision-making in an intelligent and expert way, it recognizes the need to increase trainees' and less-experienced users' understanding of, inter alia, the domain, decision-making and the system. Failure to take on this educational role could lead to a situation where important decisions will be made by humans solely on the basis of advice provided by EDSSs. They will have little or no insight into the methodology the system used to arrive at its advice and, as predicted by Starrs (1985:40) for ES users, they "would be unqualified to assess the validity of the reasoning process" of the system. EDSSs are therefore designed with this dual role in mind, allowing users to switch between interactive use-mode and training mode.

For the knowledgeable user the rudimentary explanation facility that can currently be offered, might be sufficient but it is far from ideal for the novice user. It was, however, pointed out in chapter 4 that it is not too optimistic to expect that future ESs (and therefore also EDSSs) will be able to construct rational lines of argument from

fundamental domain principles, tailored to fit the perceived audience. When this is achieved, it will help to improve the tutoring capabilities of EDSSs during normal usage.

An EDSS used in training mode, will compare favourable with stand-alone expert tutoring systems such as:

- (i) BUGGY which teaches the principles of diagnoses (Brown and Burton, 1978); and
- (ii) GUIDON which instructs trainees about how to conduct medical diagnoses [Clancey (1984); Harmon and King (1985:242-244)].

In addition, the tutoring component within an EDSS will base the training that it provides on a model of the user which it will develop automatically during training sessions. A possible way of building this user model is a technique used in the field of Intelligent Computer-Assisted Instruction (ICAI) called the "overlay approach", in which the student's knowledge is represented as a subset of the system's knowledge (Carr and Goldstein, 1977). A simple parameter associated with the representation of each individual concept in the system's knowledge base indicates whether, according to the system's assumptions, the concept is KNOWN or NOT-KNOWN by the user. Another factor associated with each concept indicates the strength of belief with which the system holds its assumptions (Wahlster and Kobsa, 1986:951). In their detailed discussion of an ICAI program called DEBUGGY which is under development at Xerox Palo Alto Research Centre, Harmon and King (1985:240-242) describe how heuristic

analysis could be used to determine the knowledge level of a trainee.

The availability of such intelligent, natural and expert problem-solving systems immediately gives rise to questions such as: "How will users react to them?" and "How will the implementation of EDSS affect the role of the human decision-maker?". The following section describes the ideal view that users should take of EDSSs. It is followed by a section outlining different ways in which an EDSS could be used and the implications thereof for the human participant in the decision-making process.

5.3 USER'S VIEW OF AN EDSS.

The objective of EDSS is to help users solve ill- and unstructured problems. However, as is the case with any software, it could become totally useless if the user does not see the need for it, does not use it or does not appreciate its elegance and power. The user's view of an EDSS is therefore of critical importance. For effective EDSS performance, it is hypothesized that users should:

- Acknowledge the existence of human shortcomings.
- Rid themselves of their fears of computers.
- Accept the computer as "partner".
- Get to know the capabilities and limitations of their systems.

(a) ACKNOWLEDGEMENT OF COGNITIVE LIMITATIONS.

First and foremost, the user should admit to the need of computerized support. That is, the user must acknowledge the existence of human cognitive limitations and the necessity of some form of "mind support", a term coined by Wagner (1981:80). Once this has been done, EDSS has the potential of becoming a kind of auxiliary electronic memory that will complement and enhance the cognitive capabilities of the user by eliminating biases in information processing and counteracting biases associated with the way in which information is presented (Remus and Kotteman, 1986:405-406).

(b) ABANDONMENT OF THE "ROBOT SYNDROME".

Past experience has indicated that humans tend to oppose automation when they suspect that some of their decision-making power may be sacrificed (Henderson and Schilling, 1985:162) or, even worse, that they may be replaced by a machine. EDSSs, as intelligent machines, stand a good chance of being regarded as robots "intent on taking over" the decision-making positions within organizations. The organizational decision-making environment, however, with its multitude of complex interfaces both internal and external, is not seen as a very likely area for robotization. It is stressed that the primary objective of EDSS is still decision-maker support and *not* replacement. Employees should therefore not view an EDSS as competition or as a threat to their organizational position (Andriole, 1985:13).

(c) ACCEPTANCE OF THE SYSTEM AS "PARTNER".

An EDSS and a user form a decision-making "team", they do not compete. For each there is a niche within the organizational structure. Users should keep in mind that one of the driving forces behind the development of computerized systems, is the need of organizations to remain competitive. Since human errors can have a serious effect on competitiveness, researchers are constantly looking for ways and means to eliminate such errors. EDSSs should be viewed as one such attempt and should therefore be consulted for a "second opinion" before important decisions with far-reaching consequences are made. Since the user knows that the system has at its disposal the problem-solving skills of one or more specialists in the domain, s/he should not hesitate to seek its advice or to use it in think-tank fashion.

(d) APPRECIATION OF SYSTEM CAPABILITIES AND LIMITATIONS.

Users should expect EDSSs to perform in a way in which they themselves are only able to work on their best days. That is, the user should expect the system to maintain a level of efficiency which is not possible in humans. If an EDSS does not produce consistent results, or shows signs of degrading performance when presented with difficult cases, or in general does not "live up to expectations", it should be referred to the software maintenance team. The following citation by Hayes-Roth (1984:29) gives an indication of what could be expected from knowledge-based systems:

"It (referring to a specific expert system) never forgets, never fails to consider relevant facts, never jumps prematurely to an incorrect conclusion. In short, the knowledge-based expert system can achieve a level of virtuous professional practice that is superhuman."

However, when formulating his/her expectations, the user should keep in mind that EDSSs are special-purpose systems that cannot be everything for everybody. They are subject, inter alia, to the following limitations:

* **Development trade-offs.** When building an EDSS the developer will, in many instances, be forced to make a choice between a broad but shallow coverage on one side and a narrow, in-depth representation on the other. This will not be the case if:

- sufficient assistance is forthcoming from specialists during the knowledge acquisition phase;
- the user is in no particular hurry to start using the system;
- both development and maintenance staff have the necessary expertise as well as access to up-to-date tools with which to develop or maintain a large and complex knowledge-based system.

* **Performance limitations.** For the time being EDSSs users will have to live with the fact that knowledge-based systems

are not yet capable of determining when a given problem falls outside their domain of competence. This point stresses the need for users to continuously verify the correctness of system reasoning, especially in cases which fall in areas where one domain overlaps with another, requiring problem-solving knowledge from both domains to be solved.

A system displaying the characteristics outlined in the previous section, could be set to task in a variety of ways. The actual way in which it is used will in turn be determined by the user's view of the system. In the following section it is investigated how an EDSS, given a particular user view, will affect the user's organizational role.

5.4 IMPACT OF EDSS ON THE DECISION-MAKER'S ROLE.

This was a topic of interest since the early days of DSS [(Ginzberg, 1978); (Laughran and Cocks, 1977); (Nash, 1977)]. In their study of 18 DSSs in major US corporations, Hogue and Watson (1985:209) found that companies indicated that DSS affect the user's job in two ways:

- (a) it alters the nature of what is actually done; and
- (b) it changes the decision-maker's effectiveness by increasing the number of alternatives evaluated while speeding up the decision-making activity.

It is foreseen that EDSS, in comparison with DSS, will have a significantly greater effect on task nature and user

effectiveness. The size of the impact of EDSS on a specific decision-maker's activities, however, will be partly user-controllable due to the inherent flexibility of EDSSs and is discussed in some depth in paragraph (a) below. This is followed by a discussion of possible effects on:

- The quality of work;
- The balance of power;
- User responsibility; and
- Workloads and employment.

(a) THE EFFECTS OF MODE OF USAGE.

The user of an EDSS decides in what mode to use the system: (i) stand-alone, (ii) user-directed or (iii) consultant mode.

(i) In *stand-alone mode* the user transfers all decision-making activities to the system except those of:

- * initial statement of the problem;
- * acceptance/rejection of the system-proposed solution;

and

- * acceptance of responsibility for the decision reached.

The system will analyze the problem (through its problem recognition facility), build a model of the situation (using its extensive modelling knowledge), generate alternative solutions to the problem, evaluate the alternatives, choose the best solution it has found and present it to the user who can then either accept or reject the solution.

The above is based on the following description of the functioning of an ES: "appropriate segments of the knowledge base are executed, as directed by the inference engine, based upon the control procedures of the inference engine, the domain-specific heuristics and data in the knowledge base and the data describing the current problem under investigation." (Ford, 1985:24). This independent functioning of the system has led Ford to state that the user of an ES is directed by the system.

Used in this way the EDSS becomes a potential decision-making surrogate, assuming the role of the human decision-maker. If all the problems that the user has to face, are solved in this way, and the solutions generated by the system are acceptable, then the organizational role filled by the user has effectively been automated. The impact of an EDSS used in this way is therefore dramatic: possible employee displacement.

(ii) In *user-directed mode* the user will state the problem together with all the information necessary for the system to build and then use a model of the situation. The user identifies either the modules (model building blocks) to be used as well as the order in which they must be combined or a complete, pre-built model. In addition the user must supply the data to be used or explicitly state how the data should be extracted from a data base (Bonczek, Holsapple and Whinston, 1981:49). For alternative solutions this explicit

statement of the computational algorithm and data retrieval steps is repeated. No use is made of the encoded model-building knowledge of experienced OR/MS practitioners which enables the EDSS to operate in stand-alone or consultant mode.

Used in this way, the impact of EDSS on the role of the decision-maker would be comparable to the moderate impact of previous generation DSSs and modelling packages such as SPSS and SAS. Since the system operates only as an assistant that serves some of the user's requests for data retrieval and computation, the user has to perform complex decision-making tasks such as structuring the problem, quantifying objectives, generating alternatives, assigning risks or values to alternatives, choosing among alternatives, and explaining the reasons for chosen solutions (Sprague and Carlson, 1983:111).

(iii) In *consultant mode* the EDSS and the user work together as a team in an effort to derive an optimal solution to the identified problem. After the user has stated the problem, a truly two-way communication channel is activated. A flexible and mixed-initiative dialog then takes place during which "the user can easily take over control from the system at any point to volunteer information or request an explanation." (Kidd and Cooper, 1985:93). The system operates as an advisor who makes recommendations based on information and problem-solving strategies available to it. The user

may override any suggestion made by the system, and if s/he insists on adhering to an "erroneous" decision, the system is loyal enough to "stay with" the user and continue to provide useful analyses and recommendations (Ben-Bassat and Freedy, 1982:483).

When used in this way, the system augments the domain knowledge and problem-solving capabilities of the user thereby increasing the effectiveness of the decision-maker. Instead of using some hit-and-miss technique, the user-system combination can now come up with educated guesses, sound alternatives and acceptable explanations for choices. In this mode it is required of the user to perform some basic, but important, decision-making tasks such as quantifying objectives, guiding the reasoning process of the system, and choosing between alternatives (Sprague and Carlson, 1983:111). These tasks, however, are now performed with far less uncertainty. The net effect of using an EDSS as a consultant is therefore the raising of the level of expertness at which the decision-maker performs his/her tasks.

(b) CHANGES IN THE QUALITY OF WORK.

An important side effect of using an EDSS in stand-alone or consultant mode is that a less experienced employee could become as an effective decision-maker as a more senior and experienced manager. Lower level employees could therefore be trusted with more important decisions, with a resultant delegation of decision-making responsibilities from

higher to lower level workers (Gregg, 1984:240). This represents a form of task enrichment for junior managers while simultaneously meeting the need expressed by some large companies to increase the decision-making power of managers close to the action (Harmon and King, 1985:216). For senior management it would mean an opportunity to concentrate on the more important issues, such as strategic or corporate planning and decision-making.

The above suggests that the author supports the point of view that information technology has a net upgrading effect on job content and job satisfaction. The possibility of deskilling, however, is not ruled out, implying that the skill content of some jobs could decline over time and that the number of workers in skilled jobs could decrease relative to the number in less skilled jobs (Andriole, 1985:13).

(c) SHIFT IN THE BALANCE OF POWER.

According to Harmon et al. (1985:216) a number of Fortune 500 corporations have expressed the desire to increase the centralized control and monitoring abilities of senior corporate officers. It is predicted that EDSS will be the vehicle through which this objective will be reached and that there will be a resultant upward shift in the locus of organizational political power. By providing senior managers with instant access to specialized, carefully tailored advice in all areas under their control, EDSSs will

amplify the skills and expertise of such managers (Turner, 1985:14) while simultaneously reducing their dependency on inputs provided by lower-level managers. This in turn will lead to less opportunity for middle managers to influence decisions taken at higher levels.

(d) INCREASED RESPONSIBILITY.

It is the responsibility of every EDSS user to continuously verify the correctness of system functioning. In organizational terms, this means that every EDSS user must assume a supervisory role and should be held responsible for all the decisions made by the man-machine team. The increment in responsibility will be directly related to the mode in which the EDSS is used with stand-alone mode causing the largest increase.

(e) DECISION-MAKER WORKLOAD AND UNEMPLOYMENT.

In an effort to increase profits (combat losses) many American companies are attempting to reduce their layers of middle managers (Harmon and King, 1985:216). The implication of this is that some middle managers are being asked to be responsible for overseeing and controlling an increased number of activities and subordinates whereas others are losing their jobs. EDSS may well prove to be indispensable for both these categories.

For those asked to monitor a larger portion of organizational activities EDSS could provide the following assis-

tance:

- gather and intelligently organize the increased volume of information since it is knowledgeable about the domain;
- reduce the time needed to analyse operational data based on its extensive modelling capabilities;
- temporarily take over some decision-making tasks by being used in stand-alone mode;
- train the manager in those areas in which he is not well-versed; and
- assist in the development of improved methods of solving problems (see section 6.2).

For those unfortunate enough to lose their managerial jobs EDSSs could assist in re-educating or re-training them. The powerful tutoring capabilities of EDSSs will make them ideal personalized educational tools that could help reduce one of the major obstacles to the evolution of the developed nations towards a postindustrial society - the discrepancy between the average educational level required by commerce and industry, and the average educational level prevailing in the available workforce (Starrs, 1985:40).

5.5 CONCLUSION.

When analyzing the characteristics of the proposed EDSSs, it becomes evident that it is irrelevant whether ESs are modified to meet the objectives of DSS or whether DSSs

are enhanced with ES capabilities. This closeness of the two technologies involved, has inspired the chapter heading "EDSS: THE MERGING OF ES AND DSS".

The explosion of data - three times the rate of expertise, according to Hayes-Roth (1987:86) - is prompting a transition from strictly expert systems to more generalized knowledge systems, of which EDSS is an example. Along with the rise of the knowledge system, Hayes-Roth (1987:87) predicts the emergence of a new kind of profession: intelligent systems engineering. These professionals will assist in the design and development of a new generation of intelligent computer systems that will "begin to appear in the next year or two and will proliferate in the 1990's" (Harmon and King, 1985:236).

To provide some sort of prescriptive guidance to this new breed of systems engineers as well as to stimulate further EDSS research, the following chapters are devoted to a discussion of possible EDSS components and architectural frameworks for developing such systems.

CHAPTER 6.

EDSS COMPONENTS.

6.1 INTRODUCTION.

In order to design systems that display certain characteristics the designer needs a conceptual framework of the system to be designed. That is, some guidance is needed as to what components to include in the design as well as how these components should be interfaced. The objective of this chapter is to determine those components that are necessary for a system to display EDSS characteristics.

DSS, like all software systems, are developed to meet specific user needs (Bonczek, Holsapple and Whinston, 1981:19). Because of the direct relationship between user needs and system features it is suggested that the search for system components should be guided by both empirical needs and conceptual requirements.

6.2 THE IDENTIFICATION OF COMPONENTS.

Sprague and Carlson (1982:33) identify three major DSS components: Dialog Management, Data Management and Model Management. Since EDSS is regarded as a step along the evolutionary path of DSS it is suggested that these traditional components should be retained as the core components of EDSS. In the following three subsections, however, it is

argued that they should be enhanced through the application of ES and NLP techniques (see also Chapters 4 and 5). The section concludes with the identification of and motivation for knowledge-based training, scenario, tool and communication/control components.

6.2.1 THE DATABASE COMPONENT.

A DSS database is a collection of organizational and environmental data representing facts. In order to make decisions the user accesses and manipulates the data via a database management system (DBMS).

However, a number of problems with traditional DBMS have been identified by eminent researchers. According to Wiederhold (1984:64) these problems center mainly around the user-interface and response time. Complex queries are difficult to phrase in the formal query language provided with a DBMS, queries submitted by users often specify inefficient search paths and system responses to queries may be ambiguous. Jarke and Vassiliou (1984:77) postulate that conventional database systems are lacking in capabilities such as reasoning power which will allow for a wider range and more concise formulation of queries. Another difficulty with many database management systems is the lack of support for operations performed in context while a user who wishes to issue a query that is based on the result of a previous query has to store that result explicitly. Lastly, Wahlster

(1981) complains that database systems do not have a partner model of the user. These and other related problems were discussed more fully in Chapter 3.

To eliminate some of the identified shortcomings, it is proposed that Data Management in an EDSS should be enhanced in the following three ways:

- (i) the database component must be complemented by a knowledge base;
- (ii) the processing component responsible for, inter alia, searching, retrieving and updating must be given deductive reasoning powers; and
- (iii) data extraction from external databases must be knowledge-based.

(i) KNOWLEDGE BASE.

The objective of the knowledge base is to provide the system with access to a wide variety of knowledge which will enable it to operate at a higher level of abstraction for both its input and the results that it produces. Wiederhold (1984:65) identifies nine categories of knowledge which could be included in the knowledge base of an Expert Database Management System (EDBMS). They are:

- enterprise directing knowledge;
- focus management knowledge;
- application-specific knowledge;
- general procedural knowledge;
- structural knowledge;

- derived knowledge;
- data-domain knowledge;
- system knowledge for query and response control; and
- resiliency and recovery support knowledge.

Each category can be shown to contribute towards enhancing DBMS capabilities. But it is when different levels of knowledge interact, that systems of greater power can be created. For example, by combining structural and focus-oriented knowledge (that is, knowledge about dependencies and constraints among data together with knowledge about the current intent of the user) query ambiguity can be resolved and in the process the user is spared a semantically redundant interrogation (Wiederhold, 1984:69).

Referring to a bibliographic retrieval system implemented by F. Corella et al. (1984) in which procedural, structural and domain knowledge are applied to the results of queries in order to present the user with meaningful answers, Wiederhold (1984:64) uses the following example query to illustrate:

```

SELECT name FROM employee
      WHERE age > 50 AND
            salary < average
              (SELECT salary FROM employee
               WHERE department = 'Sales')

```

A response of NIL could, inter alia, mean that there are no employees older than 50 or that there are no employees in

the sales department. Assume that the second possibility is the case. The NIL response triggers a procedural advice rule which, by using structural knowledge, causes the location of the 'failure' to be identified. Since an intermediate query returned a relation with cardinality 0, the appropriate response: There is no department = 'Sales' is generated.

(ii) DEDUCTIVE PROCESSING.

While knowledge management in ES is geared towards dealing with a relatively small number of instances of a much larger variety of types and classes of knowledge it lacks a DBMS's ability to efficiently search gigabytes of data. On the other hand, current DBMSs have been optimized to deal with masses of data but were never designed to cope with the kinds of knowledge proposed in the previous few paragraphs [(Kellog, 1986:76), (Sowa, 1984:303), (Jarke and Vassiliou, 1984:70)].

Since the application areas of EDSS are of the less structured type and require a combined data/knowledge management facility which efficiently performs all the functions outlined above, a marriage between data management and knowledge management is proposed. The inferencing techniques required have already been developed making it possible to construct deductive processors or reasoning engines that can support the development of practical knowledge-based DBMSs (Kellog, 1982).

(iii) EXTERNAL DATA EXTRACTION.

EDSS represents a marriage between traditional DSS and ES. Both partners rely heavily on access to a "database" but, as was pointed out above, ES databases differ from traditional databases in the sense that they store complex properties about a few entities whereas conventional databases store simple properties about a large number of entities. The inclusion of a knowledge-based DBMS is seen as an effort to reconcile these differences.

