UNIVERSITY OF NEW SOUTH WALES

Master of Commerce (Honours)

Project Report

Application of KBS Technology Within Commercial Organisations

Ian Caddy
Student Id. No.: 8862990

1993

University of New South Wales
School of Information Systems
Master of Commerce (Honours) - Project Report

ABSTRACT

The major problem addressed in this project report is the application of KBS technology within commercial organisations. Specific questions addressed were:

- What problems were solved using KBS technology?
- What sort of KBS technology was applied within commercial organisations?
- What development processes were used in the KBS development?
- Does KBS development in commercial organisations use a system development methodology?

The research strategy was to follow a case study approach in which seven organisations participated. A particular KBS that had reached production stage was selected as a focus for the study. Research data was collected using both questionnaires and a series of interviews.

The major findings of the study were that: (a) organisations solved problems at the operational level rather than higher management type problems; (b) most KBS developments used expert system shell software both as a development environment as well as part of the delivery environment; (c) there was no clear common development approach but overall a prototyping or iterative development process was found to be used the most; (d) that development of the KBS occurred outside the MIS or IT area of the organisation; and (e) that most KBS developments did not follow a system development methodology.

The major conclusion of this study was that commercial organisations were still experimenting with the technology. However, in all cases the experimental approach was not in a laboratory but to verify the effectiveness of KBS technology on real and significant problems. The KBS's developed in all cases had sizeable user populations, were large systems that absorbed significant resources (both manpower and financial resources) and had extended development schedules. The development approach did not use a system development methodology as this was seen as either not necessary or to expensive for projects that may not lead to others. In terms of the actual development there was an emphasis to be cost effective by using productivity tools such as expert system shells.

Certificate of Originality

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of a university of other institute of higher learning, except where due acknowledgment is made in the text.

(Signed) .

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

Chapter Outline

Aims of the Research Project

Knowledge Based Systems Technology and Decision Making

KBS Technology - A Research Emphasis

Commercial Interest in KBS Technology

Development of KBS's

Comparing KBS and Conventional IS Development

Importance of Conducting the Research

Schematic of the Project Report

Aims of the Research Project

The major aim of this research project is to determine how commercial[1] organisations have applied knowledge based system (KBS) technology. As will be shown in this chapter and later ones, the last decade represented a significant "commercialisation" of this technology, as organisations either released KBS development products, often known as knowledge based or expert system shells, or applied KBS technology to specific information processing problems. Prior to the early 1980's interest in this technology was

---

1 *The use of the term "commercial" includes all organisations that are not primarily involved in teaching, research or a combination of these activities, i.e. the term applies to either public or private sector organisations.*

mainly restricted to academic and research institutions who wished to further refine and extend the technology itself rather than apply it to "real world" problems. Given this thrust to the research project, the research methodology to be used is a case study approach that will look at a specific KBS development for seven different organisations. As the case study approach will be followed, then obviously it would not be suitable to merely go into an organisation with only one question to ask, namely: "How has KBS technology been applied in your organisation?"

There are two problems in doing this. First the variety of answers would prevent any meaningful comparison being made across different organisations. Second, although the answers may be quite lengthy, they would in most cases tend to be rather superficial. Accordingly, it is considered that to arrive at any meaningful results, this aggregate question obviously needs to be decomposed into a number of more detailed ones. Accordingly, this research project will concentrate on the following four major areas of interest, attempting to answer the questions shown below for that area of interest:

- *Area of interest*: **Problems solved with KBS technology.**
  Research questions:

  - What are the major features of the problems being solved by KBS technology? For example, are these problems associated with mainstream issues for the organisation or alternatively are they more peripheral to that organisation's core activities?
  - How are these problems identified by commercial organisations as requiring a KBS solution? Are KBS developments initiated by the MIS or IT area, user areas or another part of the organisation?

- Are these problems long-term or short-term, that is, does the KBS development address problems with long lives or alternatively are these systems developed to address issues of limited life?
- Are there similarities between the problems solved using KBS technology for different organisations? There will be obvious functional differences evident from the different environments within which organisations find themselves. However, the comparison could be at a broader level. For example, do these KBS's address day-to-day problems faced by operational clerical staff? Alternatively, do these KBS's address more complex problems faced by middle level and higher level management?

• *Area of Interest*: **Use of KBS technology.**
Research questions:

- How was KBS technology "introduced" into the organisation? Was it an initiative of the MIS or IT area or some other area of the organisation?
- What designs are employed in developing the major components of a commercial KBS? For example, are most KBS's built with knowledge represented as production rules, and forward or backward chaining for the inferencing strategy?
- Are knowledge based system shells the major delivery environment for commercial knowledge based systems? Alternatively, are KBS's coded systems written in either artificial intelligence languages such as LISP or PROLOG, or a more conventional language such as C or C++?

- What has been the goal in using KBS technology? For example, are commercial KBS's seen by their developers and/or users to be experts, colleagues, or merely assistants?
- How are final production versions of KBS's delivered? For example, are they stand alone systems that run on microcomputers with a simple user-oriented interactive interface? Alternatively, are they highly integrated systems operating on a range of hardware platforms?

• *Area of Interest*: **Development of KBS's.**
Research questions:

- What area within the organisation undertakes the KBS development? Is it the MIS or IT area, other areas of the organisation, or by persons outside of the organisation?
- What is the attitude of various areas within an organisation to KBS development - supportive, non-committal, obstructive?
- What sort of activities are undertaken in a typical KBS development? Are these activities similar across different KBS developments by different organisations?
- What sort of documentation is generated from a KBS development? Are there any similarities in the documentation across different KBS developments by different organisations?
- Are KBS development projects more likely to be one-off activities or part of an integrated and continuing development program?

- What type of personnel are used to develop KBS? For example, do most KBS development personnel have a conventional IS backgrounds?
- Are KBS development projects large projects in terms of size of project development team, length of development schedule, and cost of development?
- What development approaches have been used in the construction of commercial KBS's? Are KBS's developed using an evolutionary approach, or is there evidence of a more structured development environment?
- Are KBS development approaches similar to those used in conventional information systems developments?

• *Area of Interest*: **KBS Development Methodologies.**

Research questions:

- Do commercial organisations have organisations possess KBS development methodologies?
- If the organisation does not have a KBS development methodology, are there any reasons for this?
- If the organisation does have a methodology, what features do these methodologies have? For example, do these methodologies follow what has been described in the research literature, follow the design of conventional methodologies, or are unique in their structure?
- If the organisation does have a KBS development methodology, what sort of documentation or other material exists describing the methodology? In addition to documentation, is there other educational support for using the methodology, such as training courses?

- If the organisation does have a KBS development methodology, how many times has this methodology been used?
- What impact has the possession of a KBS development methodology had on the overall number of KBS developments for the organisation? For instance, are those organisations that have development methodologies more likely to have developed more KBS's than organisations that do not have a KBS development methodology?

As indicated from the above questions, it is considered that this area represents a rather fruitful domain in which a number of interesting projects could be undertaken. The current research project will attempt as far as possible to answer most of the questions outlined above for each separate area of interest.

KBS Technology and Decision Making

Does KBS technology have a role to play in the problem solving activities of commercial organisations? The answer to this question can be made from a theoretical as well as a practical perspective. It seems on the evidence available within the research and other literature that the answer to this question is "YES". As demonstrated below, a fundamental activity of an organisation is to make decisions, and from a theoretical point of view, KBS technology is a decision support technology. For example, Turban (1993) provides the following comparison of the decision support delivered by knowledge based systems and decision support systems in the table shown below [p. 27]:

Table 1. Differences Between ES/KBS and DSS

Criterion	DSS	ES/KBS
Objective	Assist human decision maker	Replicate (mimic) a human adviser and replace him/her
Who makes the recommendations (decisions)?	The human and/or the system	The ES/KBS
Major orientation	Decision making	Transfer of expertise (human-machine-human) and rendering of advice
Major query direction	Human queries the machine	Machine queries the human
Nature of support	Personal, groups and institutional	Personal (mainly) and groups
Data manipulation method	Mainly algorithmic	Symbolic
Characteristics of problem area	Complex, integrated, wide	Narrow domain
Type of problems treated	Ad hoc, unique	Repetitive
Content of database	Factual knowledge	Procedural and factual knowledge
Reasoning capability	No	Yes, but limited
Explanation capability	Limited	Yes

(Source: Turban, 1993, p. 27)

Have the decision support features of both the categories shown in the table above, that is, for both knowledge based systems and decision support systems, been applied in a practical way to decision making in commercial organisations? There is considerable case study research, undertaken over the last few years, to support this contention. For example, a large amount of case study research indicates that both the knowledge based systems technology and decision support system technology, can effectively deliver some form of assistance or support, either to individuals or to groups of decision makers (Bachant and Soloway, 1989; Barker and O'Connor, 1989; Braden, et al. 1989; Johnson, et al., 1989; Lamberti and Wallace, 1990; Le Blanc and Kozar, 1990; Nunamaker, et al., 1991; Pracht and Courtney, 1988; Sabherwahl and Grover, 1989; Shpilberg, et al., 1986).

More specifically with respect to knowledge based systems technology, Sviokla (1990) reporting on case study research, has outlined that with the installation of KBS systems there are often dramatic changes in organisational procedures. Other authors, such as Mykytyn, et al. (1990), agree that the impact of KBS technology on decision making processes is not merely superficial. In fact, Mykytyn, et al. (1990) claim that organisations should seriously consider the legal ramifications of using KBS software in their day-to-day operations [p. 27]:

> *"The development of expert systems has changed dramatically in recent years based largely on concepts dealing with artificial intelligence. These efforts are evolving from very specific, academically oriented efforts ... to more managerially oriented corporate issues. Unfortunately, many proponents of these systems may be overlooking possible legal ramifications related to both the development and use of these systems. A major issue concerns the establishment of liability for the decisions and*

recommendations made by expert systems. Some liabilities could include product liability and negligence. All individuals involved with expert systems (knowledge engineers, domain experts, and users) are potentially subject to legal scrutiny. It behoves organisations involved with these systems to investigate the potential legal problems concerning them. As these systems become more pervasive, courts may look upon them for what they are: systems of knowledge and experience, not simply passive computer software."