However, when the situation arises where the EDSS needs access to a corporate or other database outside the system boundary it will probably have to communicate with a conventional DBMS. Jarke and Vassiliou (1984:70) then propose two alternatives, namely loose and tight coupling. *Loose coupling* implies the presence of a channel between the two systems which allows for data extraction from the external database and subsequent storage of the data as a "snapshot" in the EDSS database. This approach is only feasible if:

- (a) the external database is fairly static;
- (b) the user knows in advance which portion of the external database is required for extraction; and
- (c) the EDSS does not require such extractions frequently within a single session.

All that is needed in such cases is a facility to automatically generate "expert" database entries from the extracted data (Jarke and Vassiliou, 1984:75).

Tight coupling on the other hand refers to a situation where

the external database appears to the EDSS as an extension of its own internal database. This implies that the data extraction component must be an ES in its own right. Jarke and Vassiliou (1984:72,75-76) propose that, before accessing the external database, the extraction facility should:

- (a) collect a number of requests for data;
- (b) transform the ES type database commands into traditional database queries; and then
- (c) optimize these queries.

A similar approach is proposed by Li (1984:127-130).

The main benefit to be derived when applying knowledge to the database component is the transformation of the DBMS into a knowledge system which "directly produces information suitable for decision making and planning" (Wiederhold, 1984:64). This is made possible by, for example, the extraction of information that is not explicitly stored but can be inferred by combining specific facts with encoded rule-based knowledge. Other benefits are:

- (a) the possibility to extend and adapt the DBMS language to the needs of particular users;
- (b) the generation not only of answers but also of evidence for or against these answers;
- (c) the ability to handle what-if and other kinds of high-level queries that are difficult if not impossible for present-day DBMSs;
- (d) support and enforcement of complex semantic integrity constraints (Kellog, 1986:77-78); and

(e) the exploitation of redundancy to the advantage of the user (Buchanan and Duda, 1983:204).

6.2.2 THE MODELLING COMPONENT.

According to Dolk and Konsynski (1984:619) management usage of sophisticated modelling techniques designed for decision making has been much less than desired as a result of unawareness of the existence of models, low-level knowledge about the applicability of models to the situation at hand and onerous interface requirements. They propagate the idea that a model management system (MMS) should provide "a structured milieu for storing, manipulating, and retrieving models".

In a survey undertaken by Uliana (1986:118,126) users indicated that they want more simulation and more interactive modelling while simultaneously indicating that they are not satisfied with their level of quantitative methods knowledge. A MMS incorporating the know-how of OR/MS experts could serve either as an intelligent assistant in modelling and/or simulation activities or could be called upon to train the user in the techniques about which it is knowledgeable.

Embedding the knowledge about models in the processing component of a DSS as frequently happens in the case of traditional DSS, introduces a considerable degree of system

inflexibility thereby adding to the user's problems. Compare Bonczek, Holsapple and Whinston (1981:269):

".... alterations or additions of models necessarily require changes to the processing component."

The inclusion of a knowledge-based modelling component as a subsystem in an EDSS results in a DSS processing component [called an "interrogation component" by Elam and Henderson, (1983:114)] which is completely invariant to changes in the modelling knowledge within a specific application area (Bonczek et al., 1981:269). The modelling subsystem performs its functions under the guidance of domain-specific knowledge as well as general modelling knowledge in a modelling knowledge base. The existence of an intelligent MMS developed by Elam et al. (1983:114) is proof of the practicality of this approach. A reader is also referred to the pioneering work by Bonczek, Holsapple and Whinston (1981).

It is foreseen that both qualitative and quantitative data will be available in the proposed knowledge-based database component. Subjective information might well be mixed with objective information and data could possibly be unreliable as well as redundantly represented (Wiederhold, 1984:68). Traditional DSS using quantitative methods only will not be able to cope with this situation. On the other hand, a knowledge-based MMS using inferencing, heuristics and models of inexact reasoning can make good use of qualitative, sub-

jective and redundant information. This argument is supported strongly by Buchanan and Duda (1983:204).

The main objective of an expert model management system (EMMS) is therefore to provide the user with a flexible and easy-to-use interface to quantitative methods/management science techniques while extending the range of support to the qualitative level. A secondary, but no less important objective is to "institutionalize" the modelling knowledge of organizational experts and then to use it for decision making as well as for training.

The addition of a knowledge layer to the modelling component of DSS represents a move in the direction of providing the user with the "structured milieu" referred to at the beginning of this section. It could contribute towards an increase in user efficiency and knowledge, modelling flexibility, and decision effectiveness.

6.2.3 DIALOG MANAGEMENT.

Although there is general agreement that the user-system interface is an important aspect in terms of user acceptance of a system, Bonnet (1985:60) still finds it necessary to say that "there are many examples of top managerial staff who have terminals on their desks which they seldom use, simply because too little attention has been paid to making the system friendly." This highlights

the user that may be relevant for the dialog behaviour of the system (Wahlster and Kobsa, 1986:949). Examples of such ES type intelligent front ends are:

- ASA (O'Keefe, 1982), which helps a psychologist find a suitable statistical analysis method for his/her experiment.
- REX (Gale and Pregibon, 1983), which assists in the use of a statistics package.
- GRUNDY (Rich, 1979), which recommends novels that the user might like to read.

The user modelling component should incrementally build up a user model and store it for use during subsequent sessions. It should also supply other components, for example, the explanation subsystem, with assumptions about the user (Wahlster and Kobsa, 1986:949).

While an EDSS is viewed as a far more active participant in the decision-making process than traditional DSS, the user is still responsible for the actions recommended by the decision-making team. An interactive, intelligent user-interface to keep the user well-informed of the inferencing (coherent analysis) being performed by the system is therefore an absolute necessity. That is, the user must be placed in a position to ask enough questions to follow the line of reasoning and it should be made easy for the user to assess what the system does and does not know (Buchanan and Duda, 1983:200).

Elam, Henderson and Miller (1980) argue that support for problem structuring is even more important than support for the analysis and choice phases of decision-making because one clearly cannot derive correct solutions if the problem is incorrectly formulated. In this they are supported by empirical evidence gathered by Uliana (1986:118,126). Problem recognition and problem structuring require the application of AI techniques such as semantic analysis and focus management [(Bonczek, Holsapple and Whinston, 1981); (Schank, 1984); (Wang and Yu, 1983); (Schank and Slade, 1984)].

Finally, it is suggested that the IFE should include a natural language component in order to gain one or more of the following advantages:

- (a) *Flexible dialog*: the user can switch from one dialog mode to another, whichever is convenient.
- (b) *Mixed initiative*: either party can ask a question or state a fact at any time.
- (c) *Inference*: short and simple user sentences and the system fills in the "obvious" gaps.
- (d) *Context*: the system remembers the current topic.
- (e) *Knowledge base*: the range of topics can be extended without writing detailed procedures.
- (f) *Metalanguage*: to talk about the subject or what can be said about the subject. (Sowa, 1984:293)

In a multi-user EDSS environment where the nature of the task to be performed is not well-specified, some users might

not know the capabilities or limitations of the system while others might not be willing or capable of learning a formal interface language. For such users an intelligent front end as proposed in this section will certainly contribute towards making the system more accessible.

ADDITIONAL COMPONENTS.

As was pointed out in Chapter 3 section 3.2.2(iv) the author subscribes to the viewpoint that DSSs exist to support decision-makers (that is, distinctive personalities playing out specific institutional roles), in their decision-making tasks. EDSSs are therefore also considered to be *decision-maker* support systems, implying that they should support the user in all decision-related activities. In order to perform this task successfully it is proposed that EDSSs should include additional components that will provide the user with:

- (i) support to become knowledgeable about the application domain, the task of decision making and the support system;
- (ii) support to explore alternative ways of problem-solving;
- (iii) support to tailor the system to his/her specific needs.

6.2.4 A TRAINING COMPONENT.

"... the final system will evolve only through ongoing interaction of designer and user, learning, personalized use, or the evolution of new functions." - Keen (1980a:11)

Since the conception of DSS it has been stressed that these systems should be developed using an adaptive (Keen, 1980:9), evolutionary (Moore and Chang, 1983:185) or middle-out (Hurst, Ness, Gambino and Johnson, 1983:124) approach. One of the basic underlying characteristics of these approaches is the learning process followed by the user.

The learning referred to is not just a case of acquiring some simple skills to use the system but is more a question of acquiring an in-depth understanding of the decision making system (Hurst et al., 1983:129). Stabell (1983:224) describes the end result of this learning as "active understanding" implying that the user knows the meaning and the uses of the different DSS functions in terms of the different stages of his/her own decision process.

Hurst et al. (1983:128) consider the education and training of all the people involved in developing and using a DSS to be more important than the actual system and its output. Stabell (1983:225) pleads for a learning-oriented approach

towards system development and support for the learning process which should ideally carry on through the entire life-cycle of the system by stating that:

"Increasing decision-making effectiveness through changes in how decisions are made should be a principle objective for DSS development."

Henderson and Schilling (1985:166) states that a DSS should provide the user with decision aids within an integrated, yet adaptive system. One such decision aid is the collection of quantitative techniques required by the user for the modelling aspect of his/her decision making tasks.

However, in a survey conducted by Watson and Marett (1979:124) it was found that one of the more frequently cited problems in using quantitative methods was that neither top nor middle management had the educational background to appreciate such methods (which includes modelling). Negoita (1985:119) claims that managers cannot apply models without the aid of specialists whereas Wegner (1983:122) found that adequately trained quantitative methods people are in short supply. As a result computer modelling has had very little impact on top-level management decisions. According to Bonini (1978:53-54) the reasons for the low impact are related not so much to methodology as to lack of understanding about how to use models and how to deal with the modelling process.

In the survey conducted by Uliana (1986:125-129) the current

and desired levels of education in quantitative methods amongst, inter alia, managers in a large number of public companies were tested. The results led Uliana to conclude that a full understanding (of the techniques used) appears to be the focal point to which management aspire in their quantitative methods knowledge. This finding is in line with the Watson and Marret (1979) conclusion that "most managers would rather live with a problem they can't solve than use a solution they don't understand."

This need for training is also found in related fields such as ES, where many systems have been developed to provide advice on how to use complex systems [(Buchanan and Duda, 1983:203), (Hewitt and Smith, 1975)] or to tutor a novice in the use or understanding of a body of knowledge [(Buchanan et al., 1983:203); (O'Shea, 1979), (Brown, Burton and deKleer, 1982)].

A component that will teach users the basic domain concepts, how to make more effective decisions, how to build and when to use models, and, in general, how to use the system both effectively and efficiently, will prepare the user to make better use of the support the system can provide in the decision making situation as such.

6.2.5 A SCENARIO COMPONENT.

It is not unrealistic to expect that decision makers (for example, managers) will frequently be interrupted while in the process of solving a problem (making a decision). An activity that could be interrupted is that of model building, which, like computer programming, can be a very complex and time-consuming task. The user stands to benefit by a system that automatically records the development steps. The user can then retrace these steps either to check their validity or to re-familiarize her/himself with an interrupted problem-solving process.

The importance of such a facility was accentuated by Scott Robertson, general manager of management science at Ore-Ida Foods during the INFO 83 conference when he said that models must be updated frequently, as market conditions change and as new relations are perceived (EDP Analyzer, July 1984:2).

Obviously, the user needs an audit trail of changes made to complex models so that in the event of a modified model producing unlikely or unacceptable solutions, the user can "back-track" through the modification steps in search of the "bug".

Each DSS model has many hidden assumptions and after a short time the creator forgets some of these (Howard Morgan at a

conference, EDP Analyzer, 1984:4). If such a model is used months later, some of the assumptions may be unknowingly violated, leading to erroneous results. In similar vein Stabell (1983:251) states that a DSS should provide functions for recording and revisiting key decision assumptions. These remarks are substantiated by Sprague and Carlson's observation that decision makers need memory aids (Sprague et al., 1982:104) and by the findings of Moore and Chang (1983:196) that their simple and novel "documentation-by-example" feature for the automatic generation of documentation for user specified models served as a handy memory aid.

It is, furthermore, generally accepted that a DSS does not solve a problem but that it lets individuals exploit their own skills in problem solving (Keen and Gambino, 1983:152). The decision maker has a mental model of the problem to be solved and uses quantitative methods to reach some solution to which s/he then applies her/his own judgement (Affleck-Graves, Money and Uliana, 1987:13). If an acceptable solution is not found, the user needs support to gain deeper insight into the problem, the relationships that exist between inputs, the effects of constraints and so on. In this regard DSS should provide the facilities to perform a sensitivity analysis, the results of which should ideally be reproducible at a later stage without having to rerun the model. The need for this kind of scenario generation (and management) is implied by Medsker (1984:1123):

"Information processed at one stage should be available at another stage without having to be re-analyzed. Nonetheless, the user should have the option to re-analyze if it becomes desirable to include previously unused information."

In a discussion of his adaptive framework for DSS, Keen (1980:13) states that an effective DSS encourages the user to explore new approaches to the task. The task referred to is decision making which, in many instances, is itself exploratory in nature (Thierauf,1982:62). In the author's opinion the system should provide the user with adequate support in finding such new approaches as well as in documenting all the "routes" that have been explored. In this respect Reitman (1984:6) suggests that Artificial Intelligence can make a useful contribution.

In the light of the above it is suggested that a scenario generation and management facility be included as a separate component of EDSS. The objectives of this component should be to act as a run-time historian and to provide expert support in the generation of both alternative solutions and new approaches to problem-solving.

6.2.6 A "TOOL BOX".

One of the basic requirements of DSS is that they should be easy to use (Ittmann, 1984:190). In this respect software developers in the microcomputer world have made notable progress. For example, products such as PFKEY, SMARTKEY and other keyboard enhancers allow a user to assign a frequently used sequence of keystrokes to a single key thereby reducing the number of key depressions required to achieve a specific result. It is foreseen that the widespread use of microcomputers will lead to a situation where the user will come to expect a selection of such easy-to-use tools with which to tailor the user-system interface to his/her needs.

Keen (1981:8) reports on the importance of communication between those involved in decision-making implying that the support provided by a DSS should include tools that will enable the user to communicate his/her decisions and the justification thereof, to those who need to know (See also Henderson and Schilling, 1985:164). Communication tools include both hardware and software. For example, a modem and the software required to automatically dial a number and transmit information to another computer or a graphics printer and/or plotter and some easy-to-use graphics generation software. An example of the latter is the Harvard Presentation Graphics package for microcomputers. In an edition devoted to DSS, EDP Analyzer (November, 1983:4)

reports on a case where a company had to analyse a proposed multi-million-dollar supply base project and concludes that the results, displayed in colour graphics form and captured in colour slides, played a significant role in helping management and investors decide to go ahead with the project.

In his adaptive framework for DSS Keen (1980a:12) labels the Builder-System link as the evolution loop. Because of managerial learning and personalized uses the existing system comes under strain which then builds pressure for evolution. In response to this pressure the builder adds new capabilities to the system which, according to Keen (1980a:17) is only feasible if "the designer maintains ongoing contact with the users".

However, when considering the rapid growth in end-user computing and the wide-spread use of fourth generation languages and assuming that:

- (a) it is easy for the user to define new operators, data constructs and commands (a situation which, according to Moore and Chang (1983:187) will automatically follow if their approach based on evolving system capabilities is followed);
- (b) the system is easy to modify to accommodate requirements for new functions, commands, models, and data as the use of the system evolves (Stabell, 1983:225) and

(c) the DSS contains user-friendly tools with which to develop and implement these new capabilities then the author is of opinion that by the time the technology needed to build true EDSS has permeated through from research to industry, many users will be ready (and eager) to try their hand at entering their own "expertise" into an EDSS knowledge base (after all, it is their system!). Tools that should be considered for inclusion are, inter alia, an integrated set of ES development tools including a full-screen editor, programming and knowledge representation languages which are compatible with those used to develop the EDSS and a sophisticated inference engine.

In the process of building up a case for his decision-oriented approach to building DSS Stabell (1983:225) criticizes the representation-centered designs of Carlson (1983:15-40) and Bennett (1983:41-63) as well as the verb-oriented design of Keen and Gambino (1983:133-173) by saying:

"It is not clear what to do to amplify the user's problem-solving capabilities."

It would appear as if the provision of a comprehensive toolkit that contains powerful, easy-to-use and flexible tools could enable the user to tailor the system to better fit his/her unique way of solving problems and to add new functions as well as new knowledge to the system. The net result is likely to be the enhancement not only of the decision maker's problem-solving capabilities but that of

the total decision making system.

6.2.7 A COMMUNICATION AND CONTROL COMPONENT.

As a result of the large number of identified components, an integrating component becomes necessary to facilitate inter-module communication and synchronization (Jarke and Vassiliou, 1984:81). The absence of such a component could lead to a maximum of 2^n inter-module connections thereby increasing system complexity.

An added benefit of an integrating component is that the different components could be developed as separate systems which execute on different processors thereby exploiting any possibility of parallel processing. The addition of new components is also simplified since such new components need only communicate with one other software system, namely the integrating component.

6.3 CONCLUSION.

This chapter has defined the components of an EDSS. These are necessary if the user's needs are to be met. These needs are to:

- be supported during all the phases of decision making;
- understand the system and its workings;
- be trained in quantitative methods, decision making and domain principles;

- view a problem using different decision making theories;
- apply behavioural models in addition to numerical models to the decision making process;
- have access to systems that will evolve faster in an effort to meet their requirements;
- perform flexible and interactive modelling;
- specify their problems in a variety of dialog styles;
- control the way in which output are presented;
- have access to responsive and available systems;
- explore alternative ways of solving problems;
- be more productive;
- institutionalize proven problem-solving techniques; and
- reduce training costs.

It should, however, be kept in mind that the degree to which these requirements are met will be determined by the amount and quality of the built-in knowledge in each component [(Barr, Cohen and Feigenbaum, 1981), (Buchanan and Duda, 1983:184), (Sowa, 1984:280)], the way in which this knowledge is structured and, most important, the way in which the components are organized to form a comprehensive support system.

To give substance to the ideas put forward in this thesis and to provide some prescriptive guidance to designers and developers of the new generation DSS that has been outlined, the following chapter is devoted to the development of alternative architectural frameworks for EDSS.

ARCHITECTURAL FRAMEWORKS FOR EDSS.

7.1 INTRODUCTION.

Having identified the principle components of EDSS, the next logical step is to combine these components in an architectural framework to illustrate their interactions and to provide designers and users with a clear conceptual view of EDSS.

The objective of this chapter, therefore, is to develop a framework that will allow designers/developers to create systems that are modular (adaptive, extensible, evolutionary) and which will facilitate the use of rapid prototyping. Both of these requirements are made necessary by the unique problem environment of organizational decision-makers which is aptly described by Hurst, Ness, Gambino and Johnson (1983:123) in the following way:

"In all aspects of a DSS - users, technical tools, organizational needs - the one unchanging fact is that change will occur throughout its life."

Since the problems on which they work also change, users cannot always specify in advance what exactly their requirements are and even when specified, can change at a moment's notice.

Two alternative frameworks are presented. The one is an adaptation of the Bonczek, Holsapple and Whinston (1981a:72) generic framework for DSS, the other an original EDSS design.

7.2 THE GENERIC APPROACH.

7.2.1 ARCHITECTURAL FRAMEWORK.

Bonczek, Holsapple and Whinston were amongst the first researchers to recognize the advantages that could be gained by enhancing DSS with Artificial Intelligence techniques (Bonczek et al., 1981a). The generic framework for DSS as proposed by them and depicted in Figure 7.1, views a DSS as having three principle components:

- (i) a language system (LS);
- (ii) a knowledge system (KS); and
- (iii) a generalized problem-processing system (GPPS).

The framework is restrictive enough to exclude all systems which are not knowledge-based while on the other hand it provides enough "capacity for extensive elaboration" (Bonczek et al., 1984:147). For instance, the proposed framework in conjunction with the definition of EDSS given in Chapter 1, should prove to be sufficient when trying to classify a system as being an EDSS or not. The Minch and Burns (1983) conceptual model for DSS (Figure 7.2) serves as an example of the elaboration capabilities of the framework.

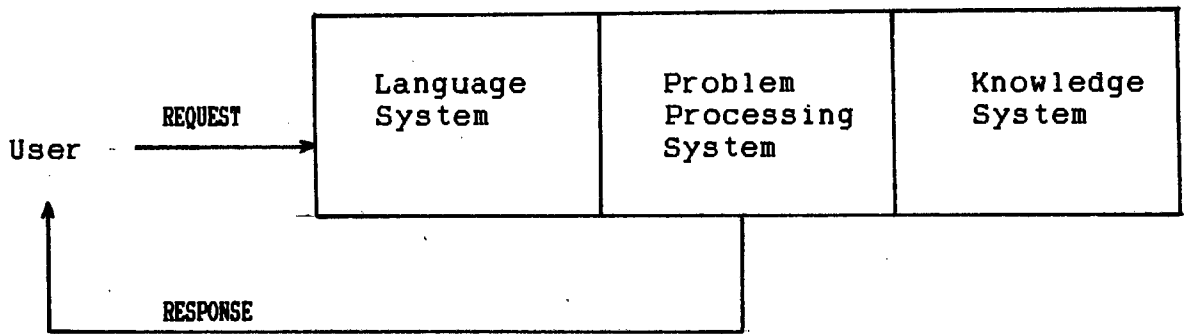


Figure 7.1

Decision Support System

Bonczek, Holsapple and Whinston (1981:77)

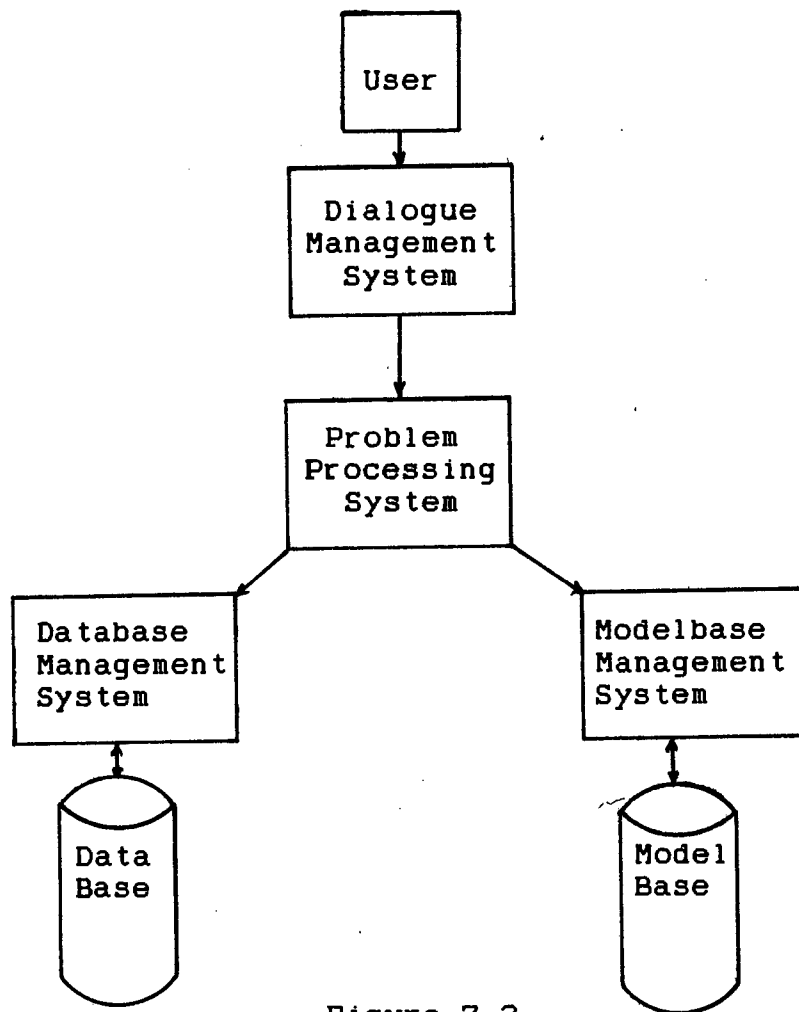


Figure 7.2

Decision Support System

Wang and Courtney (1984:302)

However, in the author's opinion this framework is more conceptual than architectural. After pursuing the ideas of Bonczek et al.(1981a) as outlined in their book "Foundations of Decision Support Systems" it was possible to construct a more detailed representation of their original framework. This architectural framework is depicted in Figure 7.3.

Assuming that the components identified in the previous chapter are indeed necessary and sufficient to uniquely identify a system as being an EDSS, an examination of Figure 7.3 reveals that it does not include those components that differentiate EDSS from traditional DSS or other knowledge-based systems such as ES. To be representative of EDSS and to provide system designers/developers with some form of guidance, the generic framework of Bonczek et al. (1981a) needs extensive modification. Figure 7.4 illustrates one way in which it could be changed to become descriptive of EDSS.

7.2.2 DESCRIPTION OF THE GENERIC FRAMEWORK.

What follows is a description of the original Bonczek et al.(1981a) framework with indications of why and how it should be modified to become acceptable for EDSS.

(i) THE LANGUAGE SYSTEM (LS).

Bonczek et al.(1981a:70) defines the LS as "the total of all linguistic facilities made available to the user of a decision support system". It serves as the communication

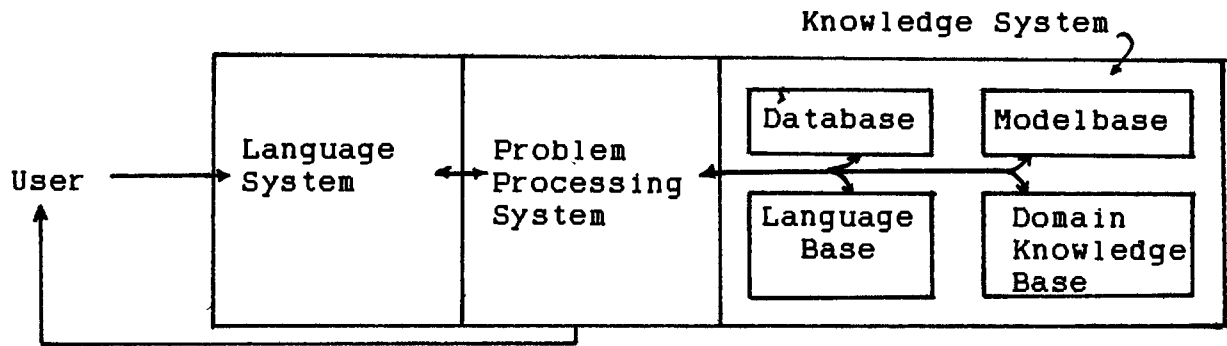


Figure 7.3

A modified version of Figure 7.1 showing more detail.

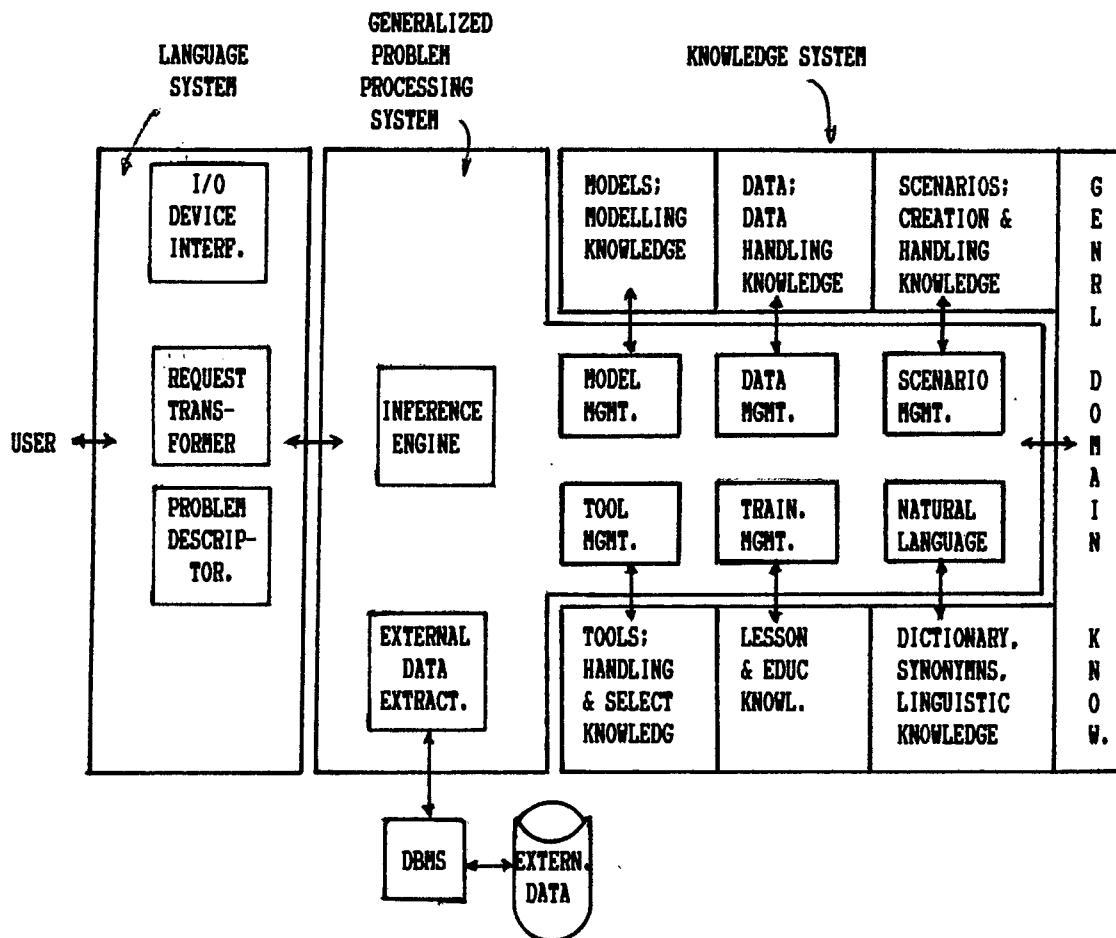


Figure 7.4

The Bonczek et al. model of DSS modified to represent EDSS.

interface between the user and the system and plays a key role in the effectiveness of DSS. All requests by the user for data retrieval and/or computations are stated using the grammar of the LS, that is, using the commands, expressions and statements allowed by the LS. According to discussions of the generic framework LS (Bonczek et al., 1981a, 1981b, 1984), it appears to be very user-friendly. For example it:

(a) may encompass either retrieval languages or computational languages or both. These languages may be characterized as covering a range between two extremes. At one extreme we find languages in which the user explicitly states how data is to be retrieved or how calculations are to be carried out. At the other extreme the user simply states the data desired or the problem to be solved without any indication of how data should be retrieved or how the problem is to be solved (Bonczek et al., 1984:148).

(b) contains a dialog style or a combination of dialog styles.

The above description clearly indicates that the LS of the generic approach is a knowledge-based, user-friendly man-machine interface. However, no reference to the building and using of *a model of the user* could be found in the available literature. The LS of an EDSS needs such a model to tailor its responses to fit the user.

(ii) THE KNOWLEDGE SYSTEM (KS).

The KS of a DSS is the body of knowledge that it has about its problem domain and according to Bonczek et al. (1983:149) the usefulness of a DSS depends to a large extent on the amount of application-specific knowledge it possesses. Whereas the framework of Sprague et al. (1982) concentrates only on the two traditional types of knowledge, namely models and data, the KS of the Bonczek framework is far more comprehensive. As illustrated in Figure 7.3 it consists of:

- (a) a data base;
- (b) a model base;
- (c) a linguistic base; and
- (d) a domain knowledge base.

As far as contents is concerned the above is in accord with the decision-oriented approach as advocated by early DSS researchers. However, as was suggested in Chapter 6, the boundaries of DSS should be adjusted to make *decision-maker support* the first priority. In addition to the above-mentioned knowledge bases the KS of an EDSS should also include a tool base, a scenario base and a training base.

Although shown above as separate entities, it should be noted that in Bonczek et al. (1981a:335-353, 358-376) the different classes of knowledge are represented in the form of predicate calculus within one comprehensive knowledge base. For the reasons given below, it is suggested that knowledge in an EDSS KS should be partitioned into logically

related "chunks".

(a) Partitioning will facilitate ease of modification to meet user's need for adaptive systems over time.

(b) The knowledge in each component is specialized to the extent that it requires the efforts of different experts to create these knowledge bases. For example:

(i) In addition to models the model base contains

modelling metaknowledge (knowledge about knowledge) to indicate for instance how to use modelling knowledge in conjunction with empirical knowledge (Bonczek et al., 1984:150), or which models to use under which conditions, clearly indicating the involvement of a quantitative methods expert.

(ii) The tool base not only contains graphics procedures but also presentation knowledge that governs the manner in which responses are issued to a user. An expert on man-machine communication is needed for this kind of knowledge.

(iii) THE GENERALIZED PROBLEM PROCESSING SYSTEM (GPPS).

The GPPS is the interfacing mechanism between expressions of problems in the LS and expressions of knowledge in the KS (Bonczek et al., 1981a:71). It is a dynamic system consisting of software which controls the actions taken by the DSS and performs four primary functions: (a) information collection, (b) problem recognition, (c) analysis and (d) model formulation (Bonczek et al., 1984:153).

(a) Information collection.

When solving some problem, the GPPS can extract information it requires from either or both of the KS and the user.

(b) Problem recognition.

This refers to the ability of the GPPS to understand or recognize aspects of a problem not explicitly stated in the original user-supplied problem statement.

(c) Model formulation.

This function refers to the ability of the GPPS to actively build a model from model building blocks in the KS which is then used to solve the problem.

(d) Analysis.

Under analysis is understood the ability to, for example, integrate data and models with each other with the resultant generation of new knowledge (which is then incorporated into the knowledge base) or the solution to a problem (which is presented to the user via the LS).

A GPPS contains no application-specific knowledge embedded in its software and its complexity depends to a large extent on:

(a) the type of LS used - the simpler the type of LS used (for example, natural language) the more complex the GPPS becomes.

(b) the type of DSS it has to support - for retrieval-only DSS the GPPS is less complex than for computationally oriented DSS, since the latter requires the capability

of model building.

From the system builder's point of view it is foreseen that the generic approach could lead to problems of focus in reasoning. The two best-known search methods in predicate calculus, namely depth-first and breadth-first, are exhaustive search methods that do not in any way "attempt to eliminate unproductive continuations" (Bonczek, et al., 1981a:316). To constrain the actions of the system it is therefore necessary to introduce complexity into the GPPS (Bonczek et al., 1981a:317).

Since it is possible to split the knowledge required for decision support into logically-related subsets it is suggested that the mechanism that uses the knowledge (GPPS) should also be partitioned.

This will:

- (a) enable reasoning or problem-solving to proceed in a relatively direct and context-dependent manner; and
- (b) contribute towards the development of modular systems.

7.2.3 SUMMARY.

The ultra high level design of Bonczek and his co-workers was modified extensively in an effort to satisfy architectural requirements and to make it more representative of EDSS. However, the resultant framework still exhibits certain shortcomings. For example, it is not prescriptive enough in terms of subsystem interfacing. The

next section presents a modular framework followed by an evaluation thereof in terms of user requirements.

7.3 A MODULAR APPROACH.

7.3.1 INTRODUCTION.

Although it could be argued that EDSS is just a new generation of DSS and could therefore be adequately illustrated by traditional architectural representations, it differs significantly enough from the previous generation to warrant a new schematic representation. The architectural framework depicted in Figure 7.5 is one that captures the essence of EDSS as described in Chapter 5.

7.3.2 GENERAL DISCUSSION OF THE FRAMEWORK.

The framework is in accordance with the Systems Approach to software design (Churchman, 1968). This is reflected by, inter alia:

- system boundaries (outer and inner circles);
- clearly identified sub-systems (sectors); and
- the minimization of sub-system interfacing (radii).

It incorporates modern design concepts such as independence of components within an integrated environment, intelligence and generality of function (Ginzberg and Stohr, 1981:15).

Bahl and Hunt (1985:81) stress the point that designing an effective DSS is a complex problem-solving task that

requires close collaboration with its prospective users. Design attention is therefore best concentrated on decision-makers rather than on decisions. By giving the user the focal point in the framework it is illustrated that the primary function of an EDSS is *decision-maker support* and not decision-making per se.

The framework is a high-level design which is not unduly restrictive in terms of:

- (a) the level of support it can provide;
- (b) the degree to which support is generalized or particularized; and
- (c) the processes it can support, for example, individual cognitive processes, learning, communication and coordination.

(a) Level of Support: As far as the level of support is concerned the framework leaves the designer/developer with enough scope to, for example, include a full-fledged ES as an intelligent computer-assisted instruction (ICAI) component or to opt for a linkage to a CBE system such as Control Data Corporation's PLATO system.

(b) Type of Support: Designers should be aware of and be prepared to adapt EDSS design to variations in decision environments and the diverse tasks of managers. The systems that they design should therefore reflect the special characteristics of the environments in which they will

operate. The framework presented could be used directly for the design and development of a specific EDSS or it could serve as the basis for the design and development of an EDSS generator.

As a general-purpose design of EDSS the framework is meaningful in that it explicitly links the environment, system role, components, and arrangement of components (Churchman, 1968). That is, the characteristics of the system's environment and the role of the system are reflected in the components and their arrangement (architecture).

The EDSS environment is characterized by problems that are semi- or unstructured as well as by the user, who, although part of the decision-making system, is a major actor in the EDSS environment. The presence of knowledge-based components in the framework reflects the need identified by Ariav and Ginzberg (1985:10) for the system to:

- (i) bring structure to the problems encountered;
- (ii) assist the user in the process of applying judgemental reasoning to these problems; and
- (iii) assist in the generation of alternative solutions to problems.

(c) Processes Supported: By explicitly specifying the modeling, data, tools, training and scenario components the framework clearly indicates the role of the system, namely

- *supporting* a user in accessing and manipulating

- data, models, tools and scenarios; and
- *training* a user in the use of the system as well as in domain-related concepts and decision-making.

Finally, from an implementation point of view the design lends itself to a phased approach in which the core components could be implemented before the others are even designed.

7.3.3 EVALUATION IN TERMS OF USER NEEDS AND CONCEPTUAL REQUIREMENTS.

Since an EDSS is a knowledge-based system, many of the advantages to be gained by applying ES technology to the problems of DSS (see chapter 4), will be realized upon implementation of such a system. The objective of this section is to illustrate how systems based on the framework of Figure 7.5 will contribute towards meeting user requirements.

For ease of reference, some problems surrounding DSS (from chapter 3) as well as some typical EDSSs characteristics (from chapter 5) are listed here again.

PROBLEMS

EDSS CHARACTERISTICS

INHERENT TO DSS:

passiveness
limited support
inflexibility

behavioural aspects

human-like intelligence
natural performance
expertness at problem-
solving
powerful reasoning
capabilities
dynamic modelling
tutoring capabilities

ENVIRONMENTAL:

low productivity
high turnaround

Each characteristic in the right-hand column above, can be shown to make a meaningful contribution towards solving at least one of the problems surrounding DSSs. More often than not, a many-to-one relationship holds, precluding an in-depth discussion of each characteristic. Since the *training* and *scenario* components are "newcomers" to the decision support scene, they are treated more extensively than the others.

(i) TUTORING CAPABILITIES.

The need for training in the use and understanding of one of the basic components of a DSS (the model bank) has been established empirically (chapter 6). Simple usage and exposure to techniques, however, will not be enough for decision makers to reach the desired level of expertise.

One approach to solving the problem would be to transfer the modelling activity in total to the system but this will undoubtedly lead to user resistance. A better approach would be to upgrade the decision-maker's knowledge of modelling techniques through formal training. Since modelling (as well as any other quantitative technique which the user might want to master) depends on domain-related data in order to make sense to the user, and since the EDSS database is the repository of domain-related data, it is suggested that this training task be assigned to the DSS. To be more precise, what is proposed is an intelligent training component that provides tutoring based on an analysis of the context in which it was invoked, and advice that has been tailored to the user and the problem at hand.

If the decision-makers, however, feel that they have too many other disciplines under their control, and that they consequently cannot be expected to master all of them (Graham, 1980) then the knowledge-based modelling component has enough "intelligence" to explain on demand the steps it goes through in the process of solving a particular problem.

As illustrated above, training of users can take place either as a secondary task during normal system usage or as the primary objective of the system when so directed by the user. The problems of DSS addressed by each mode of usage are identified below.