The discussion above shows that there is evidence that KBS technology is used in decision support, and some evidence that the technology is having an ever greater impact in this area of organisational activity. Furthermore, with respect to knowledge based systems (KBS's), Waterman (1986) claims [pp. 12-14] that this technology cannot be relied upon to replace, completely and effectively, all decision making by humans within an organisation. As indicated by the first half of the table show below, there are a number of advantages at a very basic level to adopting this technology, which must be tempered by the points considered in the second half of this table:

Table 2. Comparing Human and Artificial Intelligence

The Good News for Artificial Intelligence	
Human Expertise	**Artificial Expertise**
Perishable	**Permanent**
Difficult to transfer	**Easy to transfer**
Difficult to document	**Easy to document**
Unpredictable	**Consistent**

Expensive	Affordable
The Bad News for Artificial Intelligence	
Human Expertise	**Artificial Expertise**
Creative	Uninspired
Adaptive	Needs to be told
Sensory experience	Symbolic input
Broad focus	Narrow focus
Commonsense knowledge	Technical knowledge

(Source: Waterman, 1986, pp. 12-14)

Accordingly, given that KBS technology may not have universal applicability to all problem solving activities within commercial organisations, two questions arise. First, is this contention by Waterman (1986) true, that organisations selectively apply the technology to only certain decision problems? Second, if this contention is true how do organisations select the problems with which KBS technology offers the best solution?

Decision Problems Addressed by KBS Technology

In looking at the types of problems that commercial organisations have addressed using KBS technology, the investigation should be conducted from a number of different directions. The first issue is to determine the characteristics or attributes of these problems, and to assess whether there are commonalities between the problems addressed by different organisations. The indications are that KBS technology enjoys wide applicability across various problem domains, with KBS development occurring in fields as diverse as weather forecasting (Horsfall, 1990) to auditing (Brown and Murphy, 1990).

Therefore, as stated above, at a detailed level, there will obviously be marked differences, for example, a KBS controlling the operation of a blast furnace (Lock Lee et al., 1990) will have marked differences in the sort of decision support it provides as compared to a KBS which assesses claims for an insurance company (Beinat and Tsui, 1992). Therefore, meaningful comparisons will have to be made at a higher level, either from an organisational point of view, for example, operational problems or higher level management problems. Another classification scheme or grouping of problems that knowledge based systems or expert systems can solve, has been provided by Hayes-Roth et al., (1983) in the table shown below [p.14]:

Table 3. Types of Decision Problems Addressed by KBS Technology

Category	Problem Addressed
Interpretation	Inferring situation descriptions from sensor data
Prediction	Inferring likely consequences of given situations
Diagnosis	Inferring system malfunctions from observables
Design	Configuring objects under constraints
Planning	Designing future actions or strategies
Monitoring	Comparing observations to plan vulnerabilities
Debugging	Prescribing remedies for malfunctions
Repair	Executing a plan to administer a prescribed remedy
Instruction	Diagnosing, debugging and repairing student behaviour
Control	Interpreting, predicting, repairing and monitoring system behaviours

(Source: Hayes-Roth, et al., 1983, p. 14)

This leads to the first interesting question the research project should investigate. Does Table 3 form an appropriate taxonomy with which to classify the KBS's developed by commercial organisations? Alternatively, are there categories not mentioned which should be included on the list? Which of the above form the major categories into which commercial KBS's would fall, or it is a situation that no one category predominates? And finally, is there any relationship between a judgement of the level of maturity concerning KBS technology within an organisation and the types of problems being addressed by that organisation in using this technology?

The second direction associated with problem solving using KBS technology is to determine the level of decision support, which could vary from expert level advice and guidance to more mundane assistance. Is KBS development within commercial organisations focussed on developing true "expert systems", or is their aim driven from a more pragmatic point of view where the KBS provides assistance at a much lower level. The research literature appears to indicate at least two different roads on which KBS development could travel, the high road and the low road. The differences are indicated in the sorts of definitions used by the research literature to describe these systems. It would be interesting to determine whether these definitions have any applicability to the systems developed by commercial organisations. With respect to the high road, Pigford and Baur (1990) define artificial intelligence (AI) as [p.3]:

> *"Artificial intelligence (AI) is a segment of computer science devoted to the development of computer hardware and software that is designed to imitate the human mind. As such, AI has as its main goal the task of solving the problem of making computers smarter; that is, the capability to do more and more things that are considered by people to require some form of intelligence."* .

Would this be an appropriate definition for the sorts of systems that commercial organisations are developing and deploying? Or do these systems have a completely different goal - that is, they do not attempt to emulate the human thought processes at all, but provide only an appropriate reasoning strategy to solve a business problem. Later in their book, Pigford and Baur (1990) offer the following definition of an expert system [p. 11]:

> *"In general, the goal of an expert system is to replace the human expert by capturing his or her knowledge and expertise. On the other hand, the general goal of a decision support system is to assist rather than replace an expert."*

Is this one of the objectives for the development of an expert system within a commercial organisation, that is to extract and replace the human expert with the KBS? The classic scenario in this situation is the greying expert about to retire from an organisation and the development of this KBS will ensure on-going access to the expert's knowledge and experience after that person has left the organisation. Finally, with respect to the high road, Professor Edward Feigenbaum of Stanford University (a leading researcher in expert systems) offers the following definition (Feigenbaum, 1982):

> *" ... an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution. Knowledge necessary to perform at such a level, plus the inference procedures used, can be thought of as a model of the expertise of the best practitioners of the field."*

Feigenbaum claims that the knowledge of an expert system consists of facts and heuristics. The "facts" constitute a body of information that is widely shared, publicly available and generally agreed upon by experts in a field. The "heuristics" are mostly private, little-discussed rules of good judgement (rules of plausible reasoning, rules of good guessing) that characterise expert-level decision making in the field. The performance level of an expert system is primarily a function of the size and the quality of the knowledge base it possesses. With respect to commercially developed KBS's is there an attempt to build systems based on the knowledge of "the best practitioners in the field"?

With respect to the low road, an example of this sort of KBS is characterised in the following definition provided by Jackson (1990) [p. 3]:

> *"An expert system is a computer program that represents and reasons with knowledge of some specialist subject with a view to solving problems or giving advice. ... Such a system may completely fulfil that function that normally requires human expertise, or it may play the role of an assistant to a human decision maker."*

This definition indicates that the role of the KBS can be far more flexible than the previous two by Pigford and Baur (1990) and Feigenbaum (1982). That is the KBS may not always be the superior problem solver but could assume the role of colleague in which the KBS and the human decision makers are perceived to be of equal standing, or even assume the role of an assistant to enhance the reasoning of the human decision maker using the KBS. Accordingly, one inference that can be drawn from the above definitions is that although there is some degree of commonality, there appears to be flexibility in what describes an expert or knowledge based system. Is this apparent lack of rigour evident in the commercially developed systems? Which (if any) of the above definitions more appropriately describes the sorts of systems being developed by commercial organisations? To some extent the level of decision support will be reflected in the depth of knowledge contained within the knowledge base and the flexibility of the inferencing strategy employed to resolve the problem. Do commercial KBS's possess deep knowledge - which would be indicative of expert level performance - or do they possess only shallow knowledge - which would be indicative of assistant level performance.

The final dimension in terms of decision support, is the style of delivering the final production version of the KBS. In regard to this factor there are a number of different alternatives, which to some extent are contingent upon the problem being addressed. That is the broadening of functionality in the application of this technology has provided at least three major modes of operation for installed or implemented KBS's. First, a KBS can operate as a stand alone system with little integration with other systems of the organisation (Beinat and Tsui, 1992). These systems often conduct an interactive dialogue with a human user providing most of the input data, which when combined with the knowledge stored in the knowledge base eventually allows some form of conclusion to be reached. Second, a KBS can operate in an integrated fashion, interfacing non-transparently with other IS products, for example, extracting information from transaction processing or higher level IS systems such as the aircraft docking system developed for Tullamarine airport in Victoria (Tasker, 1992). Typically with these systems inferencing is driven by a mixture of data obtained from conventional data stores as well as user input. Third, a KBS can operate as an embedded system which is transparent to both other IS products and to most personnel within the organisation (Spang Robinson report on the INCOM system at AMP, 1991).

Other examples embedded systems are real-time systems such as X-JOB (Horsfall, 1990), a KBS used by the Bureau of Meteorology (Australia) to increase the efficiency of using weather forecasting software, or the Operator

Guidance System (OGS), a real-time system within the Rod and Bar Products Division of BHP (Newcastle) (Lock Lee et al., 1990). Accordingly, there are three major issues that this research project will endeavour to resolve in this area, namely: (a) the attributes of the problems addressed; (b) the level of decision support provided; and (c) the type of delivery of problem solving expertise or knowledge.

Components of the KBS

As indicated in the following section, KBS technology has a significant history of both theoretical and more applied research. The practical emphasis of this research has manifested itself both in hardware and software. In the hardware area, the research has produced specialised hardware for KBS's, such as LISP machines and other specialised work stations, for example, 68000 based work stations such as Sun, Apollo, Perq, and Tektronix 4404. These work stations have been specially designed to run KBS software with high levels of efficiency. Therefore, an interesting question to resolve for commercially developed KBS's is whether or these systems require specialised hardware, or whether are they can run on more general purpose computers. Furthermore, is there a trend either towards using more specialised hardware, or away from using specialised hardware with respect to these systems?