(a) Use-mode.

During normal system usage, the main activity of an EDSS is problem-solving (performing decision-related tasks) but at any point in time the user can gain insight into the system's reasoning process by asking, for example, "why" a certain input is required by the system or "how" the system reached a specific conclusion. The system responds to these questions by providing (as an absolute minimum) a trace of the rules used in the reasoning process.

The following DSS problem areas are addressed by this simple, yet useful, facility:

* **Passiveness.**

The system becomes inter-active in the true sense of the word. That is, the output produced by the system assists or actively supports the user in determining an appropriate response with which to direct the further functioning of the system.

* **User resistance.**

By providing answers to the user queries "why" and "how", the system becomes transparent to the user, thereby meeting the user's need to understand the underlying logic of the system. Since the user can follow the lines of reasoning employed by the system, s/he becomes at ease with the system. It is no longer a black box producing unexplainable solutions that must be accepted without questioning. User resistance to

using the system is thereby reduced. For the more experienced user the answers will be an indication of the correctness of system functioning which, in turn, will lead to improved system credibility and user acceptance of proposed solutions.

* **Productivity.**

For the inexperienced decision-maker or newcomer to the domain in question, this explanation facility provides the opportunity to grow in decision-making and/or domain expertise. This raising of the user's knowledge level could lead to improved performance and thus to higher productivity.

(b) **Tutoring mode.**

When tutor-mode is selected by the user, control is transferred to the expert training component. It is suggested that this intelligent computer-assisted instruction (ICAI) subsystem should:

- have multiple user-interfaces;
- teach basic domain principles, decision-making concepts, and quantitative methods techniques;
- support simulated runs of the EDSS;
- use graphics to illustrate structure, dependencies, etc;
- adapt its tutorial strategy to the user's knowledge level; and
- maintain a record of students' progress.

The objective of this component is to provide some theoretical information (basic domain, decision-making or modelling concepts) but always within the context of accomplishing a specific goal (simulated problem-solving runs). During a training session the trainee will, for example, be able to apply various decision-making theories (rational, satisficing, etc.) to a problem in order to observe and evaluate the effect of each. The inclusion of such a component will adequately meet both the user's need for training and the conceptual requirement that users need support to explore alternative ways of solving problems. This feature of EDSSs will assist in solving the following problems associated with traditional DSSs:

* User resistance.

As with the simple explanation facility described earlier, contributions will be made towards overcoming user resistance and enhancing system credibility.

* Productivity.

Improving the productivity of knowledge workers (decision makers) is one of the main objectives of this EDSS component. It achieves this objective by eliminating the need for formal training courses and by encouraging users to experiment with the system without fear of their actions causing system malfunction (simulated runs).

Experienced employees previously responsible for the presentation of formal training sessions are now freed from this responsibility and can concentrate on what they are best at: productive work. Trainees receive individualized attention based on the teaching methods of an expert in the educational field and in a fashion that allows them to assimilate new knowledge at a faster rate than is possible with the traditional seminar or lecture approach.

The provision of multiple user-interfaces allows different kinds of users (for example mathematically versus linguistically oriented) to learn the system with the same ease. Although it is debatable whether graphics contribute towards improving organizational decision-making, the value thereof has been proven in education and training. One area in which the graphics component of the EDSS TUTOR enhances productivity is in giving the user/trainee a clear picture of the structure underlying the knowledge in the knowledge-base which leads to, for example, more optimally formulated queries.

(ii) HUMAN-LIKE INTELLIGENCE.

Being knowledge-based, EDSSs will have the ability to assist decision makers in developing better ways of solving problems. For example, by following a process known as knowledge refinement through abstraction, EDSSs could contribute towards increasing the body of domain knowledge. This potentially powerful capability of systems based on the

proposed framework will transform support systems into truly active participants in the decision making process.

Another reason why all the subsystems in the framework are knowledge-based is to meet the user requirement that the system must adapt to the user and not vice versa. Relevant topics in this respect are, inter alia, the ability of knowledge-based systems to perform contextual reasoning and to switch dialog styles based on a model of the user.

The inclusion of a scenario component in the framework represents an effort to better augment the limited cognitive capabilities of the user. In order to meet the users' need for support during all the phases of decision making, EDSSs are provided with an active memory (scenario base) from which previous solutions are pulled (intelligence phase) for adaptation (design phase) by a human-like analogical reminding mechanism (expert scenario management system) to be used in the current situation.

(iii) NATURAL PERFORMANCE.

The natural language interface of an EDSS will help solve the problem of passiveness by:

- (i) increasing the amount of communication between system and user (for example, explanations); and better formulated queries).

The latter is achieved by allowing the user to phrase queries in a way which comes natural to him/her and then to

combine information from the generated user model with, inter alia, focus management, structural and optimization rules in the appropriate knowledge-base(s).

It will also address the behavioural problem of user resistance based on the cryptic nature of some traditional interfaces while simultaneously meeting the conceptual requirement that computer-based systems should be easy to use. User productivity will increase since exploration of the data base, which leads to increased knowledge of the domain and therefore improved decision-making, can commence without users having to learn a new language. Implied in this last statement is that systems based on the proposed framework will facilitate user learning by providing support for exploration in the form of natural language communication between system and user.

(iv) EXPERTNESS AT PROBLEM-SOLVING.

The extensive range of knowledge types reflected in the framework will allow EDSSs to reason intelligently about domain problems, modelling techniques, data management, and user training. The resultant discussion with the user of, for example, a number of different approaches to solving a particular problem or the applicability of a specific statistical technique to a data set, meets the users' need to understand the system's logic as well as the need for training.

Expertness also implies that there should be constant growth in knowledge. A stagnant support system in the organizational decision-making environment will become out-of-date within a very short period of time. The use of heuristics (to refine the knowledge-base) and of fifth generation programming languages make EDSSs not only flexible but also adaptable to changes in the environment. An important user requirement that will be satisfied by systems possessing expert problem-solving capabilities is therefore that of evolving system capabilities. The synergistic result of the interaction between "intelligent" agencies such as the user, the expert tool manager, and the expert scenario manager might well be systems that appear as if they are evolving automatically over time.

(v) POWERFUL REASONING.

Under the guidance of metaknowledge the system will be able to transform user queries and processing requests into standard forms which make the execution of fast special-purpose algorithms possible (Jarke and Vassiliou, 1983:79). This capability of EDSSs to select the most appropriate strategy for the prevailing situation transforms the system into an *efficient* and *active* partner in the search for answers.

The use of different inferencing strategies within a single EDSS counteracts system inflexibility while simultaneously providing support for the different cognitive styles of

users.

The scenario component, however, is the best example of the powerful reasoning capabilities of EDSSs. It addresses users' needs in terms of memory aids, training, exploration, flexible modelling and adaptive support over time. In addition it provides the user with the necessary technical support to systematically structure and analyze (re-analyze) a problem and to critically examine and evaluate alternative solutions.

As event logger the scenario system will, if so directed, capture both user and DSS actions and responses for the duration of a scenario (a logical session between the user and the system). Sufficient information is recorded to enable the system to "replay" a scenario at any future point in time. The user can "prime" a new scenario with a previously recorded scenario and can then request the system to step through the actions from any point within the scenario. The user has the option of accepting his/her previously entered responses or entering new responses. At any point during a scenario the user can submit comments for documentation purposes.

Part of the intelligence built into the scenario component will enable it to "learn" how to solve different classes of problems. The user trains the system by marking selected scenarios as instances of solving a specific type of

problem. Through a process known as "inductive learning" the system then discovers patterns in the (possibly) chaotic collections of scenarios. This process of conceptual learning from examples involves operations of generalizing, specializing, transforming, correcting and refining knowledge representations (Michalski,1983:87)

When the user faces a problem for which s/he has no "canned" solution, the scenario manager (ESMS) can be requested to compare the key elements of the current problem with those of the prototypes generated as a result of the action described in the previous paragraph. The system then locates and presents the solution scenario of the problem that fits the current problem the closest. The user/system team can then apply judgemental reasoning to the proposed solution in order to modify the scenario to fit the current problem exactly.

In a discussion of his adaptive framework for DSS, Keen (1980:13) states that

"an effective DSS encourages the user to explore new alternatives and approaches to the task (System ---) User). This in itself stimulates new uses of the system, often unanticipated and idiosyncratic (User ---) System)."

An expert scenario management component as described in this section has the potential for new and innovative uses such as playing the role of "devil's advocate". In this reversal

of roles it will force the decision-maker to consider all possible consequences of a decision by asking provocative "WHAT-IF" questions in response to a user-suggested solution to a problem. In the process a plan of action evolves based on all relevant past experiences and not just on those that the user considers relevant. This type of use is made possible by the scenario component's analogical reasoning powers and store of solution scenarios in combination with the presence of historical data in the database.

What has been described above should meet the need of strategic management to perceive future external and internal trends before they occur and to determine their impact on new products (services, etc.) so as to improve overall organizational performance [(Thierauf, 1982:63); (Stephenson and Stephenson, 1983:396)]

(vi) DYNAMIC MODELLING.

In response to the need for flexible, interactive modelling the framework supplies the user with three important aids:

- (a) an expert scenario management system with which to develop and maintain models together with their subsequent modified versions in an easy-to-use and flexible way;
- (b) an expert tool manager to tailor the output of analysis to fit the expected audience; and
- (c) a knowledge-based modelling subsystem that can perform

a range of tasks from executing a specified model with user-determined data to automatically building a model and executing it against, what it considers to be, the most appropriate data.

This ability to build models dynamically from basic building blocks using heuristic knowledge, also contributes towards more *active system participation* in the decision-making process whilst enhancing user productivity.

As was pointed out in section 7.2.3 the framework presented in this section contains more explicit design information than the generic framework. In addition it is more appealing in the sense that it focusses attention on the user while simultaneously emphasizing the system as a whole. It is therefore proposed that the framework of Figure 7.5 be adopted as representative of EDSS.

7.4 CONCLUSION.

The frameworks discussed in this chapter represent an attempt to consolidate the results of the fragmented research efforts in the DSS field into one architecture for a new generation of expert DSSs. They reflect the basic assumption made that users are in need of support on more fronts than was originally perceived. The framework therefore includes new components, such as the intelligent tutor, to cater for these needs.

Since the ideas put forward in this chapter are based mainly on deductive reasoning, complemented by some empirical results, they need to be tested in the market place. In order, therefore, to verify that the proposed modular framework for EDSS is complete and to determine its acceptability a Delphi study was performed. The results are described in Chapter 8.

CHAPTER 8.

EMPIRICAL EVALUATION STUDY.

8.1 INTRODUCTION.

The methodology proposed in this study was developed through deductive reasoning. Its acceptability stills needs to be established. Therefore, a testing against the facts of reality (that is, induction) still needs to be performed (Sowa, 1984:332).

Furthermore, as was pointed out in Chapter 1, section 1.1, the DSS field is highly interdisciplinary in nature. It was therefore decided to test the acceptability of the ideas put forward in this thesis by submitting them to a panel of academic experts and DSS/ES practitioners. A Delphi study was chosen as a vehicle since it assists a diverse group of individuals in approaching consensus on a set of issues.

8.2 OBJECTIVES.

The primary objective of this empirical part of the study, as put to the participants, was to determine whether consensus could be reached on:

- (a) the necessity and sufficiency of the components proposed in Chapter 6 of this thesis; and
- (b) the acceptability of the modular framework presented

in Chapter 7.

8.3 METHODOLOGY.

Delphi is a group process which utilizes written media to solicit and aggregate the judgements of a number of individuals (Miller, 1984). Essentially, Delphi is a series of linked questionnaires. Starting with an open-ended questionnaire, succeeding questionnaires feed back group responses to the previous questionnaire and ask for further information. The process stops when consensus among individuals has been reached or when sufficient information has been exchanged (Delbecq, Van de Ven and Gustafson, 1975).

A number of possible participants were contacted and from them a panel of ten was chosen. The rationale behind the actual composition of the panel is provided in the next section.

Data was collected in two rounds of surveys as follows:

ROUND ONE. Under a cover letter explaining the underlying concepts of EDSS participants were supplied with a two-part questionnaire. An example of this questionnaire has been included as Appendix 1. *Part one* consisted of the list of seven proposed EDSS components together with a short description of the objectives and functions of each as seen by the author. Participants were asked to rate the components on a 10-point scale ranging from "totally unnecessary"

(1) to "absolutely essential" (10). They were also encouraged to augment their ratings by means of comments and to suggest new components for consideration.

Part two consisted of four frameworks which participants were also requested to rate on a 10-point scale. A rating of one corresponds to "totally unacceptable" and a ten to "a perfect framework". Participants were asked to identify both strong and weak points in each framework and to submit alternative frameworks if they so wished. The set of frameworks consisted of:

- (a) Framework A: the modular framework proposed in Chapter 7;
- (b) Framework B: a modular framework using the more traditional rectangular presentation format and showing more detail;
- (c) Framework C: the generic framework discussed in Chapter 7; and
- (d) Framework D: an ultra-high level design.

The rating method was chosen because it would produce numerical scores for each component from which average ratings could be calculated. If any of the newly proposed components received an average rating that did not differ significantly from that of the lowest ranking traditional component, then hypothesis 3 could not be rejected. This implies that it is possible for all the new components to rank below the traditional components without causing the

rejection of the hypothesis. Had the ranking method been chosen, then hypothesis 3 would have had to be rejected if no new component ranked higher than the lowest ranking traditional component.

ROUND TWO. Participants responding to the first round were sent feedback showing the results of the first round. This included relevant statistics, comments and suggested components and frameworks. Respondents were asked to study the feedback and then to review their round one ratings in the light thereof.

8.4 THE PARTICIPANTS.

It was decided to keep the panel size small in order to offer respondents the opportunity of discussing problem areas with the study leader. Consequently, an invitation was extended to the selected ten participants to contact the author should they need clarification on any point.

The first criterion used in selecting members of the panel was that of profession with academics and practitioners represented equally. Within the *academic* group it was decided to select one participant from each of the following disciplines:

- (a) Computer Science (specialization field: systems design);
- (b) Business Data Processing;

(c) Artificial Intelligence; and

(d) Information Systems.

The group representing *industry* included four consultants actively involved in the development and marketing of expert systems and two managers in a large financial institution. One of the managers is responsible for the development of personal information systems, the other being a DSS user who is currently doing research for his Masters degree in ES-enhanced DSS.

This specific composition was chosen to reflect the interdisciplinary nature of DSS and the interdependency of theoreticians and practitioners. Unfortunately, three of the selected consultants failed to respond, leaving a panel of four academics and three practitioners.

8.5 FINDINGS.

This Delphi is seen as a pilot study. Because of its exploratory nature no inferences are drawn from the results. Furthermore, the small sample size rules out the possibility of strict statistical analyses.

ROUND ONE: The results of round one (presented in Figure 8.1) confirmed the expected in some ways and revealed surprises in others. The low ranking of the generic framework (Framework C in Figure 8.1) was as expected. On the other

COMPONENTS

RANK	COMPONENT	MEAN RATING	STANDARD DEVIATION
1	Data subsystem	8.9	1.2
2	User interface	8.4	1.9
3	Scenario comp.	7.1	2.2
4	Modelling comp.	7.0	3.0
5	Tool subsystem	6.7	2.6
6	Comm. & Contr.	6.7	2.9
7	Training comp.	5.6	3.3

FRAMEWORKS

RANK	IDENTIFICATION	MEAN RATING	STANDARD DEVIATION
1	Framework B	6.28	2.69
2	Framework D	6.0	1.8
3	Framework A	5.42	3.1
4	Framework C	5.14	2.73

Figure 8.1

DELPHI STUDY: ROUND ONE RESULTS.



hand, the fact that the two frameworks showing the most and least detail were rated number one and two respectively, was somewhat unexpected.

While modelling has traditionally been regarded as one of *the* essential capabilities of a DSS, the modelling component only managed a fourth place after the data, user interface and scenario components. The low rating of the training component also came as a surprise. The high standard deviation in both cases, however, indicated widely differing opinions in this respect.

The following additional components were suggested by participants for possible inclusion in an EDSS:

- (a) a human resources component;
- (b) a project management component;
- (c) a configuration subsystem;
- (d) a machine learning subsystem; and
- (e) an information base.

To accommodate these newly-proposed components, two new frameworks were also submitted (Figures 8.2 and 8.3).

ROUND TWO: A decrease in the standard deviation of the proposed modular framework (Framework A in Figure 8.4), was accompanied by an increase in its average rating. It now ranked second only to the more detailed modular framework B. Framework E (one of the newly-suggested frameworks) seems to be a strong contender for the position held by the generic

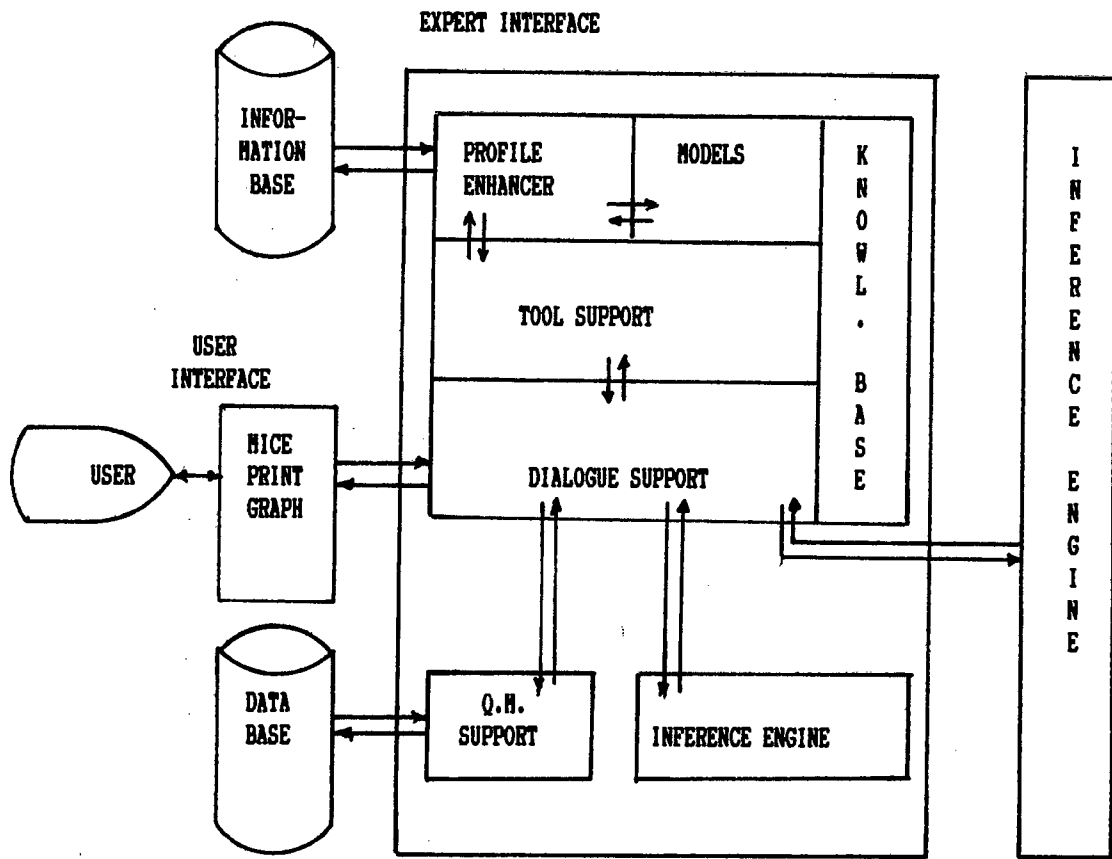


Figure 8.2

Framework E: Proposed by Jay Carson.