With respect to KBS software, the applied research had two major outcomes. The first was the development of a large and diverse range of commercially available KBS generators - or as they are more commonly known, expert system shells. Most of these ES/KBS shells provide a complete development environment for the creation of knowledge based systems. Some of the more sophisticated shells provide additional features that allow the development of more complex integrated systems, that is, KBS that have the ability to pass data to, or receive data from other conventional information systems. Appendix 1 provides a representative, but by no means exhaustive, list of those ES/KBS shells that are commercially available at the present time. Research efforts in the KBS software area also gave rise to what are termed formal AI languages in which KBS's/ES's can be coded, for example, PROLOG and LISP. Accordingly, for commercial KBS's an interesting question to resolve would be which of the above options are used more often. That is, are the majority of knowledge based systems within commercial organisations developed and deployed using expert system shell software? Alternatively, are most systems coded using one or the other of the AI languages mentioned above? Or is it a situation where systems are developed using more conventional 3GL programming languages (such as COBOL, or C) or 4GL languages (such as MANTIS or NATURAL)? Or is it a situation where elements of all three development environments referred to are used?

Finally, the KBS is a delivered piece of software, which will consist of modules and other facilities such as user interfaces and so on. In terms of the major components within a knowledge based system, Turban (1992) states [pp. 81-85] that the following components would normally be generic to different KBS's, irrespective of the type of problem addressed or the development environment (expert system shell, AI or conventional programming language or any combination of these), namely:

- **The Knowledge Base** - which stores all the information that is necessary for understanding, formulating, and solving the problem such as: (1) facts (often known as the fact base), that give insight to the problem situation and structure to the theory of the problem area; and (2) special heuristics, or rules that direct the use of knowledge to solve problems in a particular domain.

- **The Blackboard (Workplace or Working Memory).** This is an area of working memory set aside to describe the current problem, as specified by the input data; it is also used for recording intermediate hypotheses and decisions. Three decision types are recorded on the blackboard: (1) *plan* - how to attack the problem; (2) *agenda* - potential actions awaiting execution; and (3) *solution* - candidate hypotheses and alternative courses of action generated thus far.

- **The Inference Engine.** This component is the software that allows some form of reasoning strategy about information in the knowledge base and in the blackboard, in order to formulate conclusions. Generic elements of the inference engine are: (a) an *interpreter* (a rule interpreter in most systems) which executes an agenda (a prioritised list of rules whose patterns satisfy facts in working memory); (b) a *scheduler*, which maintains control over the agenda; and (c) a *consistency enforcer*, which attempts to maintain a consistent representation of the emerging solution.

- **User Interface.** In interactive systems, the user interface is the point of contact between KBS and user, or in embedded systems it is the point of contact between KBS and application program. With respect to the former, the interface can depend on the implementation of KBS, with the user interface ranging from a simple text-oriented display, to a sophisticated high-resolution bit-mapped display.

- **Explanation Sub-system (Justifier).** The ability to trace responsibility for conclusions to their sources is crucial both in the transfer of expertise and in problem solving. The explanation sub-system can trace such responsibility and explain the KBS behaviour by interactively answering questions such as the following: why was a certain question asked by the KBS, or how was a certain conclusion reached by the KBS.

Are these components evident in the production systems constructed by commercial organisations? Alternatively, are there additional components, or is it a situation where some of the components shown above, or are they combined rather than being separate modules within these commercial systems? Which of the above components are perceived to be more important in the eyes of the project managers developing KBS's?

KBS Technology - A Research Emphasis

The area of artificial intelligence, expert systems and knowledge based systems has been one with a long history of interest to researchers, principally within academic and research institutions. For example, Giarratano & Riley (1989), offer the following table which provides a review of the more significant developments or research efforts which have led to important contributions in this field of intellectual endeavour [p. 12]:

Table 4. Important Events in the History of Expert Systems[2]

Year	Events
1943	Post production rules; McCulloch and Pitts Neuron Model (McCulloch and Pitts, 1943)
1954	Markov Algorithm for controlling rule execution (Markov, 1954)

[2] *This list is not considered either an exhaustive list or an up-to-date compendium, but merely an extensive and quite representative sample of the research efforts that have occurred.*

Year	Events
1956	Dartmouth Conference; Logic theorist; Heuristic Search; "AI" term coined
1957	Perceptron invented by Rosenblatt; GPS (General Problem Solver) started (Newell, Shaw and Simon, 1960)
1958	LISP AI language (McCarthy, 1960)
1962	Rosenblatt's Principles of Neurodynamics on perceptrons (Rosenblatt, 1961)
1965	Resolution of method of automatic theorem proving (Robinson, 1965) Fuzzy logic for reasoning about fuzzy objects (Zadeh, 1965) Work begun on DENDRAL, the first expert system (Buchanan et al., 1969)
1968	Semantic nets, associative memory model (Quillian, 1968)
1969	MACSYMA math expert system (Moses, 1975)
1970	Work begins on PROLOG (Colmerauer et al., 1973; Passero and Kanui, 1972; Roussel, 1975)
1971	HEARSAY I for speech recognition Human Problem Solving popularises rules (Newell and Simon, 1972)
1973	MYCIN expert system for medical diagnoses (Shortliffe, 1976) leading to GUIDON, intelligent tutoring (Clancey, 1987) and TEIRESIAS, explanation facility concept (Davis, 1979; Davis and King, 1977) and EMYCIN, first shell (Van Melle, 1979) HEARSAY II, blackboard model of multiple co-operating experts
1975	Frames, knowledge representation (Minsky, 1975)

Year	Events
1976	AM (Artificial Mathematician) creative discovery of math concepts (Lenat, 1976) Dempster-Schafer Theory of Evidence for reasoning under uncertainty Work begun on PROSPECTOR expert system for mineral exploration (Duda, Hart, et al., 1979)
1977	OPS expert system shell (Forgy and McDermott, 1977), used in R1/XCON
1978	Work started on XCON/R1 (McDermott, 1980) to configure DEC computer systems Meta-DENDRAL, metarules and rule induction (Buchanan and Feigenbaum, 1978)
1979	Rete algorithm for fast pattern matching (Forgy, 1982) Commercialisation of AI begins Inference Corp. formed (releases ART expert system tool in 1985)
1980	Symbolics, LMI founded to manufacture LISP machines
1982	SMP math expert system; Hopfield Neural Net; Japanese Fifth Generation Project to develop intelligent computers
1983	KEE expert system tool (IntelliCorp)
1985	CLIPS expert system tool (NASA)

(Source: Giarratano & Riley, 1989, p. 12)

One inference that can be drawn from the above table is that the research effort connected with artificial intelligence (AI), expert systems (ES) and knowledge based systems (KBS) appears to have a very strong practical emphasis. That is the research has provided tools and techniques - with which to develop actual KBS's, rather than concentrating on more abstract things such as guidelines for design tools and frameworks - which merely provide indications as to how KBS's should work or be developed. It would appear that the research has provided a sound foundation for the transfer from theory into practice. However, as evidenced in the following section, the use of KBS technology by commercial organisations has been very patchy. Consequently, an interesting issue to resolve would be the reliance or the awareness of commercial KBS developers on the research that has been, or is currently being conducted in this area. For instance, are the sorts of expert systems or knowledge based systems that are being developed by commercial organisations using some of the languages, knowledge representation formalisms (such as frames), or expert system shell software listed in the above table? Alternatively, has little been picked up and used by commercial organisations from the efforts of researchers in this field?

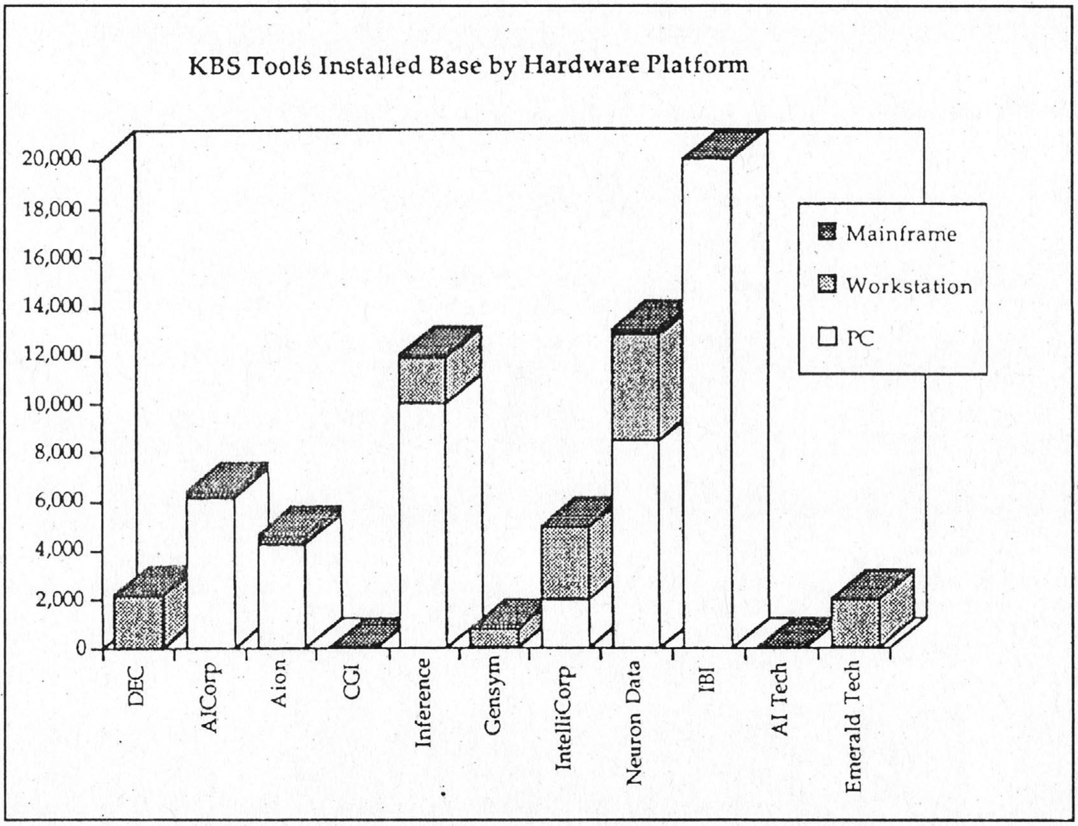
Commercial Interest in KBS Technology

As already alluded to earlier in this Chapter, the literature has shown a growing interest during the 1980's and early 1990's by commercial organisations in using KBS, as distinct from academic and research institutions which have different interests and objectives with respect to KBS technology. For example, in "The Spang Robinson Report on Artificial Intelligence", (Wolters, July 1991) reported, [p.4]:

> *"Knowledge Based Systems (KBS) companies, no longer at the bleeding edge of technology, have traded in evangelism for pragmatic mainstream application delivery. KBS technology is regarded as mainstream technology and is among the top three technologies of interest to MIS managers according to a recent ComputerWorld survey. .... The industry's move to C and C++ has provided the essential integration and practical deployment vehicle for knowledge systems that industry needed."*

Furthermore, although "The Spang Robinson Report on Artificial Intelligence" (1991) states that the market for KBS technology is still small, $250 million out of a total $87 billion market for software and services, there is evidence that the market will continue to grow at around 20 to 25 per cent. per annum. In terms of actual products sold to date, the July 1991 Spang Robinson report provides the following graph show below [p. 7]:

Figure 1. KBS Tools Installed Worldwide, 1991

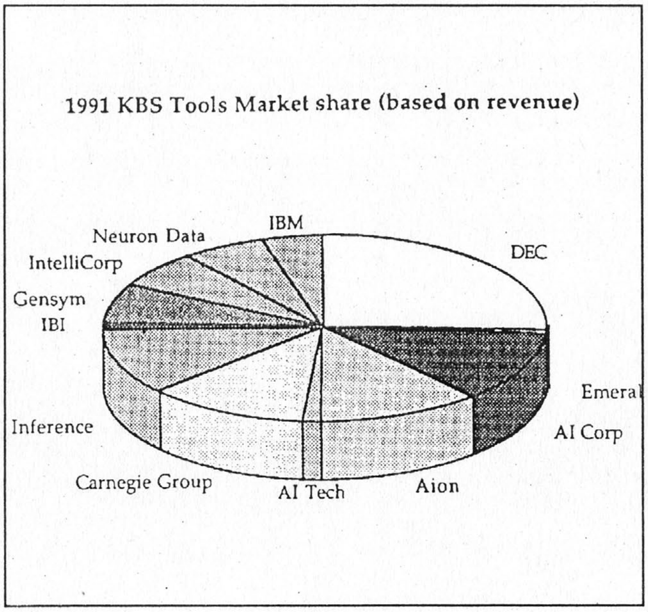


(Source: July 1991 Spang Robinson Report on Artificial Intelligence, p. 7)

In fact, Baskin and Michalski (1989) have argued that even more growth and interest in applying this technology would occur if certain limitations with the technology were solved, for example, the ability to use more than one form of knowledge representation, the lack of a learning capability, the use of only one type of inference procedure, the use of only a single control strategy, and the inability to deal with data or situations that are time-dependent [p. 113]. Although this would appear to be an extremely interesting research topic, it is not a goal of this research project to consider the impact of the technical aspects of this technology with respect to the development of commercial KBS's.

The July, 1991 Spang Robinson Report on Artificial Intelligence also documents that a number of well known (and not so well known) organisations are actively involved in developing and selling commercial KBS tools/shells as indicated in the graph below - which also provides information on the share of the KBS market that each organisation has within the United State of America [p. 7]:

Figure 2. Market Share of Major KBS Tools, 1991



(Source: July 1991 Spang Robinson Report on Artificial Intelligence, p. 7)

In addition, there appears to be an indication of productivity improvements (that appear coupled with advances in KBS development tools) that commercial organisations could take advantage of, as evidenced from the table shown below (Michie, 1991, p. 145):

Table 5. Comparison of Four KBS Developments

Application	Knowledge Domain	No. of rules	Development time (man-years)	Maintenance (man-years per year)	Inductive Tools (KA)
MYCIN	medical diagnosis	400	100	N/A	N/A
XCON	VAX computer configuration	8,000	180	30	N/A
GASOIL	hydrocarbon separation system configuration	2,800	1	0.1	ExpertEase and Extran 7
BMT	configuration of fire protection equipment in buildings	> 30,000	9	2.0	1st Class and RuleMaster

(Source: Michie, 1991, p. 145)

However, all does not seem to indicate that commercial organisations are rushing into KBS developments. For example, Konsynski (1988) reports that very few active ES projects, have led to significant organisational impact (approximately 1,500 projects were investigated). Furthermore, while texts and research articles refer frequently to the MYCIN system, in fact this KBS never got past a research/technology demonstration system. The researchers in the end found that development of the final production version would be too costly. However, in terms of systems alluded to in journal articles and textbooks in this area, the MYCIN is discussed within the context of being a paradigm of KBS development, although this particular system never became a production system. Jackson (1990) [p. 54] documents some of the reasons why MYCIN never became a production system, namely: (a) even after many years of development the knowledge base was still considered incomplete; (b) running the system, at the time, would have required more computing power than was available within most hospitals; and (c) the user interface was primitive, requiring too much typing by doctors trying to use the system. Are these still relevant factors that mitigate against this technology becoming more accepted within the commercial sphere?

Another system often quoted in the literature is X/CON, which is seen as one of the KBS success stories, but is now considered to be at the limits of its maintenance capacity. Bachant and Soloway (1989) report on the significant re-engineering efforts associated with this system in order for it to remain viable within the order configuration processes at Digital Equipment Corporation. Therefore, are the following factors important in why KBS's are not a more commonplace product in commercial organisations, that is these systems are: (a) hard to develop in the sense of finally deploying a production system; and (b) hard to keep in production because of significant maintenance difficulties. It seems, at least in the United States, that there is evidence to suggest that the development of commercial KBS's with real productivity advantages to organisations, that have extended production lives and low operating costs have yet to emerge.

Furthermore, there is also a reasonable number articles reporting case study research that refer to the failure of the technology. These reports, which come mainly from IS professionals within commercial organisations, indicate that after a reasonable trial of KBS technology, the technology has failed to live up to its expectations. For instance, Cook (1991) discusses the experience of using KBS technology within American Airlines. Cook's major premise is that ES/KBS technology has been oversold within American Airlines, with only one viable production system developed - Gate Manager. From the perspective of the major goal of this research, the article is an interesting one. First, there is no mention by Cook that the technology has failed because of any inherent deficiencies within knowledge based software. Moreover, Cook indicates that failure is due to poor application of KBS technology, and argues that KBS projects should only be initiated after they have leaped "over an appropriate return on investment hurdle" [p.8]. Cook's arguments appear to support Hopple's law of the hammer [Hopple, 1986] - give a child a hammer and he will use it on everything encountered.

A recent Australian survey conducted by RMIT (Takac & Lerner, 1990) seems to confirm this inability to develop commercial production KBS's. Cranswick (1990), who discusses the RMIT report, states [p.21]:

> *"Among those organisations that have become involved in expert systems work, it is generally true to say that the trend has been for their first organisational experimentation."*

One of the goals of this research project will be to verify or deny this 1990 assessment of the current "state of play" within Australia. That is, are the majority of commercial organisations still at the experimentation stage? Alternatively, do commercial organisations treat this technology as merely another way to develop information systems? Or is it a situation that rather than being a mainstream IS activity, the technology is still perceived as novel or new? To some extent, the novelty of the technology can be gauged by how the organisation currently treats it. One could consider that the organisation still has an immature attitude to the technology if most KBS development occurs outside of the MIS department, say within a research and development area of the organisation, and that an organisation has a more mature attitude where most KBS development activity is located within the MIS department. Furthermore, is it a situation for most organisations identified as being involved in KBS development that systems development did not complete, in the sense that a final production of version of the KBS was not deployed? Alternatively, in terms of commercial production systems, is the experience different to that reported in the United States?

Furthermore, both Cranswick and Cook provide another insight into one of the underlying thrusts of this research project. It would appear that failed KBS developments are not due to defects in the technology, but rather due to an inappropriate approach to systems development. In terms of realising production KBS's should the focus be more on acquiring or building and then utilising some suitable form of development methodology, rather than concentrating purely on the technical aspects invariably associated with building these systems.

For example, would the efforts devoted to developing MYCIN have had greater success if a formal development methodology had been applied to address two of MYCIN's shortcomings. That is, with the use of a development methodology, a more disciplined and rigorous approach to knowledge acquisition was undertaken thereby ensuring that all relevant knowledge concerning infectious blood diseases had been elicited. Furthermore, that the methodology had directed system builders to engage in closer and more regular contact with users, thereby ensuring greater acceptance of the user interface. In another case study discussed above, namely, American Airlines, would the use of a methodology have ensured that over time the organisation experienced greater success with the application of this technology?. The failure in this organisation would appear to be a system development problem again. That is, the article indicates that if a formal set of processes or procedures regarding selection of the most appropriate problem domains was in place (such as one would expect to be incorporated into a development methodology), then this would minimise the chance of aborted or unsuccessful system developments. Accordingly, this discussion leads to another interesting issue in connection with commercial organisations developing KBS's. That is, for these organisations is success more associated with the use of a formal system development methodology than in situations where a more informal development approach was adopted?

Development of KBS's

Concomitant with this increased interest in using KBS technology referred to above, there is been increased interest in how knowledge based systems should be developed. To a large extent in the early days of knowledge based systems, development of these systems occurred in research or academic institutions, which had completely different objectives to those being developed for commercial reasons. For example, Guida and Tasso (1989), in the introduction to their book, state [p. 21]:

> *"Expert system technology is still largely relying today on empirical methods and is not supported by sound and general methodologies. It is therefore more like handicraft than engineering, and it lacks several of the desirable features of an industrial process (reliability, repeatability, work-sharing, cost estimableness, quality assurance, etc.).*
> *The transition of an expert system technology from handicraft to engineering is widely recognised today as a much needed step*

aimed at overcoming some of the major bottlenecks which presently hamper the large-scale application of expert systems in business and industry."