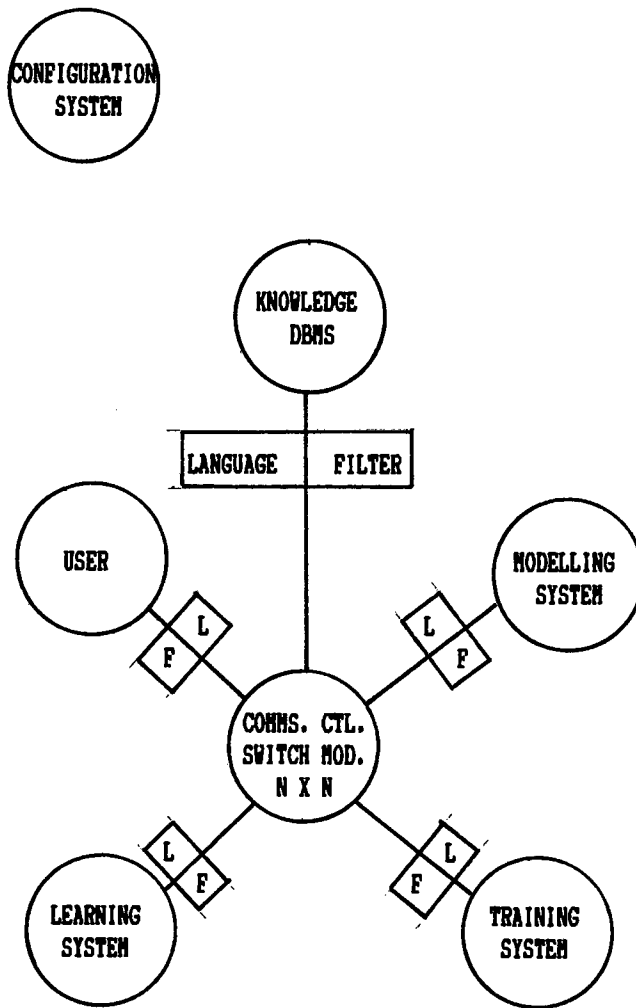


Figure 8.3

Framework F: Proposed by H. Messerschmidt.

COMPONENTS

RANK	COMPONENT	MEAN RATING	STANDARD DEVIATION
1	Data subsystem	9.3	0.8
2	User interface	8.7	1.2
3	Tool subsystem	7.0	1.8
4	Scenario comp.	7.0	1.9
5	Modelling comp.	6.0	2.2
6	Training comp.	5.8	1.9
7	Comm. & control	5.8	2.1
8	Config. subsyst.	5.0	3.6
9	Project mgmt.	3.0	3.8
10	Human resources	1.5	2.3
11	Information base	1.5	3.0

FRAMEWORKS

RANK	IDENTIFICATION	MEAN RATING	STANDARD DEVIATION
1	Framework B	6.8	1.9
2	Framework A	6.2	2.0
3	Framework D	5.8	1.9
4	Framework C	5.3	1.3
5	Framework E	5.2	2.9
6	Framework F	4.8	2.2

Figure 8.4

DELPHI STUDY: ROUND TWO RESULTS.

framework.

Against all expectations the *modelling* component dropped to the fifth position with both the *tool* subsystem and the *scenario* component outranking it. With all participants assigning a value of zero to at least one of the newly-suggested components (thereby indicating that it should be part of one of the originally-proposed components), the low rankings for these components were to be expected.

8.6 DISCUSSION.

The following remarks are based on the combined results of the two rounds presented in Figure 8.5.

- (a) With the exception of framework D there was a general decrease in standard deviation. This was interpreted as a definite move towards consensus.
- (b) Although a third round was planned, the results after two rounds indicated that sufficient information had been exchanged. Since it was considered most likely that a next round would not have produced significantly different results, the process was stopped.
- (c) With all the originally proposed components achieving above average ratings (larger than five on a scale of ten), it would seem as if they will be *necessary* com-

COMPONENTS

FINAL RANK	COMPONENT	- ROUND ONE -		- ROUND TWO -	
		MEAN RATING	STANDARD DEVIATION	MEAN RATING	STANDARD DEVIATION
1	Data subsystem	8.9	1.2	9.3	0.8
2	User interface	8.4	1.9	8.7	1.2
3	Tool subsystem	6.7	2.6	7.0	1.8
4	Scenario comp.	7.1	2.2	7.0	1.9
5	Modelling comp.	7.0	3.0	6.0	2.2
6	Training comp.	5.6	3.3	5.8	1.9
7	Comm. & control	6.7	2.9	5.8	2.1
8	Config. subsyst.	- new comp.-		5.0	3.6
9	Project mgmt.	- new comp.-		3.0	3.8
10	Human resources	- new comp.-		1.5	2.3
11	Information base	- new comp.-		1.5	3.0

FRAMEWORKS

FINAL RANK	IDENTIFICATION	- ROUND ONE -		- ROUND TWO -	
		MEAN RATING	STANDARD DEVIATION	MEAN RATING	STANDARD DEVIATION
1	Framework B	6.3	2.7	6.8	1.9
2	Framework A	5.4	3.1	6.2	2.0
3	Framework D	6.0	1.8	5.8	1.9
4	Framework C	5.1	2.7	5.3	1.3
5	Framework E	- new -		5.2	2.9
6	Framework F	- new -		4.8	2.2

Figure 8.5

RANKING DATA BY DELPHI ROUND.

ponents in future EDSS. Similarly, with the possible addition of a *configuration subsystem*, the set of proposed components could probably be regarded as *sufficient* for effective functioning of an EDSS.

(d) The main difference between frameworks A and B, apart from representation format, is that of level of detail. With framework B receiving the highest rating, it would seem as if, in the case of framework A:

(i) either the particular representation format; or

(ii) the lack of detail; or

(iii) both

contributed towards its low rating. The low overall performance of all the frameworks suggests that this is an area for further research.

8.7 CONCLUSION.

The Delphi study described in this chapter represents a first attempt at verifying the deductive reasoning that has gone into the development of a methodology for a new generation of enhanced DSS. It was only an exploratory survey in that it solicited the opinions of a handful of academics and practitioners.

The results of the study indicated that there is reason to believe that a full-scale investigation of decision-makers'

needs will reveal that the average user has come to expect a *wider range of support* from computerized systems. For DSS, being support-oriented, this could have far-reaching effects.

The following chapter is devoted to an evaluation of research efforts in terms of objectives and hypothesis. Final conclusions are drawn and recommendations for future research are made.

CONCLUSIONS AND RECOMMENDATIONS.

9.1 RESTATEMENT OF OBJECTIVES.

The twofold objective of this research was to establish the adequacy of traditional DSS to adapt to changing user needs and to examine alternative conceptual frameworks for Expert Decision Support Systems.

Three subgoals were set in order to determine progress:

- (1) Establish whether DSS is meeting user requirements or whether it exhibits limitations and shortcomings.
- (2) Survey the ES field for possible components for inclusion in EDSS and for tools and techniques that could assist in the construction of knowledge-based components.
- (3) Develop an architectural framework for a new generation of knowledge-based DSS that incorporates elements of ES.

The following hypotheses were then proposed for testing:

Hypothesis 1.

DSSs which are developed using traditional concepts and development methodologies are not meeting user requirements due to shortcomings in their design and limitations imposed on them by technology and developers.

Hypothesis 2.

The field of ES will prove to be a rich source of tools and techniques to assist in the construction of knowledge-based components for EDSS.

Before restating the last hypothesis, a reader is reminded that the approach whereby *only* the traditional DSS components are enhanced with ES features, was termed the *enhancement route*. A knowledge-based approach to DSS design and development, on the other hand, was labelled the *EDSS route*.

Hypothesis 3.

The EDSS route will be more complete in terms of meeting decision-maker needs than the enhancement route.

9.2 EVALUATION IN TERMS OF HYPOTHESES/OBJECTIVES.

(a) SUBGOAL 1.

The survey of the DSS literature (Chapter 3) has indicated that DSSs developed along traditional lines:

- (a) could possibly be regarded as being passive in their support of decision-makers;
- (b) provide only limited support;
- (c) are inflexible and therefore insensitive to changes in their environment; and
- (d) that they are not always easy to use.

Some empirical evidence to this effect does exist, for example, Uliana (1986), but ideally, it needs to be comple-

mented by a survey of the DSS user community in order to determine the exact scope and magnitude of DSS shortcomings.

Enough evidence, however, was found to warrant the acceptance of hypothesis 1 thereby implying the achievement of subgoal 1.

(b) SUBGOAL 2.

With the enhancement of DSS by means of ES tools and techniques already well under way [(Dolk and Konsynski, 1984); (Wiederhold, 1984); (Swartout, 1985); (Blanning, 1985); (Bonczek, Holsapple and Whinston, 1981a)], it made sense to try and achieve subgoal 2 by borrowing from AI and its related fields of study, namely ES and Natural Language Processing (NLP).

Since general-purpose knowledge-based building blocks (components) are only starting to appear (Gevarter, 1987:40), the search concentrated on ES tools and techniques. A large number of these were consequently found to have direct implementation possibilities in DSS. The following are just a few examples taken from Chapter 4:

- (a) domain-specific knowledge base;
- (b) inference engine;
- (c) logical deduction and judgemental reasoning;
- (e) knowledge representation schemes such as production rules and predicate calculus;
- (f) forward- and backward-chaining as well as a mixed

- control strategy; and
- (g) an explanation facility.

Since:

- (a) in Chapter 6, users were found to be in need of:
 - (i) training in domain and task related matters;
 - (ii) flexibility in manipulation of system capabilities;
 - (iii) exploration of alternative problem-solving methodsand
- (b) these needs had to be accommodated in a system that would be adaptive to future changes.

it was decided to add an ICAI (Intelligent Computer-Assisted Instruction) component, a knowledge-based tools component and an expert scenario component to the three traditional DSS components. As was pointed out earlier, the latter are already in varying stages of being enhanced with ES technology; a practice that, with certain reservations, should be encouraged. This addition of new components represents a major departure from the traditional approach to decision support *in that a widening of system boundaries is suggested.*

With ES and NLP technology well represented in the components mentioned above, hypothesis 2 could be accepted. The paper model of EDSS that is presented in Chapter 5, is proof that subgoal 2 was indeed achieved.

(c) SUBGOAL 3.

Although it was found that considerable progress has been made in updating DSS technologically, little evidence was found of efforts to integrate the individual attempts into a unified whole. It is the contention of the author that if this state of affairs is to continue, the user will be deprived of many, if not all, of the benefits that could be derived from the synergistic interaction between intelligent system components.

Since the author subscribes to the Systems Approach and *systems thought* emphasizes the need to take a holistic view in order to explain why a system is structured as it is, or how it should be structured (Ariav, et al., 1985:1046), careful attention was paid throughout the research to, inter alia:

- (a) the environment in which a DSS operates (Chapter 2,3);
- (b) the role or objective of the system (Chapter 5);
- (c) the components of the system (Chapter 6); and
- (d) the arrangement of components (Chapter 7).

By the time that the identified components had to be arranged in some architectural framework, the author was convinced of the following two facts: *Firstly*, a new generation of knowledge-processing or expert decision support systems (EDSS) has become necessary and *secondly*, the needs of decision-makers would be better served by a system with a different structure than traditional DSS.

Two frameworks were consequently developed; one based on the generic DSS framework of Bonczek, Holsapple and Whinston (1981), the other based on the Systems Approach. A Delphi study (reported on in Chapter 8) was then conducted to test the sufficiency and completeness of the set of components as well as the representational acceptability of the two frameworks.

The results of the study indicated, amongst other things, that there are more acceptable architectural representations than those developed in Chapter 7. However, with an average rating of 6,2 out of a possible 10 and a ranking of number two amongst five other frameworks, the proposed modular framework could not be labelled as "unacceptable".

Furthermore, Hypothesis 3 was supported by the fact that two of the "new" components received ratings that ranked them above the traditional DSS modelling component. One of these components is a *knowledge-based tool handling system*, the other an *expert scenario management system*. It is envisaged that the latter will create an audit trail of model building activities and actual decision-making sessions in addition to utilizing analogical reasoning techniques to provide expert support in the generation of alternative solutions and new approaches to problem-solving.

With the *acceptance* of hypothesis 3 and the availability of a reasonably acceptable architectural framework for EDSS,

subgoal 3 could be considered as reached. It is therefore concluded that traditional DSS is probably not well enough equipped to continue meeting users' changing needs. Furthermore, a new generation of knowledge-based DSS (EDSS) that includes components not found in the previous generation, could be designed using the guidelines presented in this thesis. It is hypothesized that such systems will be more adaptive to changes in their environment mainly because (i) they will be knowledge-based and (ii) they will include a larger number of support-providing components.

9.3 REMAINING PROBLEMS.

One of the main obstacles on the path of EDSS development is the scarcity of adequately qualified and experienced Intelligent Systems Engineers (ISE), a new kind of professional identified (or labelled) by Hayes-Roth (1987:87). The design and development of an EDSS require a fair amount of expertise in knowledge representation, inferencing, knowledge engineering, and deductive learning techniques in addition to the traditionally required information systems design and development skills. It is foreseen that it will be some time before training in knowledge processing technology will permeate a significant number of tertiary educational institutions' pre-graduate syllabii. A possible intermediate solution is to involve post-graduate students at an earlier stage in knowledge systems research and development projects.

Another problem is the lack of interchangeable, re-usable and standardized knowledge system construction products. The state-of-the-art in expert or knowledge system building tools is such that resultant systems are either highly customized because of the use of custom-built tools or they are narrow, specialized and esoteric when commercially available tools are used. However, according to Gevarter (1987:40) more versatile and modular tools that will allow developers to choose various knowledge representations and inference techniques are "in the works". Tools with easy-to-use interfaces to other software systems such as data bases and spreadsheets are already becoming available while increasing use is made of graphics to enhance the end-user and developer interfaces.

9.4 RESEARCH CONTRIBUTIONS.

The research described in this thesis could be classified as "applied research", that is, the deliberate investigation of a problem of practical importance (Beveridge, 1957:126). Research of this nature could therefore contribute towards both theory and practice.

With respect to the academic world, it is expected that this research will focus attention, and generate discussions on, the essence of DSS. By deviating from the traditional data-model-dialog view of DSS the author expects severe criticism

from the more conservative section of the DSS research community. However, it is also expected that some of the ideas expressed in this thesis will result in further research efforts, possibly and also hopefully, developmental research.

The re-iteration of the fact that researchers should adopt a systemic view of DSS in order to design effective DSSs (Ariav and Ginzberg, 1985), is seen as the main contribution towards the discipline. It is expected that attempts will be made to integrate the diversity of fragmented research efforts in order to create a solid basis for the design of future generation DSSs.

For those involved in the more practical side of DSS, for example system builders, the research is expected to provide clarity on the structural composition of knowledge-based DSS through identification of system components and their interfaces. The user, however, is expected to benefit most from this research. It is expected that EDSSs developed in accordance with the guidelines in the latter part of this thesis will be better equipped than their DSS counterparts to meet the unique requirements of managerial decision-makers.

9.5 FUTURE RESEARCH DIRECTIVES.

Based on the results of the Delphi study it is recommended that a full-scale survey of DSS users be undertaken to empirically prove/disprove the need for training, tools and scenario components. It is also possible that additional user needs will be uncovered by such a survey.

The one element of the Systems Approach that has not been covered in this thesis is consideration of the resources needed to build and operate a system. The key questions that must be answered are:

- (a) How can the proposed EDSS best be realized?
- (b) How close to that ideal can a feasible system come?
- (c) Which resources should be employed to build and support the EDSS?

(Ariav and Ginzberg, 1985:1049).

Hardware Resources.

The architecture of the majority of currently available commercial computers is such that they could be termed "numerical processors". These machines are relatively slow when it comes to performing deductive processing on the symbolic representations of the information in knowledge-based systems. Until machines appear on the market that employ, for example, the so-called data flow architecture [(Simons, 1984:232); (Gurd, Kirkham and Watson, 1985)] or the Cosmic Cube concept (Seitz, 1985), alternative ways must be

sought to improve system efficiency.

An area that appears to be particularly promising is that of parallel computing. The proposed modular framework lends itself ideally to implementation on multiple processors which would, for example, make it possible for a user to be trained by the training subsystem while waiting for the modelling subsystem to generate and execute some complicated model.

Software Resources.

The key research requirement here is seen as the design and development of specialized knowledge representation languages with which to capture the diverse kinds of knowledge required for successful EDSS operation.

In addition, the fragmented approach to DSS enhancement (frequently referred to in this thesis) could lead to the development of single-function building blocks with which EDSSs could be constructed. Examples are knowledge-based model and data management systems. A key tool that will become indispensable is an arrangement package or software environment which addresses the interfaces among the other system building blocks, but not their contents (Ariav, et al., 1985). If the different components are implemented in a multi-tasking or parallel computing environment this simple communications tool will need modification to include elements of control.

8.5 CONCLUSION.

The fact that organizations are facing an information glut and have a dire need to distribute the know-how of their top performers to all employees (Hayes-Roth, 1987:86) combined with the ever-increasing competitiveness of the business world will, in the author's opinion, lead to the commissioning of research and development of exactly the type of system proposed in this thesis.

APPENDIX 1

QUESTIONNAIRE ON EXPERT DECISION SUPPORT SYSTEMS

PART A.

COMPONENTS OF EXPERT DECISION SUPPORT SYSTEMS.

The components listed below are suggested for inclusion in an Expert Decision Support System. Indicate the importance of each component for the effective functioning of a knowledge-based decision support system by entering numbers between 1 and 10 (inclusive) in the squares provided. Note: A 10 implies that the component is regarded as absolutely essential.

Substantiate your ratings by providing reasons for or against the inclusion of each component. Short and to-the-point notes will be appreciated.

Any suggestions as to other components that should be considered are welcome. List such components in the space provided, rate them and provide reasons for both their inclusion and the rating they are given.

1. A knowledge-based database management system.

It is proposed that a wide variety of knowledge should be included in the DBMS of an EDSS with the objective to directly produce suitable information for decision making and planning. Enterprise directing, focus management, application-specific, general procedural and structural knowledge are a few examples.

REMARKS:

2. A knowledge-based model management system.

The main objective of this component is to provide the user with a flexible and easy-to-use interface to quantitative methods/management science techniques. It is hypothesized that the inclusion of inferencing, heuristics and models of inexact reasoning will extend support to the qualitative level.

REMARKS:

3. A knowledge-based training system.

An intelligent computer-assisted instructional component could teach users the basic domain concepts, how to make more effective decisions, how to build models as well as how and when to use the various powerful decision-making tools at their disposal.

REMARKS:

4. A knowledge-based scenario management system.

One function of this component is to act as a run-time historian. It creates an audit trail of model building, model modification and actual decision-making sessions enabling the user to retrace his/her steps. In addition, it utilizes analogical reasoning techniques to provide expert support in the generation of alternative solutions and new approaches to problem-solving.

REMARKS:

5. A knowledge-based tool handling system.

It is proposed that the decision maker be provided with a comprehensive toolkit that contains powerful, easy-to-use and flexible tools with which to tailor the system to better fit his/her unique way of solving problems and to add new operators, functions and knowledge to the system. Tool management (for example, tool selection) should be knowledge-based.

REMARKS:

6. A knowledge-based user-interface system.

An intelligent front end will hide the technical details of the system from the user, contribute towards problem recognition and problem structuring and tailor its responses to fit the user's profile. It will enhance communication by supporting multiple dialog styles and allowing mixed initiative.

REMARKS:

7. A knowledge-based communication and control system.

The purpose of this component is to keep the number of inter-module connections to a minimum, synchronize communication and simplify the task of adding new components. An added benefit is the possibility to implement the different components as separate systems executing in parallel on different processors.

REMARKS:

Additional components suggested:

(State the component and give it a rating. A short description and motivation for its inclusion will be welcomed.)

PART B.

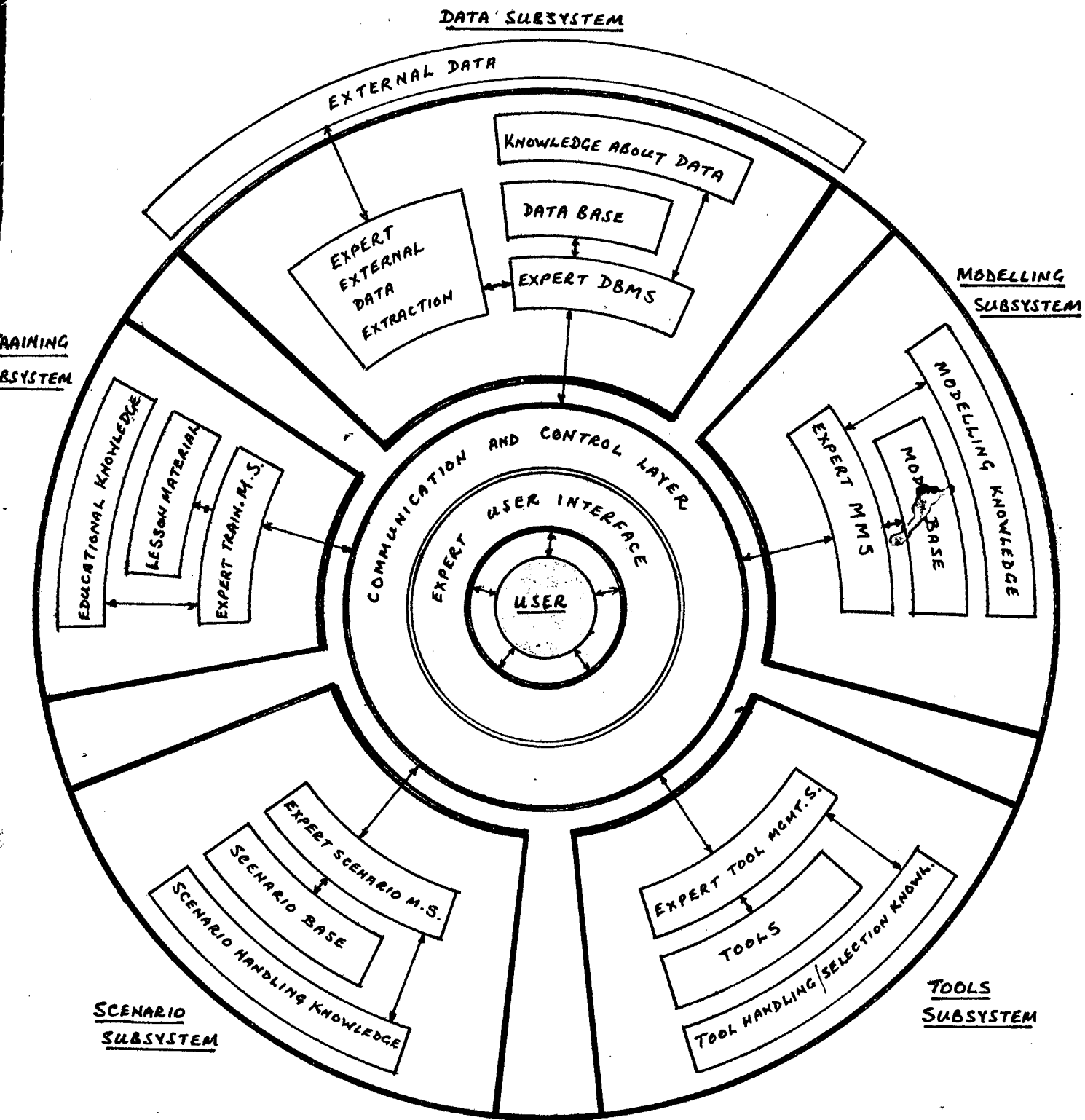
ARCHITECTURAL FRAMEWORKS.

A number of possible architectural frameworks are presented on the following few pages. The objective is to determine the "best" framework for EDSS. You are requested to order the given frameworks from most to least acceptable by assigning a value on the scale of 1 to 10 to each framework. Use the squares provided for your ratings. Note: A rating of 1 implies a totally unacceptable framework.

To facilitate ease of processing you are requested to rate the frameworks in such a way that no two have the same rating.

Please write short commentary notes in the space provided on both the positive and the negative aspects of each framework. These remarks will be used to improve the designs before they are submitted to all participants during a subsequent round of evaluation.

FRAMEWORK A



FRAMEWORK A.

RATING

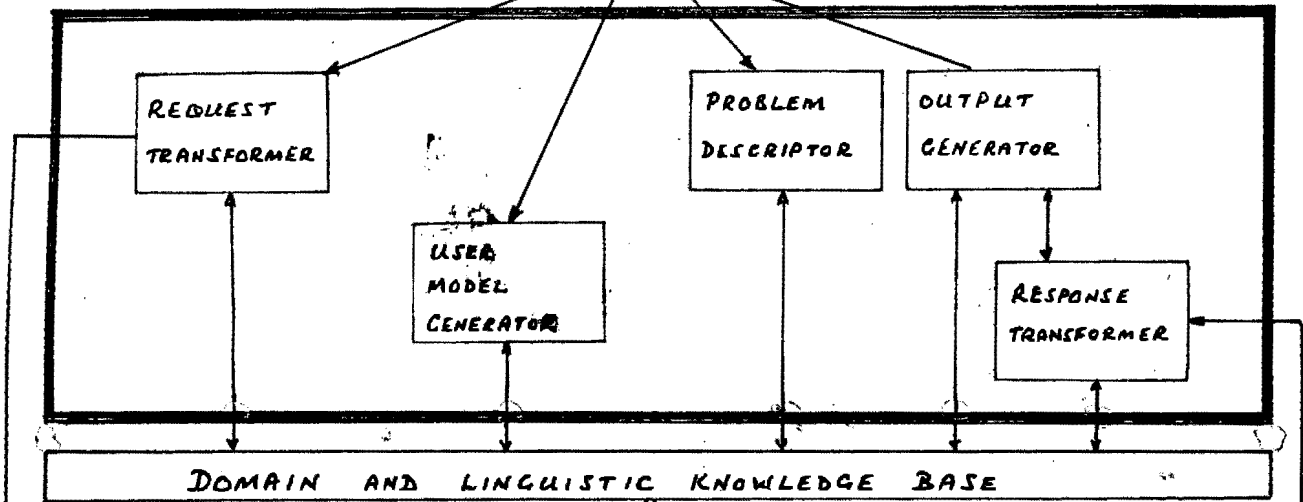
STRONG POINTS.

WEAK POINTS.

FRAMEWORK B

USER

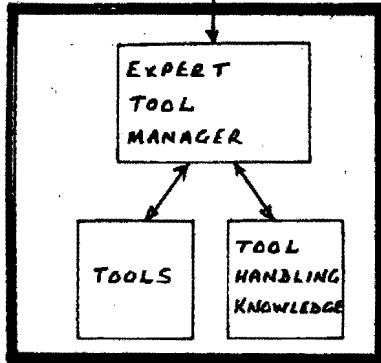
EXPERT USER INTERFACE



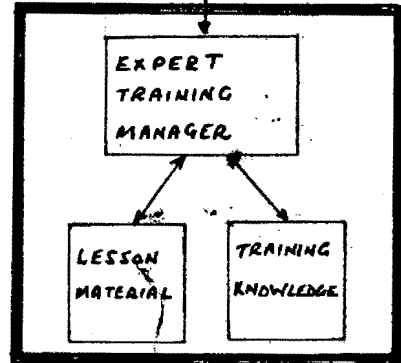
DOMAIN AND LINGUISTIC KNOWLEDGE BASE

COMMUNICATION AND CONTROL LAYER

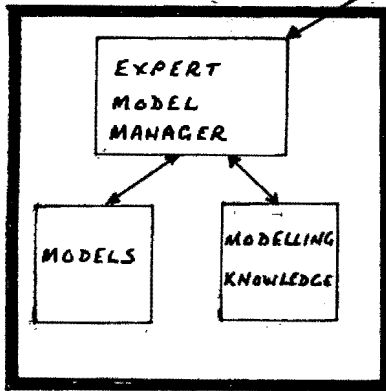
EXPERT
TOOL
MGMT.
SYSTEM



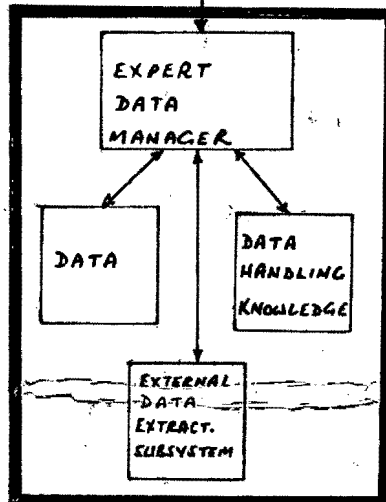
EXPERT
TRAINING
MGMT.
SYSTEM



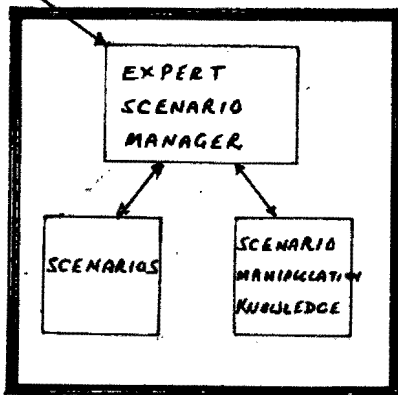
EXPERT
MODEL
MGMT.
SYSTEM



EXPERT
SCENARIO
MGMT.
SYSTEM



EXPERT DATA MGMT. SYSTEM

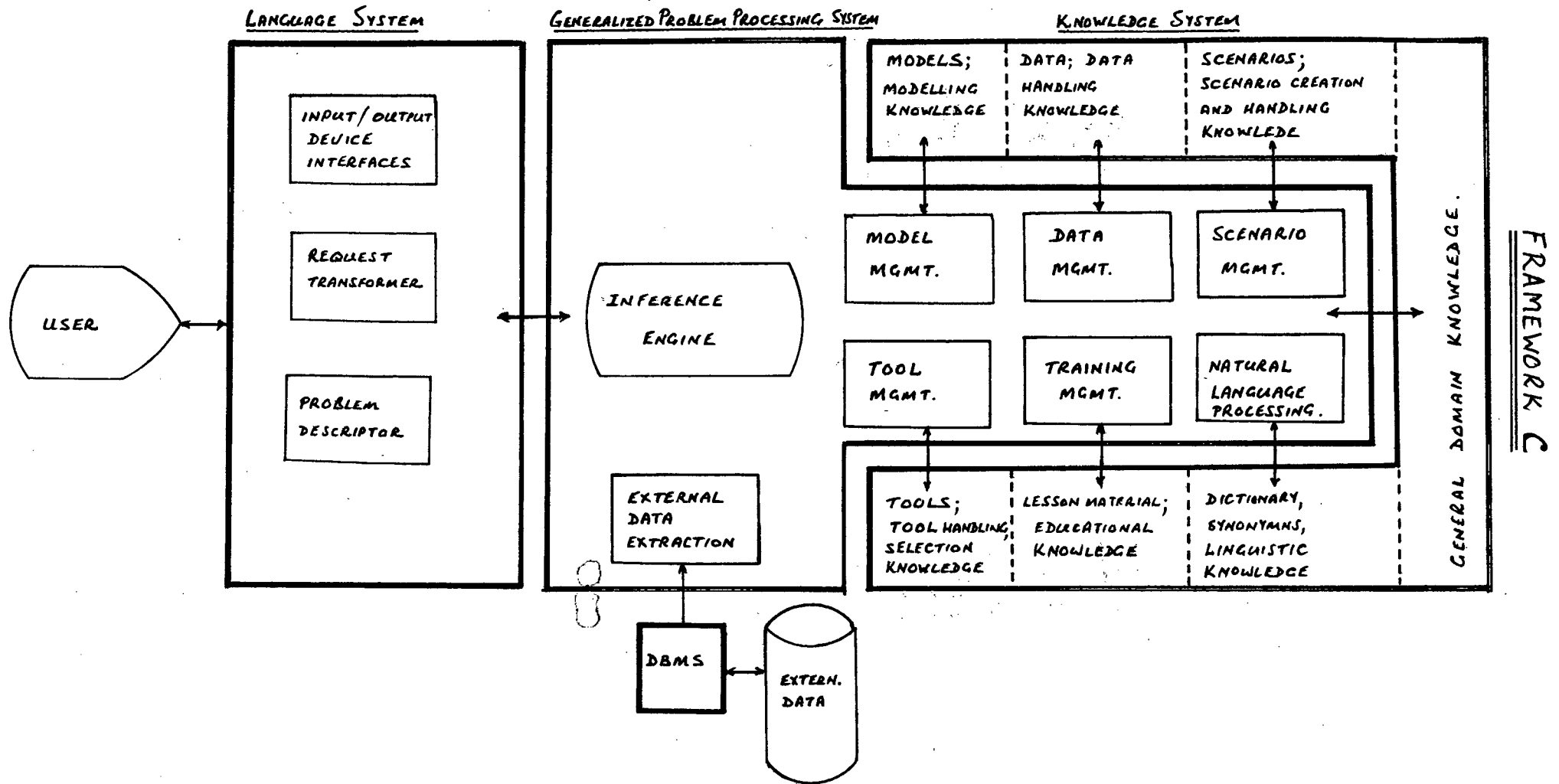


FRAMEWORK B.

RATING

STRONG POINTS.

WEAK POINTS.



FRAMEWORK C.

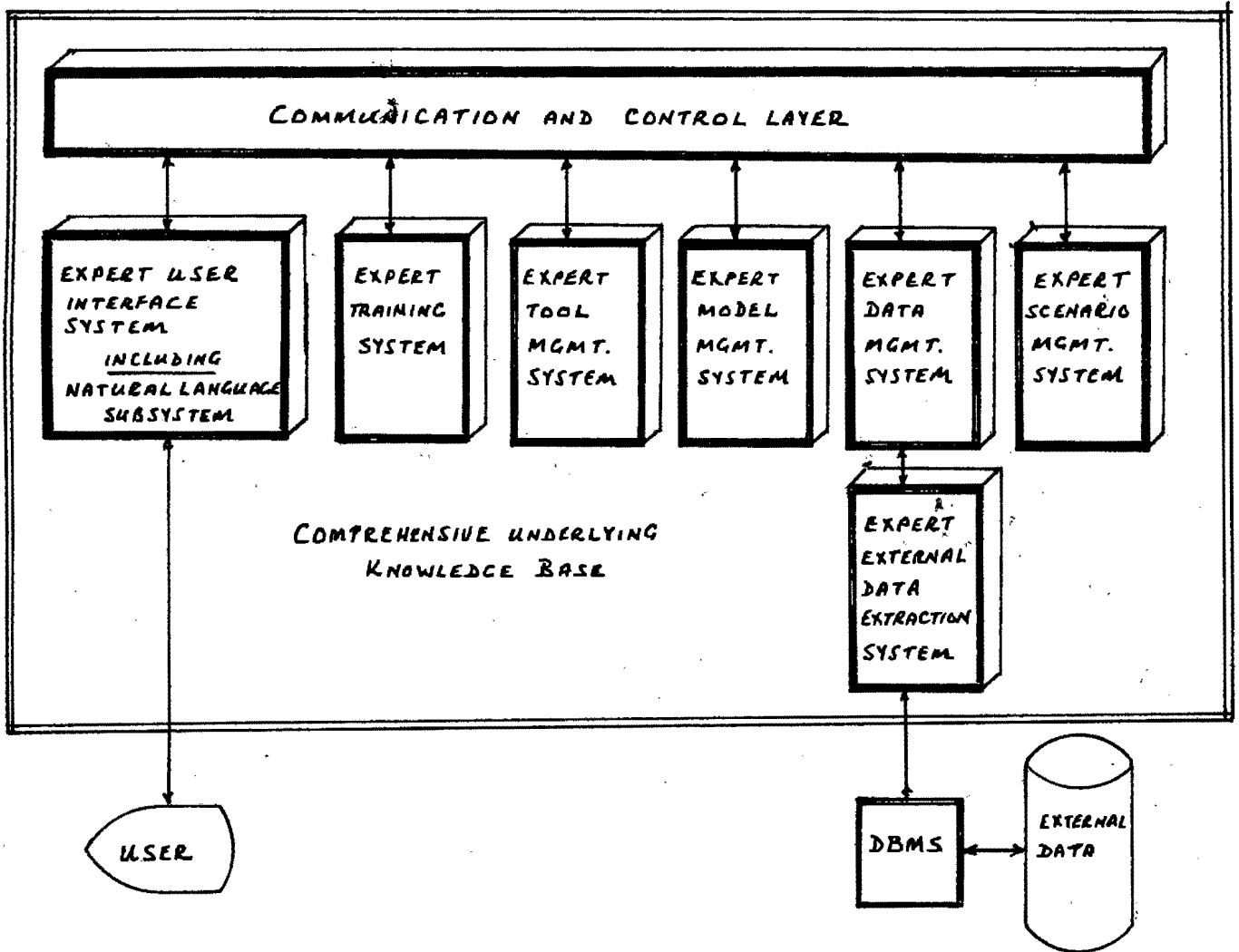
RATING

STRONG POINTS.

WEAK POINTS.

FRAMEWORK D

EXPERT DECISION SUPPORT SYSTEM



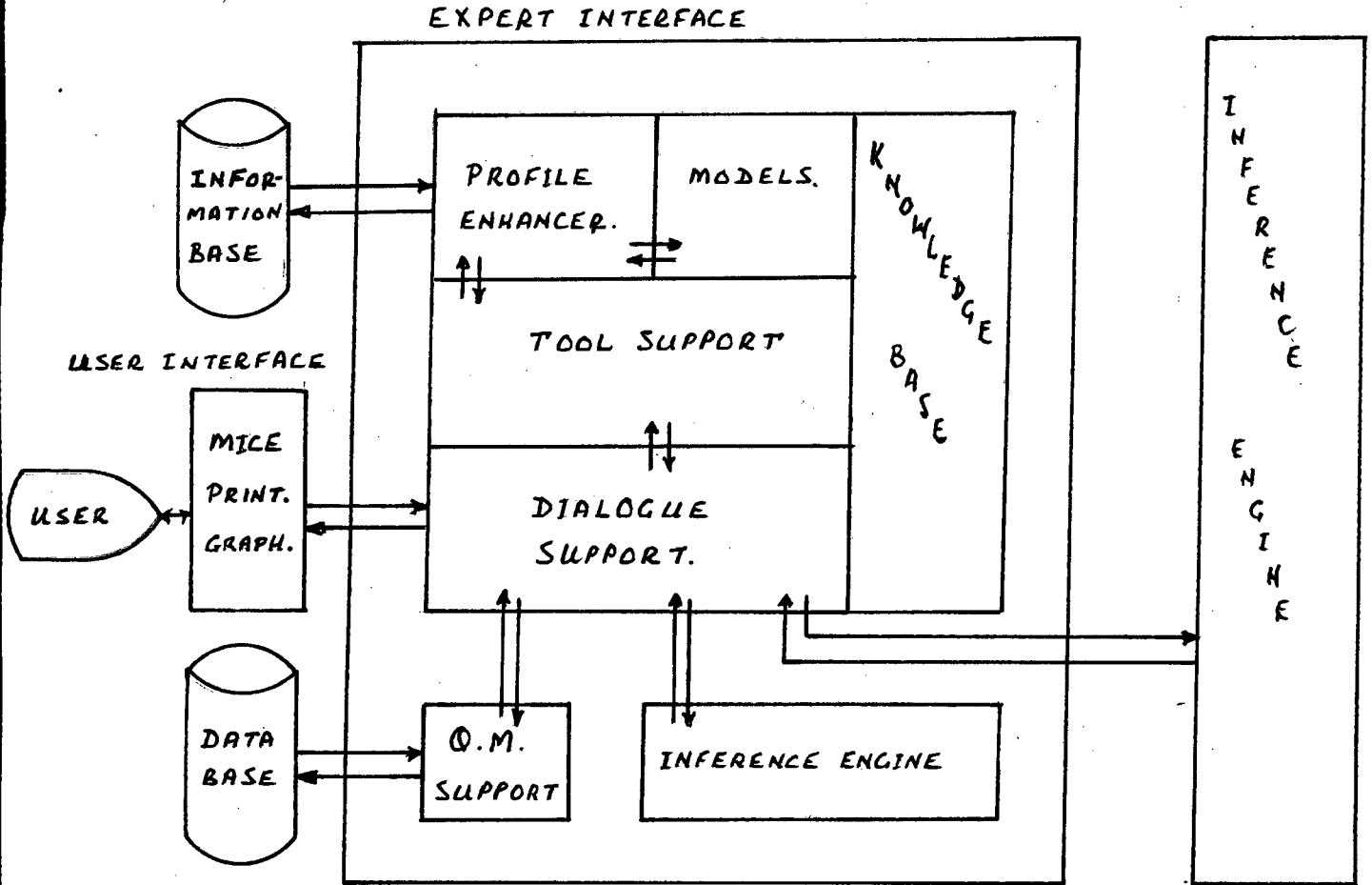
FRAMEWORK D.