As indicated in the above quote, given the increased use of KBS, it would appear eminently sensible that development of these systems move away from an ad hoc, one-off approach towards something that is more structured. The research literature has evidence of continued effort in this area with a variety of development approaches offered, namely: (a) traditional systems development life cycle approach (Burns and Dennis, 1985); (b) an iterative development approach (Lock Lee, 1986; Harmon and King, 1985); (c) a contingency development approach (Burns and Dennis, 1985); or (d) some other form of development methodology (Messier and Hanson, 1983; Hayes-Roth, et al., 1983).

In fact using a prescriptive proven methodology (or methodologies) should provide, amongst other things, the following advantages for the project manager and the KBS development team as a whole:

- Inform project team managers and members of the types of tasks and activities required, together with a schedule that indicates when particular tasks are to be performed.
- Assist in the overall management of the system development by indicating to the project manager key checkpoints and deliverables required as system development progresses.
- Provide a framework within which an assessment can be made of whether the KBS development has been a success or failure - either for the project as a whole, or within a particular phase of the development. This assessment process should then feed back into the development methodology in order to improve it,

that is developers can not only learn from their own mistakes but also be preventing from making the same mistakes as others.

- Allow for a certain amount of rigour to be introduced into system development tasks and activities.

With respect to the last point, to understand what is meant by rigour, the following example is used. The evaluation activity in KBS development is considered very important (Gashnig, 1983; Perry, 1983; and Liebowitz, 1986). Accordingly, a more rigorous evaluation procedure would consider all of the following issues (which possibly would not be addressed by adopting a more ad hoc approach):

- What level of expertise should be incorporated into a KBS for its target task environment? It is important for users to know the limitations of the KBS.
- Who evaluates the KBS? When does the evaluation take place, say only at the end of development or continuously throughout the development process?
- How do you test a KBS that incorporates fuzzy logic and uncertainty factors in its inference process?
- What happens when a KBS is being developed using knowledge on which experts within the field disagree?

Accordingly, when looking at systems development approaches for KBS's, the following issues would seem relevant. For a commercial organisation that uses a formal systems development approach, this fact would indicate that organisation has passed the experimentation stage of using KBS technology. On the other hand, the absence of a formal systems development approach would be indicative that the organisation is still experimenting with the technology.

Other interesting questions also arise when considering the development approach adopted by commercial organisations. For instance does the level of commitment by an organisation (say at the corporate level, from the MIS department or even from user areas) have an influence on the mode of KBS development. Would those organisations that show high levels of commitment follow a more formal development approach, whereas those with low levels of commitment follow a more ad hoc approach.

Comparing KBS and Conventional IS Development

What sort of reactions, adjustments and changes have organisations adopted with the introduction and development of knowledge based systems? For this aspect of the development of commercial KBS's, a comparison could be made between the development of decision support systems and knowledge based systems. Given that both systems have a focus on decision support, do commercial organisations adopt the same structures to the development of DSS's as they do to the development of KBS's? For example, what area of the organisation championed the introduction of KBS technology? What area of the organisation, either at the start of using the technology or currently, develops knowledge based systems - the MIS department or some other area? What sort of management control structures and reporting procedures are associated with the development of KBS's?

With respect to the development of commercial KBS's, it is interesting to compare the development of other sorts of information technology, such as decision support systems. To a some degree DSS's represent an outgrowth of conventional IS development, either in the sense of enhancing functionality of an MIS - "upgrading" it to become a DSS, or using conventional systems development and software tools and techniques to construct a DSS from scratch. On the other hand, KBS's have a different heritage - emerging from the research performed primarily in the area of artificial intelligence. This difference becomes apparent when the two technologies are compared with respect to how they assist and support decision making, as indicated earlier in this Chapter (see table on page 5).

To some extent there should be similarities in the development approach adopted, the development tasks and activities undertaken, and the type of documentation generated. This is because at the highest level, the ultimate objective of KBS and DSS development is the same, namely to deliver a piece (or a number of pieces) of software to run on a computer. Accordingly, in comparing development procedures a number of interesting issues arise. For example, does the level of integration between a KBS and conventional IS have an effect on whether activities are similar or not? That is, KBS that are essentially stand alone systems have less similarities in development approach, development tasks and activities and development documentation than do KBS's which have a high level of integration.

Another interesting issue would be to determine where the similarities occur. For example, are reporting and management structures more equivalent (for example, KBS and IS development report to steering committee) than development milestones or documentation. Alternatively, are similarities to be found more in the tasks and activities of the system development, such as the need for both KBS's and DSS's to train and educate users in the effective utilisation of the new system.

On the other hand, has the different heritage for KBS technology, coupled with the different functionality of the technology when compared to DSS's (as shown in previous tables), led to different mechanisms for the development of these systems in commercial organisations? As indicated in the following chapter of this research report, there does not appear to be a commonly agreed upon methodology for the development of KBS's. This is in contrast to conventional IS developments. Although there are a number of different development methodologies many of these are variations derived from the dominant paradigm - structured systems development, and the systems development life cycle paradigm. A new system development paradigm is also emerging, namely, object oriented design and development. It may be this new paradigm will produce a development methodology that encompasses both conventional IS as well as KBS development.

Other differences can also be discerned on a more detailed examination of the development process. For example, different people and development activities are required to successfully develop and deploy a production KBS, as compared with conventional IS development. Some of these differences are shown in the list below:

- knowledge engineer - different skills required;
- identification and selection of the expert or knowledge source;
- knowledge acquisition;
- knowledge representation;
- evaluation and validation procedures of the knowledge base;
- documentation, analysis and design - there is no equivalent to a dataflow diagram in a KBS development;
- relies more on prototyping than conventional systems; and
- may not follow a formal system development methodology, but rather pursue an incremental development approach.

An interesting issue to resolve would be whether these differences are purely superficial, or whether they are more fundamental and so affect the whole structure of the development process.

Importance of Conducting the Research

The importance of this research project is linked predominantly to the fact that knowledge based systems represent a form of decision support. Decision making is arguably one of the most fundamental activities driving the engine of organisational activity. It occurs and impacts upon every level within an organisation - from very low clerical positions where personnel would be making predominantly structured operational decisions, to the highest senior executive making more complex and unstructured decisions (Kirs et al., 1989). Decision making is also probably the most frequent organisational activity undertaken by human beings during a normal working day, irrespective of whether the organisation is driven by a profit motive ("private sector" organisations), or one driven by some other goal or objective ("public sector" organisations).

However, in terms of providing effective decision support, a problem arises due to the multi-dimensionality of the decision making activity. One of these dimensions is the quantum of effect a decision made can have on an organisation. For example, when considered individually, decisions can have very little impact on the organisation as a whole, such as when a manager decides to contact a small customer regarding the payment of an overdue account. Alternatively, decisions may have a very significant impact on an organisation's operations and viability, for example, an Australian insurance company decided to adopt a more aggressive marketing strategy with respect to its insurance products in order to become the largest insurance company in Australia - with a rather dramatic and severe impact on its ability to operate as a going concern (ABC Four Corners program, 1993). Obviously, the above two examples are at opposite ends of a continuum with other decisions made by employees or groups of employees having a varying level of influence and impact. Furthermore, the collective impact of a large number of small decisions, although in themselves may not be significant, when combined together, may have the same sort of impact alluded to in the second example discussed above, that is lead to significant consequences (for the better or worse) for the organisation, either as a whole or for a significant part of it.

Although there can be little doubt that decision making is a multi-dimensional activity, there are some reservations about the whether these separate dimensions are orthogonal or not. (It should be noted that the discussion below does not attempt to provide an exhaustive list of all the dimensions to decision making - as this is not an aim or objective of this research project.) For example, one dimension of decision making is time. That is, decisions can be made very quickly, say taking only a matter of seconds or minutes, or decisions may take a long time to make, say of the order of many years. Looking at another dimension to this activity - decisions can be made by humans following a formal approach ("going by the book"), or by a more informal and intuitive approach ("by the seat of their pants"). Decisions can be made by individuals with very little advice or assistance from other personnel either within or outside of the organisation. As an illustration, where decisions have an impact across departmental boundaries within an organisation, if they are to be successfully implemented, then these decisions may require the co-operation, input and agreement of personnel located within the other departments affected.

The lack of orthogonality or independence of these dimensions can be seen from the following scenario. A senior manager of an organisation must make a time critical decision which has a broad organisational impact. The manager considers that time is the most important dimension - that is a solution or decision is required almost immediately. In this situation the manager considers that he or she does not have time to consult either with other personnel (who may not be readily available), or to follow a formal (but rather time consuming) decision making procedure. It is evident from the above that the importance of this activity to organisations continued health, and the multi-dimensionality of the process makes this area a rich source of potential research projects. The focus discussed in this project report attempts to provide further light on but one small part of this area of interest.

In simplistic terms, the focus of this research is to look at another, and again possibly not orthogonal, dimension to the activity of decision making. In today's organisations with increasing complexity and pressure to make decisions at a faster pace, many decisions cannot be made effectively without substantial support from some form of information systems/information technology (Davis, 1988). In this situation, human beings may either lack appropriate knowledge, or may lack access to pertinent information and so cannot confidently make correct decisions. In this state of uncertainty most human beings would look for support or assistance during the decision making process. One form of assistance or support can be derived from applying information technology to assist them in some fashion - either for the whole of the decision making process, or for a minor part of it.

For example, this support may come in the form of an easy to use and flexible data enquiry and retrieval system attached to a specially designed database (for example, an executive information system - Watson, et al., 1991), or the provision of a modelling capability in which the decision maker can both structure and simulate solutions to a complex problem (for example, a decision support system - Brightman, et al., 1988), or enhancing the intellectual capacity of the decision maker (for example, a knowledge based system - Caddy and Stephens, 1992). The focus within this research project will be on how commercial organisations have used the last example of assistance, namely the provision of knowledge based systems technology. In more specific terms, the goal of this research project is to better understand the processes that commercial organisations adopt when trying to develop this particular type of decision support mechanism.