RATING

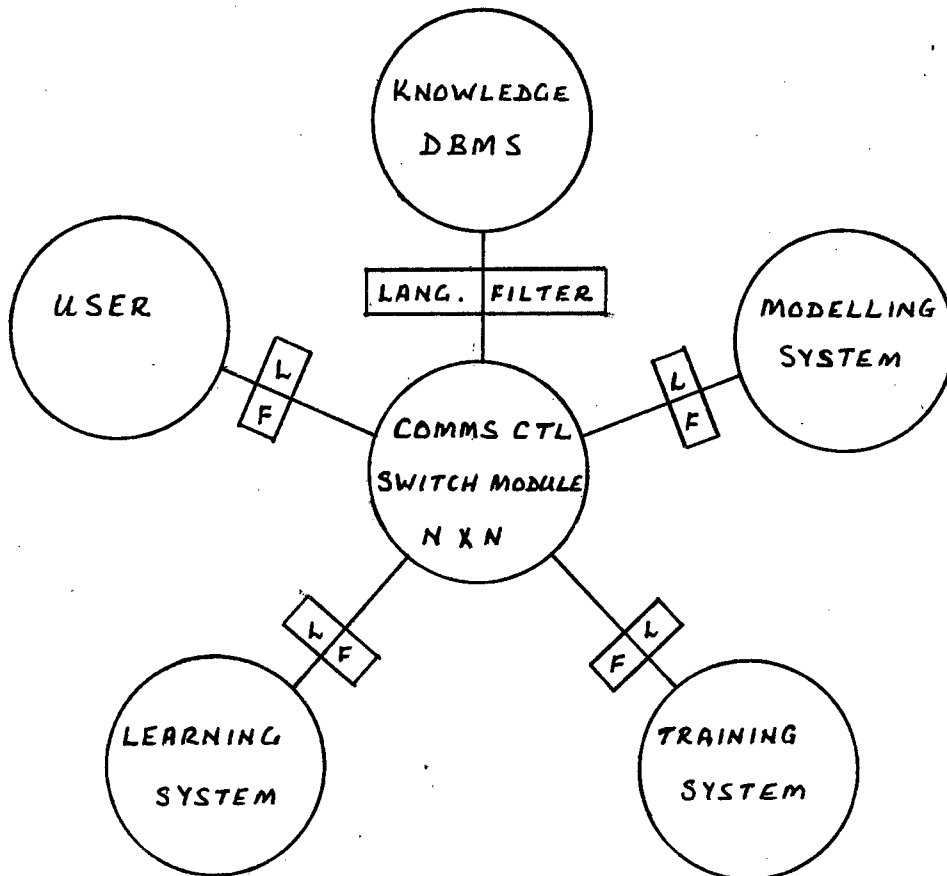
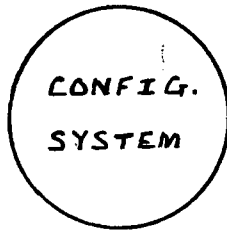
STRONG POINTS.

WEAK POINTS.

FRAMEWORK E



FRAMEWORK F



PART C.

ADDITIONAL FRAMEWORKS FOR CONSIDERATION.

Additional components specified in part A obviously do not appear in any of the frameworks depicted in part B. In the event that you have proposed such components in part A it will be appreciated if you put forward a framework which includes those additional components.

Even if you did not propose any new components for consideration but you do have suggestions in terms of the arrangement of components in an architectural framework, please feel free to submit your ideas at this stage.

Assign a rating to your framework and briefly discuss its strong and weak points.

APPENDIX 2.

NAME AND ADDRESS LIST OF DELPHI PARTICIPANTS.

Mr. J.K. Mattison
c/o Dept. of Accounting
UCT
Private Bag
RONDEBOSCH
7700

Dr. M. Link
Dept. of Computer Science
UCT
Private Bag
RONDEBOSCH
7700

Prof. H. Messerschmidt
Dept. of Computer Science
UOVS
BLOEMFONTEIN
9300

Mr. R. Trott
c/o Old Mutual Properties
P.O.Box 66
CAPE TOWN
8000

Mr. J. Miller
Graduate School of Business
UCT
Private Bag
RONDEBOSCH
7700

Mr. J. Carson
Individual Information
Support Centre
Old Mutual
P.O.Box 66
CAPE TOWN
8000

Mr. Theo Vith
26 Luisa Way
HOUT BAY
7800

BIBLIOGRAPHY.

- Ackoff RL 1967 Management Misinformation Systems. Management Science, Dec.1967, pp. B147-B156.
- Ackoff RL 1979a The future of operational research is past, Journal Oper. Res. Soc., vol.30, no.2, 1979, pp. 93-104.
- Ackoff RL 1979b Resurrecting the future of operational research, Journal Oper. Res. Soc., vol.30, no.3, 1979, pp. 189-199.
- Ackoff RL, Sasieni MW 1968 Fundamentals of Operations Research, Wiley and Sons, New York, 1968, pp. 455.
- Affleck-Graves J, Money A, Uliana E 1987 Quantitative Methods in a Developing Country: Managers' Perceptions and Desires, Unpublished Working Paper, UCT, 1987. (Accepted for publication in the October 1987 edition of Omega.)
- Alavi M 1984 An Assessment of the Prototyping Approach to Information Systems Development, Comm. of the ACM, Vol.27, no.6, June 1984, pp. 556-566.
- Alexander T 1982a Practical uses for a "useless" science. FORTUNE, May 31, 1982, pp. 139-145.
- Alexander T 1982b Computers on the road to self-improvement. FORTUNE, June 14, 1982, pp. 147-160.
- Alexander T 1984 Why computers can't outthink the experts. FORTUNE, August 20, 1984, pp. 99-108.
- Alter SL 1976 How Effective Managers Use Information Systems. Harvard Business Review, Dec.1976, pp. 97-104.
- Alter SL 1977 A taxonomy of decision support systems. Sloan Management Review, vol.19, no.1, pp. 39-56.
- Alter SL 1980 Decision Support Systems: Current practice and continuing challenges, Addison-Wesley, Reading, Mass., 1980.
- Andriole SJ 1985 The promise of Artificial Intelligence. Journal of Systems Management, July 1985, pp. 8-17.

- Ariav G, Ginzberg MJ 1985 DSS design: a systemic view of decision support. Comm. of the ACM, Oct.1985, vol.28, no.10, pp. 1045-1052.
- Bahl HC, Hunt RG 1984 A Framework for systems analysis for decision support systems, Information & Management, vol.7, no.3, 1984, pp. 121-131.
- Bahl HC, Hunt RG 1985 Problem-solving strategies for DSS design. Information and Management, 8(1985), pp. 81-88.
- Bariff ML, Ginzberg MJ 1980 MIS and the behavioral sciences: research patterns and prescriptions, Proc. of the First Int. Conf. on Information Systems, Dec. 1980, pp. 49-58.
- Barr A, Cohen PR, Feigenbaum EA 1981 The Handbook of Artificial Intelligence, volumes 1 and 2, William Kaufman Inc., Los Altos, 1981.
- Bates M, Bobrow RJ 1984 Natural Language Interfaces: What's here, what's coming and who needs it, Artificial Intelligence Applications for Business, Reitman W (ed.), Ablex Publ.Co., Norwood, NJ, 1984, pp. 179-194.
- Belkin NJ, Hennings RD, Seeger T 1984 Simulation of a distributed expert-based information provision mechanism, Information Technology: Research and Development, vol.3, no.3, 1984, pp. 122-141.
- Ben-Bassat M, Freedy A 1982 Knowledge requirements and management in expert decision support systems for military) situation assessment. IEEE Transactions on systems, man and cybernetics, vol. SMC-12, no. 4, July/august 1982, pp. 479-489.
- Benbasat I, Dexter AS 1982 Individual differences in the use of decision support aids, Journal of Accounting Research, vol.20, no.1 (Spring 1982), pp. 1-11.
- Benbasat I, Dexter AS 1986 An Investigation of the Effectiveness of Color and Graphical Information Presentation under Varying Time Constraints, MIS QUARTERLY, vol.10, no.1, March 1986, pp. 59-83.
- Benbasat I, Dexter AS, Todd P 1986 An Experimental Program Investigating Color-Enhanced and Graphical Information Presentation: An Integration of the Findings, Communications of the ACM, vol.29, no.11, Nov. 1986, pp.1094-1105.

- Benbasat I, Taylor RN 1982 Behavioural aspects of information processing for the design of management information systems, IEEE Trans. Systems, Man, Cybernetics, vol.SMC-12, no.4. (July/Aug.), pp. 439-450.
- Bennet JL 1983 Building Decision Support Systems, Addison-Wesley, Reading, Massachusetts, 277 pp.
- Bennett JL 1983 Analysis and design of the user interface for Decision support Systems, Building Decision Support Systems, JF Bennet (ed.), Addison-Wesley, Reading, Massachusetts, pp. 41-64.
- Beveridge WIB 1957 The Art of Scientific Investigation, W.W. Norton & Co., Inc., New York, 1957, 178 pp.
- Bic L, Gilbert JP 1986 Learning from AI: New trends in database technology, Computer, March 1986, vol.19, no.3, pp. 44-54.
- Biermann AW, Rodman RD, Rubin DC, Heidlage JF 1985 Natural Language with Discrete Speech as a Mode for Human-to-Machine Communication, Comm. of the ACM, vol.28, no.6, June 1985, pp. 628-636.
- Blanning RW 1979 The functions of a decision support system. Information and Management, 2(1979), pp. 87-93.
- Blanning RW 1983a Issues in the design of relational model management systems. National Computer Conference, 1983.
- Blanning RW 1983b What is happening in DSS? Interfaces, vol. 13, no.5, October 1983, pp. 71-80.
- Blanning RW 1984a Issues in the design of expert systems for management. National Computer Conference, 1984.
- Blanning RW 1984b Management Applications of expert systems. Information and Management 7, (1984), pp. 311-316.
- Blanning RW 1984c Conversing with management information system in natural language. Communications of the ACM, March 1984, vol.27, no.3, pp. 201-207.
- Blanning RW 1985 Expert Systems for management: Research and Applications. Journal of Information Science 9(1985), pp. 153-162.
- Bonccek RH, Holsapple CW, Whinston AB 1981a Foundations of Decision Support Systems. Academic Press, NY, 393p.

- Bonczek RH, Holsapple CW, Whinston AB 1981b A Generalized Decision Support System using predicate calculus and network data base management. Operations Research, vol.29, no.2, March-April 1981, pp. 263-281.
- Bonczek RH, Holsapple CW, Whinston AB 1982 The evolution from MIS to DSS: extension of data management to model management, Decision Support Systems, Ginzberg MJ, Reitman WR, Stohr EA (eds.), North-Holland Publishing Co., New York, 1982, pp. 61-78.
- Bonczek RH, Holsapple CW, Whinston AB 1983 The DSS development system. National Computer Conference, 1983.
- Bonczek RH, Holsapple CW, Whinston AB 1984 Developments in Decision Support Systems. Advances in Computers, vol.23, 1984, pp. 141-175.
- Bonini CP 1978 Computers, modeling and management education, California Management Review, Winter 1978, vol.21, no.2, pp. 53-54.
- Bonnet A 1985 Artificial Intelligence: Promise and Performance, Prentice-Hall, Englewood Cliffs, NJ, 1985, pp.221.
- Borgida A, Greenspan S, Mylopoulos J 1985 Knowledge Representation as the basis for Requirements Specifications, Computer, April 1985, vol.18, no.4, pp. 82-91.
- Borko H 1983 Information and knowledge worker productivity. Information processing and management, vol.19, no.4, pp. 203-212.
- Boulding KE 1956 General Systems Theory. Management Science, vol. 2, 1956, pp. 197-208.
- Brachman JR, Amarel S, Engelman C, Engelmores RS, Feigenbaum EA and Wilkins DE 1983 What are Expert Systems?, Building Expert Systems, Hayes-Roth F, Waterman DA and Lenat DB (Eds.), Addison-Wesley, pp. 31-57.
- Brown JS, Burton RR, deKleer J 1982 Knowledge engineering and pedagogical techniques in SOPHIE I, II and III, Intelligent Tutoring Systems, D.Sleeman and JS.Brown (eds.), Academic Press, NY, 1982, pp. 227-282.

- Brown M 1985 Potential for change. Data Processing, vol.27, no.4, May 1985, pp. 35-39.
- Buchanan BG 1985 Expert Systems, Journal of Automated Reasoning, Wos L (ed.), vol.1, no.1, 1985, pp. 28-35.
- Buchanan BG, Duda RO 1983 Principles of rule-based expert systems. Advances in Computers, vol.22, 1983, pp. 164-216.
- Buchanan BG, Barstow D, Bechtal R, Bennet J, Clacey W, Kulikowski C, Mitchell T, Waterman DA 1983 Constructing an Expert System IN: Hayes-Roth F, Waterman DA and Lenat DB (Eds.): Building Expert Systems, Addison-Wesley, 1983, pp.127-167.
- Buede DM, Yates G, Weaver CA 1985 Concept design of a program manager's decision support system, IEEE Trans. Systems, Man, Cybern., July/Aug.1985, vol. SMC-15, no.4, pp. 457-469.
- Bundy A 1985 Intelligent Front Ends, Research and Development in Expert Systems, Bramer MA (ed.), Cambridge University Press, Cambridge, England, 1985, pp. 192-203.
- Carbonell JG, Michalski RS, Mitchell TM 1983 An overview of machine learning IN: Michalski RS, Carbonell JG and Mitchell TM (Eds.) Machine Learning: An Artificial Intelligence Approach, Tioga, Palo Alto, pp. 3-23.
- Carlson ED 1983a An Approach for designing Decision Support Systems, Building Decision Support Systems, JF Bennet (ed.), Addison-Wesley, Reading, Mass., pp. 15-40.
- Carlson ED 1983b Developing the user interface for Decision Support Systems, Building Decision Support Systems, JF Bennet (ed.), Addison-Wesley, Reading, Mass., pp. 65-88.
- Catton D 1985 AI tools for DP - programming under uncertainty. Data processing, vol.27, May 1985, pp. 24-27.
- Chandrasekaran B 1984 Expert Systems: Matching techniques to tasks, Artificial Intelligence Applications for Business, Reitman W (ed.), Ablex Publ.Co., Norwood, NJ, 1984, pp. 41-64.

- Charniak E, McDermot D 1985 Introduction to Artificial Intelligence, Addison-Wesley Publ.Co., Reading, Mass., 1985, pp.701.
- Churchman CW 1968 The System Approach, Dell, New York, 1968.
Clancey WJ 1986 Heuristic Classification in Knowledge Based Problem Solving, Janusz S. Kowalik (ed.), Prentice-Hall, New Jersey, 1986, pp.1-67.
- Coelho H 1983 PROLOG: a programming tool for logical domain modeling IN: H.G.Sol (ed.) Processes and Tools for Decision Support, North-Holland Publ., Amsterdam, 1983.
- Cohen J 1985 Describing PROLOG by its Interpretation and Compilation, Communication of the ACM, vol. 28, no. 12, December 1985, pp. 1311-1324.
- Cohen MD, March JG, Olsen JP 1972 A Garbage Can Model of Organizational Choice, Administrative Science Quarterly, vol.17, no.1, 1972, pp. 1-25.
- Cohen PR, Feigenbaum EA 1981 The Handbook of Artificial Intelligence, volume 3, William Kaufman Inc., Los Altos, CA, 1981.
- Connell J, Brice L 1984 Rapid Prototyping, Datamation, Aug.15, 1984, pp. 93-100.
- Corella F and co-worker(s) 1984 Cooperative responses to Boolean queries, Proc. IEEE Data Eng. Conf., 1984.
- Cyert RM, March JG 1963 A Behavioural theory of the firm. Prentice-Hall, Englewood Cliffs, NJ, 1963.
- Davis MW 1984 Anatomy of Decision Support, Datamation, June 15, 1984, pp. 201-208.
- Davis RE 1985 Logic Programming and PROLOG: a tutorial. IEEE Software, Sept. 1985, pp. 53-63.
- Delbecq AL, Van De Ven AH, Gustafson DH 1975 Group techniques for program planning: a guide to nominal group and Delphi process, Scott Foresman and Co., Glenview, Ill., 1975.
- Demolombe R 1981 Assigning meaning to ill-defined queries expressed in predicate calculus language, Advances in Database Theory, vol.1, H.Gallaire, J.Minker, JM.Nicolas (eds.), Plenum, NY, 1981, pp. 367-395.

- DeSanctis G 1984 Computer Graphics as Decision Aids: Directions for Research, Decision Science, vol.15, no.4, Fall 1984, pp. 463-487.
- Dolk DR, Konsynski BR 1984 Knowledge representation for model management systems. IEEE Transactions on Software Engineering, vol SE-10, no. 6, Nov 1984, pp. 619-629.
- Douglas RJ 1985 A Qualitative Assessment of Parallelism in Expert Systems. IEEE Software, May 1985, pp. 70-81.
- Eilon S 1969 What is a Decision?, Management Science, vol.16, no.4, Dec.1969, pp. B172-189.
- Elam JJ, Henderson JC 1983 Knowledge Engineering Concepts for Decision Support Systems design and implementation. Information and Management, 6(1983), pp. 109-114.
- Elam JJ, Henderson JC, Miller LW 1980 Model management systems: An approach to decision support in complex organizations. Proc. 1st Int. Conf. Information Systems, Philadelphia, PA, Dec. 1980, pp. 98-110.
- Ellis P 1983 Expert Systems - a key innovation in professional and managerial problem solving. Information Age, vol. 5, no. 1, Jan. 1983, pp. 2-6.
- Emory CW, Niland P 1968 Making Management Decisions, Houghton Mifflin, Boston, 1968.
- Feigenbaum EA, McCorduck P 1983 The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World. Addison-Wesley, Reading, Massachusetts, 275 pp.
- Ferris D 1983 The little engine that can, Comput. Dec., vol. 15, no. 5, 1983.
- Ford FN 1985 Decision Support Systems and Expert Systems: a Comparison. Information And Management, 8(1985), pp. 21-26.
- Franz CR, Robey D 1984 An Investigation of User-led System Design: Rational and Political Perspectives, Comm. of the ACM, Dec.1984, vol.27, no.12, pp. 1202-1209.

- Frenkel KA 1985 Toward automating the software development cycle. Comm. of the ACM, June 1985, vol.28, no.6, pp. 578-589.
- Gale W, Pregibon D 1983 Building an Expert Interface, Bell Telephone Laboratories, Murray Hill, NJ. Unpublished.
- Gallaire H, Minker J, Nicolas J 1984 Logic and Databases: a deductive approach. Computing Surveys, vol.16, no. 2, June 1984, pp. 153-185.
- Gevarter WB 1983 Expert Systems: Limited but powerful. IEEE SPECTRUM, Aug. 1983, pp. 39-45.
- Ghani JA 1981 The effects of information representation and modification on decision performance, Ph.D. dissertation, Wharton School, Univ. of Pennsylvania, 1981.
- Gilbert GN 1985 Decision Support in large organizations. DATA PROCESSING, vol.27, no.4, May 1985, pp. 28-30.
- Ginzberg MJ 1978 Redesign of managerial tasks: a requisite for successful decision support systems, MIS Quarterly, vol.2, no.1, March 1978, pp. 39-52.
- Ginzberg MJ, Reitman WR, Stohr EA (editors) 1982 Decision Support Systems, North-Holland, NY, 174 pp.
- Ginzberg MJ, Stohr EA 1982 Decision support systems: Issues and perspectives, Decision Support Systems, Ginzberg MJ, Reitman WR, Stohr EA (eds.), North-Holland Publishing Co., New York, 1982, pp.9-31.
- Gorry AG, Krumland RB 1983 Artificial Intelligence Research and Decision Support Systems in Building Decision Support Systems, JF Bennet (ed.), Addison-Wesley, Reading, Massachusetts, pp. 205-219.
- Gorry AG, Scott Morton MS 1971 A framework for management information systems, Sloan Management Review, vol.13, no.1, pp. 55-70.
- Graham RJ 1971 Management Science Process - on the culture of management science, Interfaces, vol.2, no.1, Nov.1971, pp. 20-24.
- Graham RJ 1980 Management Science Process - on the culture of management science, Interfaces, vol.10, no.1, Febr.1980, p. 48.

- Gregg G 1984 Thinking about Artificial Intelligence, Institutional Investor, Jan. 1984, pp. 239-241.
- Gurd JR, Kirkham CC, Watson I 1985 The Manchester Prototype Dataflow Computer. Communications of the ACM, vol.28, no. 1, January 1985, pp. 34-52.
- Hackathorn RD, Keen PGW 1981 Organizational strategies for personal computing in decision support systems, MIS Quarterly, vol.5, no.3, 1981, pp. 21-27.
- Hammer C 1984 Beyond the Data Processing Horizon. Proceedings ACM'84 Annual Conference, 1984, Oct. 8-10, pp. 281-286.
- Harmon P, King D 1985 Expert Systems: Artificial Intelligence in Business. John Wiley & Sons, New York, 1985, pp.283.
- Harrison EF 1981 The Managerial Decision-making Process. Houghton Mifflin Co., Boston, 391pp.
- Hayes-Roth F 1984 The machine as partner of the new professional. IEEE SPECTRUM, June 1984, pp. 28-31.
- Hayes-Roth F, Waterman DA, Lenat DB (editors) 1983 Building Expert Systems. Addison-Wesley, 444 pp.
- Henderson JC, Schilling DA 1985 Design and implementation of decision support systems in the public sector, MIS Quarterly, Jun.1985, pp.157-169.
- Hewitt C, Smith B 1975 Towards a Programming Apprentice, IEEE Trans. Software Enq., SE-1(1), pp. 26-45.
- Hirouchi T, Kosaka T 1984 An effective database formation for decision support systems, Information & Management, 1984, no.7, pp. 183-195.
- Hogue JT, Watson HJ 1985 An examination of decision-makers' utilization of decision support system output, Information & Management 8 (1985), pp. 205-212.
- Holmes FW 1985 Whatever happened to operations research? Journal of Systems Management, April 1985, pp. 5-6.
- Holsapple CW, Moskowitz H 1980 A Conceptual Framework for studying Complex Decision Processes, Policy Science, vol.12, no.1, June 1980.

- Hong R 1983 What managers can expect from Artificial Intelligence. Management Review, vol.72(12), Dec 1983. pp. 14-19.
- Huber G 1981 The nature of organizational decision making and the design of decision support systems. MIS Quarterly, June 1981, pp. 1-11.
- Huber G 1984 Issues in the Design of Group Decision Support Systems, MIS Quarterly, vol.8, no.3, Sept.1984, pp. 195-204.
- Hurst EG, Ness DN, Gambino TJ, Johnson TH 1983 Growing DSS: A Flexible Evolutionary Approach, Building Decision Support Systems, JL.Bennett (ed.), 1983, pp. 111-132.
- Ittman HW 1984 Decision Support Systyms: a survey. SA Journal of Business Management, vol. 15(4), pp. 189-196.
- Ives B 1982 Graphical user interfaces for business information systems, MIS Quarterly, Special Issue, December 1982, pp. 15-42.
- Jarke M, Vassiliou Y 1984 Coupling Expert Systems with Database Management Systems, Artificial Intelligence Applications for Business, Reitman W (ed.), Ablex Publ.Co., Norwood, NJ, 1984, pp. 65-85.
- Johnson J 1984 Expert Systems: For You?, Datamation, vol. 30, no. 2, Febr. 1984, pp. 82-88.
- Jones KS 1985 Natural language interfaces for Expert Systems: an introductory note. Research and Development in Expert Systems, Bramer MA (ed.), Cambridge University Press, Cambridge, England, 1985, pp. 85-94.
- Keen PGW 1976 Computer systems for Top Managers: A Modest Proposal, Sloan Management Review, vol.18, no.1, 1976, pp. 1-17.
- Keen PGW 1980a Decision Support Systems: a research perspective. Sloan WP no. 1117-80, CISR no.54, March 1980.
- Keen PGW 1980b Adaptive design for decision support systems, Data Base, vol.12, no.1 and 2, Fall 1980, pp. 15-25 (31-40?).

- Keen PGW 1981 Value Analysis: Justifying Decision Support Systems. MIS Quarterly, March 1981, pp. 1-15.
- Keen PGW, Gambino TJ 1983 Building a decision support system: The mythical man-month revisited, Building Decision Support Systems, JF Bennet (ed.), Addison-Wesley, Reading, Massachusetts, pp. 133-172.
- Keen PGW, Scott Morton MS 1979 Decision support Systems: an organizational perspective, Addison-Wesley, Reading, Mass., 1979.
- Kellogg C 1986 From Data Management to Knowledge Management, Computer, January 1986, vol.19, no.1, pp. 75-84.
- Kickert WJM 1980 Organization of Decision-making: A Systems -theoretical Approach, North-Holland, 277 pp.
- Kidd A 1985 Human Factors in Expert Systems. DATA PROCESSING, vol. 27, no.4, May 1985, pp. 15-17.
- Kidd A, Cooper MB 1985 Man-Machine interface issues in the construction and use of an expert system. International Journal of Man-Machine Studies, 22(1985), pp. 91-102.
- King JJ 1981 QUIST: A system for semantic query optimization in relational databases, Proc. 7th Int. Conf. on Very Large Data Bases, Cannes, France, Sept.1981. IEEE, NY, pp. 510-517.
- Kirsch W 1977 Einführung in die Theorie der Entscheidungsprozesse, 2nd ed., Gabler, Wiesbaden, 1977.
- Konopasek M, Jayaraman S 1984 Expert Systems for personal computers, Byte, May 1984, pp. 137-156.
- Kotter JP 1982 What Effective General Managers Really Do, Harvard Business Review, 60(1982), pp. 156-167.
- Koontz H, O'Donnell C 1972 Principles of Management: An Analysis of Managerial Functions, 5th ed., McGraw-Hill, NY, 1972.
- Kowalski R 1979 Logic for Problem Solving, Elsevier North Holland Inc., 1979, 287 pp.

- Kowalski R 1984 AI and Software Engineering. DATAMATION, Nov 1984, pp. 92-101.
- Lehner PE, Probus MA, Donnell ML 1985 Building decision aids: Exploiting the synergy between Decision Analysis and Artificial Intelligence. IEEE Trans. Systems, Man, Cybern., July/Aug. 1985, vol. SMC-15, no.4, pp. 469-474.
- Lenat DB, Davis R, Doyle J, Genesereth M, Goldstein I and Schrobe H 1983 Reasoning about Reasoning IN: Hayes-Roth F, Waterman DA and Lenat DB (Eds.) Building Expert Systems, Addison-Wesley, pp. 219-239.
- Lerner AW 1976 The politics of organizational decision making: strategy, cooperation and conflict, Sage Publications, Beverly Hills, Cal., 1976.
- Li D 1984 A PROLOG Database System. Research Study Press, England, 207 pp.
- Little JDC 1970 Models and Managers: The Concept of a Decision Calculus, Management Science, vol.16, no.8, Apr. 1970, pp. B466-B485.
- Loughran BP, Cocks PJ 1977 Airline programme planning in British Airways European Division, Interfaces, vol.7, no.2, February 1977, pp. 34.
- Lucas HJ (Jnr) 1981 An experimental investigation of the use of computer-based graphics in decision making, Management Science, vol.27, no.7, July 1981, pp. 757-768.
- March JG, Simon HA 1958 Organizations, Wiley, 1958.
- Mason R, Mitroff I 1973 A program for research on management information systems, Management Science, vol.19, no.5, Jan. 1973, pp.475-487.
- Mason R, Mitroff I 1981 Challenging Strategic Planning Assumptions, John Wiley, New York, 1981.
- Mayr HC, Lockemann PC, Bever M 1985 A Framework for application systems engineering, Information Systems, vol.10, no.1, 1985, pp. 97-111.
- McDermot J 1984 Building Expert Systems in Artificial Intelligence Applications for Business, Reitman W (ed.), Ablex Publishing Corporation, NJ., 1984, pp. 11-22.

- McGuire JW 1964 Theories of Business Behaviour, Prentice-Hall, Englewood Cliffs, NJ, 1964.
- McKeown KR 1985 The need for text generation, AFIPS Conf. Proc., vol.54, 1985 National Computer Conference, Chicago, Illinois, July 1985, pp. 87-92.
- Meador CL, Guyote MJ, Keen PGW 1984 Setting Priorities for DSS Development, MIS Quarterly, vol.8, no.2, June 1984, pp. 117-130.
- Meador CL, Mezger RA 1984 Selecting an End User Programming Language for DSS Development, MIS Quarterly, vol.8, no.4, Dec.1984, pp. 267-281.
- Medsker LR 1984 An Interactive Decision Support System for Energy Policy Analysis. Comm. of the ACM, Nov.1984, vol.27, no.11, pp. 1122-1128.
- Michaelsen R, Michie D 1983 Expert Systems in Business. DATAMATION, Nov. 1983, pp. 240-246.
- Michaelsen R, Michie D and Boulanger A 1985 The Technology of Expert Systems, Byte, vol.10, no.4, April 1985, pp. 303-312.
- Michalski RS 1983 A Theory and Methodology of Inductive Learning, in Michalski, Carbonell and Mitchell (eds.) 1983, pp. 83-134.
- Michalski RS, Carbonell JG, Mitchell TM (editors) 1983 Machine Learning: An Artificial Intelligence Approach. Tioga Publishing Co., Palo Alto, 572pp.
- Minch RP, Burns JR 1983 Conceptual design of decision support systems utilizing management science models. IEEE Trans. Syst. Man Cybern., July 1983, vol. SMC-13, no.4, pp. 549-557.
- Mintzberg H, Raisinghani D, Theoret A 1976 The Structure of 'Unstructured' Decision Processes, Administrative Quarterly, vol.21, June 1976, pp. 246- 274.
- Mitroff IM 1983 Stakeholders of the Organizational Mind, Jossey Bass, San Francisco, 1983.
- Moore JH, Chang MG 1980 Design of Decision Support Systems, Data Base, vol.12, no.1 and 2 (Fall 1980), pp. 8-14.

- Moore JH, Chang MG 1983 Meta-design considerations in building DSS, Building Decision Support Systems, JL Bennet (ed.), Addison-Wesley Publishing Co., Reading, Massachusetts, 1983, pp. 173-204.
- Nash DR 1977 Building EIS, a utility for decisions, Data Base, vol.8, no.3, 1977, p.3.
- Naylor C 1983 Build your own Expert System: Artificial Intelligence for the aspiring microcomputer, Halsted Press (John Wiley & Sons, New York), 1983, pp.249.
- Naylor TH 1982 Decision support systems or whatever happened to MIS?, Interfaces, vol.12, no.4, pp. 92-97.
- Negoita CV 1985 Expert Systems and Fuzzy Systems, Benjamin/Cummings Publ.Co., Menlo Park, California, 1985, pp.190.
- Nowak EJ, Szablowski BF 1984 Expert Systems in Scientific information exchange. Journal of Information Science, 8(1984), pp. 103-111.
- Ofstad H 1961 An Enquiry into the freedom of Decision, Allen and Unwin, 1961.
- O'Keefe RA 1982 Automated Statistical Analysis, Working Paper 104, Dept. of Artificial Intelligence, Edinburgh, 1982.
- O'Shea T 1979 Rule-based computer tutors, Proc. 1979 AISB Summer School, 1979, pp. 226-232.
- O'Shea T, Eisenstadt M (editors) 1984 Artificial Intelligence: Tools, techniques and Applications, Haper and Row, NY, 497 pp.
- Oxman SW 1985 Expert Systems represent ultimate goal of strategic decision making, Data Management, Apr. 1985, pp. 36-38.
- Pettigrew AM 1972 Information Control as a Power Resource, Sociology, vol.6, no.2, May 1972, pp. 187-204.
- Pettigrew AM 1973 The politics of organizational decision making, Tavistock, London, England, 1973.
- Pountain D 1985 Parallel Processing, Byte, vol.10, no.5, May 1985, pp. 385-395.

- Radford KJ 1978 Information Systems for Strategic Decisions. Reston Publishing Co., Virginia, 239pp.
- Radford KJ 1981 Modern Managerial Decision making. Reston Publishing Co., Virginia, 258pp.
- Rathwell MA 1985 Information Systems support for group planning and decision-making activities. MIS Quarterly, Sept. 1985, pp. 255-271.
- Reitman W 1982 Applying Artificial Intelligence to Decision Support: Where do good alternatives come from?, Decision Support Systems, Ginzberg MJ, Reitman WR, Stohr EA (eds.), North-Holland Publishing Co., New York, 1982, pp. 155-174.
- Reitman W 1984 Artificial Intelligence Applications for Business, Ablex Publ.Co., Norwood, NJ, 1984, pp. 343.
- Remus WE, Kotteman JE 1986 Toward Intelligent Decision Support Systems: An Artificially Intelligent Statistician, MIS Quarterly, Dec.1986, pp. 403-418.
- Robinson JA 1965 A Machine-oriented Logic based on the Resolution Principle, Journal of Ass. Comput. Mach., vol. 12, no. 1, Jan. 1965, pp. 23-41.
- Rosenman BB 1980 Problems of Quantitative Models in large Management Information Systems, Interfaces, vol.10, no.2, Apr.1980, p. 104.
- Sanders GL, Courtney JF 1985 A field study of organizational factors influencing DSS success, MIS Quarterly, Mar. 1985, pp. 77-89.
- Schank RC 1982 Dynamic Memory: A Theory of Reminding and Learning in Computers and People, Cambridge University Press, Cambridge, 1982.
- Schank RC, Hunter L 1984 The Quest to Understand Thinking, Byte, vol.10, no.4, Apr.1984, pp. 143-155.
- Schank RC, Slade S 1984 Advisory Systems, Artificial Intelligence Applications for Business, Reitman W (ed.), Ablex Publ.Co., Norwood, NJ, 1984, pp. 249-265.
- Scott Morton MS 1977 Management Decision Systems: Computer-based support for decision making, Cambridge, Massachusettes: Division of Research, Harvard

University.

- Seitz CL 1985 The Cosmic Cube. Communications of the ACM, vol. 28, no. 1, January 1985, pp. 22-33.
- Shen SH 1981 Design of a Management Information System. Ph.D. Thesis, Purdue University.
- Shull FA, Delbecq AL, Cummings LL 1970 Organizational Decision Making, McGraw-Hill, NY, 1970.
- Simon HA 1945 Administrative Behaviour, Free Press, NY, 1945.
- Simon HA 1960 The new science of management decisions, Harper and Row, New York, NY, 1960.
- Simon HA 1976 Administrative Behaviour, 3rd ed., Free Press, NY, 1976.
- Simon HA 1977a The New Science of Management Decision, 2nd ed., Prentice-Hall, 1977.
- Simon HA 1977b Models of Discovery, D. Reidel Publ. Co., Dordrecht, Holland, 1977. 456 pp.
- Simons GL 1984 Introducing Artificial Intelligence, NCC Publications, Manchester, England, 1984, pp.281.
- Sloman A 1985 Why we need many knowledge representation formalisms in Research and Development in Expert Systems, Bramer, MA (ed.), Cambridge University Press, 1985, pp.163-183.
- Sowa JF 1981 A conceptual schema for knowledge-based systems, Proc. Workshop on Data Abstraction, Databases, and Conceptual Modelling, SIGMOD Record 11:2, pp. 193-195.
- Sowa JF 1984 Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, 481pp.
- Sprague RH 1980 A framework for the development of decision support systems, MIS Quarterly, Dec.1980, vol.4, no.4, pp. 1-26.
- Sprague RH, Carlson ED 1982 Building Effective Decision Support Systems. Grolier, 340pp.
- Stabell CB 1983 A decision-oriented approach to building DSS, Building Decision Support Systems, JF Bennet (ed.), Addison-Wesley, Reading, Massachusetts,

pp. 1221-260.

- Starrs A M 1985 Expert Systems - Their uses and possible impact on society. IEE Electronics and Power, Jan 1985, pp. 39-41.
- Stewart D 1985 Putting Natural Language to Work, Popular Computing, vol.4, no.11, Sept.1985, pp.64-68,123-124.
- Stephenson RW, Stephenson MK 1983 Design requirements for decision support systems for RDT&E, Information Processing & Management, vol.19, no.6, pp. 391-397.
- Sturdza P 1984 Augmented semantic networks for an enterprise knowledge base, Information Technology Research and Development, vol.3, no.1, Jan.1984, pp. 43-53.
- Swartout W 1985 Knowledge needed for Expert System explanation, AFIPS Proc. of the Nat. Comp. Conf., vol.54, 1985, pp. 93-98.
- Thierauf RJ 1982 DSS for effective planning and control: a case study approach, Prentice-Hall, NJ, 1982, pp. 536.
- Thierauf RJ 1984 Effective Management Information Systems. Charles E. Merrill Publishing Co., London, 558pp.
- Turner JA 1980 Computers in Bank Clerical Functions: Implications for Productivity and the Quality of Working Life, unpublished Ph.D. dissertation, Columbia University, 1980.
- Turner M 1985 A Consultant's view of expert systems. DATA PROCESSING, vol 27, no.4, May 1985, pp. 12-14.
- Uliana EO 1986 The identification and Evaluation of Factors Influencing the Use of Quantitative Methods in South African Business, Unpublished M.Comm. thesis, University of Cape Town, 1986.
- Vazsonyi A 1978 Decision support systems: the new technology of decision making?, Interfaces, vol.9, no.1, pp. 72-77.
- Vazsonyi A. 1982 Decision support systems, computer literacy, and electronic models, Interfaces, vol.12, no.1, pp. 74-78.

- Vince N 1985 Adviser and Reveal as frameworks for expert systems. DATA PROCESSING, vol 27, no.4, May 1985, pp. 21-23.
- Wadia MS 1966 The Nature and Scope of Management, Scott Foresman, Chicago, 1966.
- Wagner GR 1981 Decision Support Systems: the real substance. Interfaces, Apr.1981, pp. 77-87.
- Wahlster W 1981 Natural language AI systems: State of the art and research perspective, Proc. of the German Workshop on Artificial Intelligence 81, J.Siekman (ed), Springer, NY, 1981.
- Wang MS, Courtney JF 1984 A Conceptual Architecture for Generalized Decision Support System Software. IEEE Transactions on systems, man and cybernetics, vol. SMC-14, no. 5, Sept/Oct 1984, pp. 701-711.
- Wang MSY 1983 Bridging the gap between modeling and data handling in a decision support system generator. Int. Journal Information and Policy, Dec. 1983, vol.7, no.2, pp. 87-93.
- Wang MSY, Yu KC 1983 A hierarchical model for knowledge-based decision support system software. Proc. 26th IEEE Computer Soc. Int. Conf., CompCon, Fall 1983, Arlington, VA, Sept.1983, pp. 86-92.
- Watson HJ, Marett PA 1979 A Survey of Management Science Implementation Problems, Interfaces, vol.9, no.4, Aug.1979, pp. 124-128.
- Watson HJ, Hill MM 1983 Decision Support Systems or what did not happen with MIS. Interfaces, vol. 13, no. 5, Oct. 1983, pp. 81-88.
- Weber BL, Nilsson NJ (editors) 1981 Readings in Artificial Intelligence. Tioga Publishing Co., Palo Alto, 547pp.
- Wess BP 1984 Artificial Intelligence techniques speed software development. Mini-Micro Systems, vol. 17(11), Sept. 1984, pp. 127-136.
- Wiederhold G 1984 Knowledge and Database Management. IEEE Software, Jan. 1984, pp. 63-73.

- Witte E 1972 Field Research on Complex Decision-making Processes - the phase theorem, Int. Stud. of Management and Organization, 1972, pp. 156-182.
- Zadeh LA 1984 Coping with the imprecision of the real world, Comm. of the ACM, Apr.1984, vol.27, no.4, pp. 304-311.
- Zmud RW 1979 Individual differences and MIS success: a review of the empirical literature, Management Science, vol.25, no.10, 1979, pp. 966-979.