There are two major reasons why this research area and the research project are important. First, there is the obvious importance of decision making to the organisation. As alluded to above decision making is probably the most frequent activity engaged in by an organisation, and probably the most common activity undertaken across functional areas or managerial layers within an organisation. Therefore in applying computer based decision support it is important to understand how to most effectively interface this support with the actual conduct of the decision making activity. Second, the fact that decision making is multi-dimensional means that effectively interfacing computer based decision support is not a simple matter but rather very complex. Adoption of an ad hoc approach to completing this activity is more than likely to be unsuccessful, and the case study literature seems to support this contention. On the other hand what sort of optimum approach should be taken is uncertain. It is hoped that this research project will remove some of that uncertainty, and that a better understanding hopefully will lead to a more mature approach being adopted by organisations in the development and delivery of KBS technology - resulting in more successful KBS development projects.

Schematic of the Project Report

The structure, in terms of the major chapter headings, of this project report is as follows:

- **Chapter 1 - Introduction**: The objective of this Chapter was to provide an outline and discussion of relevant research issues with respect to the development and utilisation of the KBS technology by commercial organisations. The importance of the research is also considered in the final section of this chapter.

- **Chapter 2 - Literature Review:** This Chapter reviews the research literature in terms of the following points:

  (a) whether the literature indicates that organisations have applied KBS technology to similar problems;
  (b) whether the literature indicates that organisations have adopted similar KBS components in the delivery of KBS solutions to these problems;
  (c) whether the literature provides an indication that similar development approaches have been used by organisations to construct KBS's;
  (d) comparisons of development issues between conventional IS development and KBS development;
  (e) an investigation of the research literature on system development methodologies and the formulation of an assessment framework to determine what system development products should be considered development methodologies; and
  (f) a validation of the system development methodology assessment framework discussed (e) above.

- **Chapter 3 - Research Methodology and Research Questions:** This Chapter discusses the specific research questions to be addressed and the research methods that were adopted to find answers to these questions. The research methodology is also outlined in this chapter, together with the reasons why the case study approach was considered to provide the most fruitful method to pursue. The final section includes a consideration of threats to the validity of the research, both construct, internal and external.

- **Chapter 4 - Case Study Results and Analysis:** This chapter is concerned with the results of the research effort. As indicated in the previous chapter, the research followed a case study approach which involved seven target organisations (organisations A, B, C, D, E, F, and G), who were previously identified as having developed at least one production knowledge based system (systems A1, B1, C1, D1, E1, F1, and G1). The case study consisted of interviews and questionnaires concerning the development of the KBS system, both with regard to the features of the knowledge based system, as well as various issues concerning the actual system development. The analysis of the case study data is concerned both with the data collected from the questionnaires as well as a semantic analysis of the interview transcripts.

- **Chapter 5 - Conclusion and Future Research Issues:** This final chapter provides an overall analysis of the research findings, an assessment of whether the research issues were answered, and a discussion of any perceived limitations within the research. This Chapter also looks at future research issues and projects that could be forthcoming as a result of having undertaken this research.

Chapter Summary

The prime focus of this research project is to investigate how commercial organisations are applying KBS technology to assist in decision support activities. As indicated this area represents a wide field of interest covering many issues and questions. Out of all the questions and issues that could have been investigated, the following areas of interest have been selected:

- **Problems solved with KBS technology.** Some of the more substantial research questions to be considered under this heading are: (a) whether or not similar sorts of problems are addressed by different organisations; (b) is KBS technology being applied to important fundamental problems facing an organisation, or are these problems quite peripheral to core activities; and (c) what procedures are used to identify problems that are suited to a KBS solution.

- **Use of KBS technology.** With respect to this heading, there are a number of separate issues to consider. First it would be interesting to determine how and who "introduced" the technology into the organisation, namely MIS, or IT, research and development, or a user area. Who introduces the technology is considered to have influence on what problems are solved using KBS technology. For example, a user area championing the technology will select problems within the limited domain of its expertise rather than address more fundamental and core problems.

  Another issue to consider is the internal structure and make up of the KBS solution. Are there similarities here across different

KBS developments. For example, are most commercial KBS's built with knowledge represented as production rules, and forward or backward chaining for the inferencing strategy? Another interesting issue would be to determine the development environment. Are most commercial KBS's delivered using knowledge based system shells software? Alternatively are programming languages, either conventional or AI languages, typically used to build these systems?

- **Development of KBS's.** From an organisation point of view, issues considered worthy of investigation under this heading are: what area undertakes KBS development - MIS or IT or some other area, and what level of support or interest do various parts of the organisation have in the KBS development. It should be interesting to determine the relationship between who introduced the technology and who undertook the KBS development. It should also be interesting to determine the level of organisational support against the delivery of a final product. Another issue will be to determine the degree of similarity between activities undertaken, and the personnel who participate, in a KBS development as compared to a conventional IS development.

- **KBS development methodologies.** This heading is concerned with whether these products exist, what features do they possess, in particular how similar are they to conventional IS development methodologies, and how many times the product has been used. The study will try to determine what advantage or impact a KBS development methodology has had, that is do organisations with a KBS development methodology develop more KBS's than organisations without a methodology?

> Another interesting issue will be to determine the reasons why organisations did not create, acquire and then use a KBS development methodology. It is a situation that an on-going program of KBS development was not foreseen and therefore the effort to build, acquire and use a KBS development methodology was considered a waste of scarce and valuable resources?

Other issues covered in this chapter included the change in focus, for artificial intelligence in general, and knowledge based systems in particular, that has occurred in the last decade. The trend has been a movement from basically an area with academic and research interests to one with a growing interest by commercial organisations. Evidence of this is provided by the development and marketing of commercial KBS development environments (expert system shell software). Furthermore, the case study research literature reports a growing number of organisations, spread across many different industry sectors are undertaking KBS development.

However, linked to this increased commercial interest is an indication that not all KBS developments are successful. Therefore one aim of this research project is to identify factors that are conducive to successful KBS development (and so should be followed by organisations undertaking this activity) as well as those factors which lead to developments that are constantly questioned and extended (and so should be avoided). In this regard it is considered that the research has worth because, decision making and the provision of decision support (via KBS technology) are important processes tackled by commercial organisations. Unfortunately, clear directions on how to effectively apply the decision support is often confounded by the multi-dimensionality of the decision making process. That is, the solution is not easy and therefore requires research to determine more effective processes and procedures than those being currently used.

Chapter 2: Literature Review

Chapter Outline

How Do Commercial Organisations Develop KBS's?

How Are Knowledge Based Systems Developed?

Case Study Research Concerning KBS Development

A Better Understanding of the Term Methodology

A Framework to Assess Systems Development Methodologies

Validation of the SDM Assessment Framework

How Do Commercial Organisations Develop KBS's?

This question inevitably leads to other questions. For example, by combining the type of problems that a commercial organisation may try to resolve using KBS technology and the type of KBS technology applied to that problem by the organisation, leads to the following two way contingency table:

Table 6. Application of KBS Technology to Commercial Problems

Application of technology	Type of problems solved	
	Tackle similar problems	Tackle different problems
Used KBS technology similarly	(1)	(3)
Used KBS technology differently	(2)	(4)

Furthermore, it would also be interesting to add a third and orthogonal dimension to this contingency table. That is, rather than looking at just a single organisation, do different organisations tackle similar types of problems, and do these organisations apply the same sort of KBS technology to the resolution of these problems. For example, do different organisations apply similar knowledge representation techniques, similar inferencing strategies within the inference engine, similar knowledge acquisition techniques and so on. This would lead to the following two way contingency tables:

Table 7. Types of KBS Problems - Across Commercial Organisations

Organisations	Type of problems solved	
	Tackle similar problems	Tackle different problems
Within a single organisation	(5)	(7)
Across different organisations	(6)	(8)

Table 8. Use of KBS Technology - Across Commercial Organisations

Organisations	Use of KBS technology	
	Use KBS technology similarly	Use KBS technology differently
Within a single organisation	(9)	(11)
Across different organisations	(10)	(12)

After surveying a number of production KBS's both within a single organisation, as well as across a number of organisations, what quadrants would contain the highest frequencies? Would these "high frequency" cells be (7) and (11) for a single organisation, and (8) and (12) across different organisations? Some of research would support this supposition. First, the literature indicates that problems in which KBS solutions have been applied are more often different than similar. It would appear that the problems addressed/resolved using KBS technology both across organisations and within an organisation have little in common other than they are knowledge intensive. For example, within a single organisation, Irgon et al. (1990) reviewed five production KBS's developed within Bellcore, an AT&T company, namely:

(a) FAST - a system for the performance tuning of a large software system;

(b) Maven - used to assign Bellcore's Common Language equipment codes;

(c) GADS - a diagnostic shell for constructing hot-line support applications;

(d) Bellcore's OS-Assist system - a support tool for managers in directory assistance offices; and

(e) Planner - a planning tool for determining call set up signalling.

As the above demonstrates, although concerned with activities within a telecommunications carrier, the problems being solved are quite diverse. In terms of across different organisations, Jamieson (1986) reviewed development of expert systems within the audit industry. The types of systems found are detailed in Appendix 2, and indicate that even within the same industry, the problems being solved by KBS technology can vary dramatically. Brown and Phillips (1990) have performed a similar survey more recently with the same results. Two examples from the Brown and Phillips (1990) are Cooper & Lybrand's QSHELL, and Peat Marwick's Loan Probe. QSHELL covers the entire audit process, including planning, execution, and automatic generation of work papers and other audit reports. QSHELL uses an intelligent questionnaire to guide the auditor through the audit assignment. On the other hand, KPMG Peat Marwick's Loan Probe is an 8,000 production rule system that analyses bank loans and determines the level of loan reserves needed.

Furthermore, the research literature also provides ample case study evidence of a diversity of problems tackled using KBS technology. Not only is this the case overseas, where problems ranging from assistance in configuring DEC computer hardware (XCON - Barker and O'Connor, 1989 plus the Appendix to this article by Bachant and Soloway, 1989; Leonard-Barton, 1987), manag bush fires (Klein, et al., 1989), tax accrual accounting (ExperTAX[tm] - Shpilberg, et al., 1986), to assistance in production planning at a large motor vehicle manufacturer (DLMS - Johnson et al. 1989). As indicated in the previous chapter, the same experience has occurred within Australia (Lock Lee, 1990; Beinat and Tsui, 1992; and Horsfall, 1990). Finally, Hochman and Pearson (1991) report on the X-Breed expert system used to assist professional advisers to the beef cattle industry in the task of recommending to producers appropriate cross-breeding strategies.

Accordingly, from the above discussion, there appears to be a substantial body of evidence supporting the assertion that individual organisations or a number of different organisations use KBS technology to solve different problems. This indicates at least two differences with conventional information systems. That is, there is commonality within conventional IS across organisations in terms of problems addressed. For example, most organisation would have some form of accounting information system (covering debtors, creditors, stock control, cash movements), all with a similar functional goal. Furthermore, if two accounts receivable systems were examined, there would be little functional difference. Flowing from this is an on-going program of new system development and re-engineering of existing systems within organisations. However, looking at the case study research, it appears that KBS technology is applied on a more ad hoc basis, such as at Bellcore and the KBS's referred to in the paragraph above (although the DuPont case study (Bailey, 1987) discussed below provides a counter argument to some extent), and for this reason there does not appear to be an on-going commitment to a program of introducing KBS technology into organisations.

What about how KBS technology is applied, when the knowledge representation, the inferencing strategy and the user interface are examined? Do most commercial production KBS's represent their knowledge as production rules, use a forward or backward chaining inference strategy, and have interactive explanation and help facilities built into their user interface? Or is the alternative true in which no commonality can be discerned in these areas either for different KBS's developed in the same organisation, or different KBS's developed by different organisations? With respect to developments within organisations, the case study research is not as clear in this area as it was on the above issues. For example, in some cases indicate diversity such as in Irgon et al. (1990) who state [p.39]:

> *"Using a powerful multiparadigm expert system programming environment enables developers to experiment with various design approaches and to implement an initial prototype rapidly and incrementally, but this requires significant training and startup consideration."*

On the other hand, Bailey (1987) reporting on the experiences within the DuPont Corporation indicates similarity rather than diversity. At DuPont three factors drove similarity, namely: (a) the organisation wanted to introduce KBS technology into all areas of the organisation; (b) the activities of the organisation were diverse in range and dispersed geographically; and (c) the level of literacy within the client user population was considered low in general (users were spreadsheet literate but had little programming knowledge or experience) and practically no KBS experience. Accordingly, introduction of the KBS technology and system development followed an end-user computing model, the preferred development environment was expert system shell software, and the development approach or "method" was similar across different projects.

However, this research does not provide any evidence on how expert system shell software was applied to different problems within different divisions of the corporation. As indicated in Appendix 1, even if development is restricted to creating a KBS using expert system shell software, there is substantial diversity within these products with respect to knowledge representation formalism and inferencing strategies, and other features. Accordingly, an interesting question thus arises from this discussion. That is, within Australian commercial organisations, is the development environment and the type of knowledge representation, inferencing strategy, and other components that make up the KBS determined on a project by project basis? Alternatively, do organisations adopt a common application of the KBS technology across a range of different problems as seems to be the case with DuPont?

The research literature indicates that KBS components are tailored on a number of different factors. One of these factors is the modality of the problem. In some situations, the KBS works like an intelligent on-line encyclopedia. In this scenario, the KBS conducts a dialogue with the user, and using the knowledge contained within its knowledge base and what information the user enters, to reach some conclusion. After delivering this conclusion back to the user, possibly with some form of explanation facility to describe how the conclusions were reached, the KBS would then complete that session and wait to be re-activated by another user session. These systems are static KBS's, where the knowledge base remains unchanged, and new inferencing only occurs as a result of data entered by the user. Examples of these KBS's are the RAIDER system developed by Caddy et al. (1990), and the Automated Air Loading System (AALPS) developed to support U.S. Army and Air Force Personnel responsible for loading cargo aircraft (Klein, et al., 1989). On the other hand a KBS can be designed as a dynamic system in which initial conditions and knowledge base can alter as the session proceeds. An example of this type of KBS is the tactical control and evaluation KBS developed for the U.S. Department of Defense and used in relation to potential air strikes against a hostile naval force (Noble, 1989).

Another example, would be case based reasoning systems developed using products such as CBR Express[tm] created and marketed by Inference Corporation.

The research literature associated connected with the classic systems also points towards diversity in the make up of KBS's, at least across different organisations. Alty's (1989) review of MYCIN, INTERNIST, and PROSPECTOR, indicates diversity in how the KBS technology was constructed as well as how it was applied to problems. However, Alty (1989) does claim that the majority of KBS's are built on a rule-based approach with respect to knowledge representation, and more often than not the inferencing strategy was either forward or backward chaining. For example, MYCIN was built using a dialect of LISP (which ultimately gave birth to EMYCIN, the first expert system shell). The representational scheme employed in MYCIN is production rules, knowledge triples, a context tree to guide the consultation, and an uncertainty handling mechanism based on measurements of belief. Although this knowledge representational formalism has proved to be relatively successful in diagnostic situations, it has a number of limitations and a number of other approaches (incorporating different control or inference mechanisms) have been explored.

On the other hand, INTERNIST's inference strategy was frame-based abduction using hypothesis-and test-cycles, with the domain knowledge represented in descriptive frames of information. From an initial set of observations a set of potential hypotheses is generated, which are checked to see if they account for all observed features. New questions are generated in order to narrow down the set of possible hypotheses and so in the end arrive at a "solution" - only one hypothesis left. Finally PROSPECTOR was built using a dialect of LISP to aid in geological prospecting. Alty (1989) states that the nature of the problem domain led to a different representational scheme, namely a set of production rules in the form of an inference net. Overlaid on this net was a semantic network connecting different objects in the domain in a geologically meaningful way. The inference engine was backward chaining in the main, and the uncertainty mechanism was based on Bayes rule.

A counter argument would be that these systems were developed during the early phases of using this technology. One would expect to find diversity, as these systems were experimenting with new techniques, none of which at that time had assumed a dominant position. Unfortunately, more recent case study research to back up this supposition is hard to find. Often the published research is concerned more with outlining a new technique and applying this technique to an artificial system, rather than discussing the application of the new technique within a real world KBS's. However, there are indications that diversity is more common than similarity even within organisations, and even within a single KBS. For example, Prerau et al. (1990) report on the COMPASS (the Central-Office Maintenance Printout Analysis and Suggestion System) expert system developed by GTE. COMPASS has demonstrated performance comparable to that of domain experts, and employs multiple AI/KBS paradigms within the one system, such as production rules, frame hierarchies, demon mechanisms, object oriented programming, and LISP code, all accessed through the KEE (Knowledge Engineering Environment) expert system shell. COMPASS contains approximately 500 Lisp functions, 400 rules, and 1,000 frames with about 15,000 slots in these frames.

Although Huang (1989) supports Alty's (1989) claim that there has been a predominance of using production rules in KBS's to date, there are disadvantages associated with this knowledge representation paradigm, namely [p. 489]:

> *"Rule-based systems have been widely used in knowledge based intelligent systems for engineering applications and cognitive models for human learning and development. However, most of these systems are brittle in the sense that they are not immune to even minor flaws in their encoded knowledge or slight changes in the environment."*

In order to overcome this brittleness, rule based systems should incorporate additional functionality so that they can adapt or learn from their changing environment through some sort of feedback mechanism. In fact case based reasoning is an attempt, using another form of inferencing other than forward or backward chaining through rule structures, to try and cope with this particular problem that must be addressed by all operational KBS's.

Finally, Mullin (1989) argues that as the nature of expertise varies, so should the way that knowledge engineers approach the task of knowledge acquisition. For example, the techniques that would be applied to domains where knowledge is reasonably well defined or documented (such as in physics or computer science) may be quite different to fields such as the prediction of neurosis versus psychosis from standardised psychological survey data. Noble (1989) refers to recent systems constructed for the U.S. Naval War College in which [p. 473]:

> *" ... the expert system for situational assessment has two main components: a knowledge elicitation system and an assessment system. The knowledge elicitation system is a tool for capturing an expert's understanding of how hostile forces conduct military operations. The assessment system uses the elicited knowledge to infer hostile objectives and plans from a pattern of observables."*

That is, in this system the knowledge acquisition is dynamic and linked into the structure of the expert system. As was indicated above, problems addressed by KBS technology are vastly different and so different knowledge acquisition techniques should normally be applied in these different problem domains. As Mullin (1989) states, the knowledge elicitation would therefore be different, and using the model of development offered by Noble, the inherent functionality of the delivered systems that will service the problem would be different. Accordingly, there appears to be some direct case study research and some indirect evidence to support the contention that across organisations the application of KBS technology is different. Within Australian commercial organisations, is this the case, that is do different Australian organisations develop KBS's with markedly different knowledge representation formalisms, inferencing techniques, user interfaces and other components that make up these KBS's? Or alternatively, is the a degree of similarity between the components used in these different KBS's that may be addressing substantially different problems? This will form an interesting question to address in the case study investigation reported in later chapters of this report.

How Are Knowledge Based Systems Developed?

Following on from the issues discussed above, another question arises: do these contingency factors impact upon the development of KBS's. That is, in keeping with the two-way contingency table technique applied in the discussion above:

Table 9. KBS Develop Approach - Across Commercial Organisations

Organisations	Type of systems development used	
	Within a single organisation	Across different organisations
Similar systems development approach	(1)	(3)
Different systems development approach	(2)	(4)

and, extending this analysis in a similar fashion to that outlined above, with respect to the types of problems addressed using the KBS technology, and the different applications of the KBS technology, do:

Table 10. Type of Problem by KBS Development Approach

Type of systems development used	Tackle similar problems	Tackle different problems
Within a single organisation		
Similar systems development approach	(5)	(7)
Different systems development approach	(6)	(8)
Across different organisations		
Similar systems development approach	(9)	(11)
Different systems development approach	(10)	(12)

Table 11. KBS Technology Used by KBS Development Approach

Type of systems development used	Used similar KBS technology	Used different KBS technology
Within a single organisation		
Similar systems development approach	(13)	(15)
Different systems development approach	(14)	(16)
Across different organisations		
Similar systems development approach	(17)	(19)
Different systems development approach	(18)	(20)

Before, this analysis can be extended further, a fundamental question should be answered first, that is do different development approaches exist for KBS development? That is, has the application of KBS technology components to different knowledge intensive problems, led to different ways of developing KBS's. With respect to construction of KBS's, and in particular knowledge engineering, the research literature indicates that a variety of development strategies are available. Lehner and Adelman (1989) state that KBS development could pursue either of the following development approaches:

(a) ***The classical approach.*** Knowledge engineering is concerned with acquiring knowledge from human experts, and then encoding it into a KBS. The KBS should reason in the same way as the human expert, that is the KBS should emulate the human expert.

(b) ***Psychology-based knowledge elicitation.*** Extends the classical approach by basing knowledge representation schemes and elicitation procedures on trying to understand the psychology of what thought processes an expert uses when solving problems or making decisions.

(c) ***Machine-learning and knowledge engineering.*** This approach is based on the belief that application of machine learning techniques to knowledge engineering can improve the efficiency of the process, and may lead to the KBS learning directly from a human expert.

(d) ***Decision research perspective.*** Human judgment and decision making (JDM) research has shown that JDM behaviour is subject to a variety of cognitive biases, and that for most JDM tasks, simple linear regression models can outperform human experts. Accordingly, KBS's should help compensate for weaknesses of human expert reasoning, and should not emulate it (as in the classical approach).

(e) ***Software engineering.*** This approach assumed no differences between developing a KBS and a conventional IS, except that knowledge engineering occurs throughout the development of the KBS. Accordingly, advances and discoveries in the field of software engineering, for example, object oriented design, will also have great application in knowledge engineering.

(f) ***Empirical approaches.*** Many researchers within this field now believe that progress can best occur by resolving questions such as which KBS development methodology is the best, and that their resolution will only occur by conducting empirical analysis and experimental research.

(g) ***Philosophical perspective.*** This approach considers that knowledge engineering has a lot to gain from the formal models of reasoning found in traditional philosophical fields such as logic and epistemology. These models have had a significant influence on artificial intelligence in general and, will have the same influence within the development of KBS's.

However, how different will these strategies be? For instance, is it only a philosophical gloss, a superficial difference in the "skin" or outside appearance of the strategy? Alternatively, can more fundamental differences be identified between these strategies, if they are analysed in more detail? Finally, if fundamental differences do exist, then does that cause the overall development process for a KBS to be different? The answer to these questions will never be a simple "YES", "NO", or "BOTH". In terms of "YES", obviously no matter what approach to knowledge engineering is selected from the above group, there must be similarities in the development of different KBS's. At the broadest level, the activities that are encapsulated in the different development approaches are there to achieve the same result, that is construct a piece of intelligent or "smart" computer software that runs on computer hardware. Accordingly, any study that compares different KBS developments, must find similarity. For example, there would be regular meetings held with representatives from users, management, experts and developers to discuss the on-going concerns of the project (Mumford, 1986). The players or cast involved in the activity may differ slightly but processes and procedures associated with the activity will be essentially the same.

There are also other factors that different KBS developments would have in common. For instance, Ching (1987) in her literature review [p. 6-73, and 6-74] with respect to an interview with Sue Zawa (1986), states that different KBS's developed within the same organisation or across different organisations, should have similar objectives:

- The KBS should serve as a training vehicle, that is users of the KBS should be able to acquire knowledge from it. One way of doing that would be through help screens (similar to those found in conventional IS) but also through an a suitable explanation facility that can tell users how the KBS reached it's conclusions.

- Where interactive KBS's are developed, the KBS design should force users into conducting a structured dialogue. This ensures that consistent decisions are offered by the KBS across different time periods, as well as across different users of the KBS. For example, the KBS should avoid coming to a premature conclusion where insufficient information has been provided.

- The KBS should provide convincing solutions at the conclusion of its processing. Therefore user confidence in the KBS is increased, even if the system operates only at the assistant level and not at the expert level.

- While the knowledge domain may be complex, the KBS design should endeavour to make a KBS easy to use, that is you should not need to become an "expert" in using the KBS! For example, the KBS should require a minimum amount of user data entry, in order to avoid incorrect conclusions being reached because of user data entry errors.

- There should be an optimal trade-off between ensuring 100% accuracy within the inference process and the amount of time required for a user to complete a dialogue and have the KBS present a conclusion. Obviously the closer one gets to 100% accuracy the larger the number of questions that must be asked, with possibly the progression being a geometric one.

- The KBS should be easy to maintain. This objective can cover a number of issues. For example, there should be a loose coupling between the user interface and other components of the KBS. In addition, no matter what knowledge representation is used, the KBS should be designed in such a way that maintenance of the knowledge base is simplified. That is as much as is possible, new knowledge can be added, and existing knowledge changed or deleted without having to re-structure or re-compile the entire knowledge base.

With respect to KBS development, Irgon et al. (1990) claims that the following issues (which are not knowledge engineering issues) are often more important than the technical aspects of the KBS, namely [p. 39]:

> *"Commitment, cooperation, and ownership by the domain expert(s) throughout the development process facilitates knowledge acquisition. An actively involved and informed intended user community enhances acceptance of the final product. Both are therefore crucial to the overall success of the knowledge-based system.*
>
> *The most difficult and time-consuming issues are often non-AI issues: Design and development of the user interface and*

integration with conventional systems are absolutely essential, and require substantial time and energy. Plan to deal with these issues early in the development life-cycle.

Employ knowledge-based system technology for existing, persistent problems that are not easily solved by conventional software techniques."

Therefore, in summary there appear to be a number of factors that should lead to similarity in the development approach of KBS's both within the same organisation as well as across different organisations. These development approaches, such as prototyping, will be examined in more detail in a later section of this chapter. Furthermore, if Irgon et al. (1990) are correct, that the most important system development issues are not strictly knowledge engineering ones, then there may be little difference between KBS development and conventional systems development. These similarities could range over a number of different dimensions, such as the make up of KBS development teams, the sorts of activities that are involved in KBS development, and the types of system development methodologies applied to creating commercial KBS's. For example, prototyping can be used in both conventional and KBS development. Obviously the counter-argument is whether these factors dominate KBS development, and therefore are enough to make KBS development similar within and across organisations.

Are KBS's Developed the Same Way as Conventional IS?

As alluded to above, one question to resolve is whether KBS development is all that different to conventional IS development. That is, can conventional development approaches or even conventional development methodologies be used to build commercial production KBS's? If the answer to this question is

"YES", then this would also seem to answer the other questions raised in the previous section. Conventional system development activities are often similar if not equivalent, and if conventional development approaches are applied to KBS's, then similarity in development within organisations, and to a lesser extent across organisations should flow from this fact. As was discussed earlier with respect to different KBS developments, there will also be obvious similarities between KBS development and conventional IS and KBS development. At a very high level both produce a similar product, that is software that runs on computer hardware. Also at a high level, there will be similar activities undertaken in the development. For example, in both conventional IS and KBS development would conduct regular meetings with representatives from users, management, experts and developers to discuss the on-going concerns of the project (Mumford, 1986). As indicated earlier, the players or cast involved in the activity may differ between KBS and IS developments, but the processes and procedures associated within the activity would be very much the same.

Wielinga et al. (1992) in discussing the KADS KBS development methodology (which is considered in more detail later in this chapter) claim that although in 1983, there was little interest in KBS development issues. This was because most development occurred within a research context on special purpose hardware with specialist purpose software and adopting a rapid prototyping approach to system construction. However, as commercial organisations began to develop these systems, an awareness grew that at least for these types of developments, the KBS development process did not differ all that much from the development of more conventional information systems. "Aspects of KBS development such as information analysis, application selection, project management, user requirement capture, modular design, re-useability, etc. are similar to those encountered in conventional system development." [p. 5]

On the other hand, continuing with the above comparison, there will also be differences - some things may be harder to identify in a KBS development than with conventional IS developments. For example, issues concerning users for a new KBS may be more difficult. Rather than merely enumerating or identifying the potential users of a new MIS application, there are other more important issues concerning users of a KBS, and the user interface of the system. For instance, there may be significant knowledge or skill differences within the KBS user population. Potential users could be either laymen, para-professional, or professional people.

With respect to the first two categories, KBS design would have to consider issues such as avoiding technical terms which may have no clear meaning, and greater importance placed on the inferencing process and the explanation facility. Alternatively, for professional users who possess a certain degree of knowledge in the field (for example, medical interns using a medical diagnosis KBS), less attention will be required with respect to using technical terms, and the other issues discussed above. This issue would not arise in designing an MIS application, that is, whether or not the KBS should be designed for different knowledge levels within the identified user population.

Are these issues such as difficulty in identifying or defining the user population, that most knowledge elicitation occurs with a person who may not use the final production version of the KBS, or that consideration of a user population which is stratified are indicative that the differences individually may be small (however, collectively significant?)? Another way of putting this is - are these issues at a "macro" level in terms of their impact on the overall development process, or are they merely at the "micro" level? In this context "macro" means that entirely different phases within a system development methodology are required for a KBS as against a conventional IS, and "micro" means that although similar phases are required to develop both types of systems, different tasks need to be completed within each phase for the development of a KBS as against a conventional IS. The literature seems to indicate that differences occur at both levels. For example, we would expect some form of systems analysis to occur when developing either a KBS and a conventional IS. However, at the "micro" level, this phase is usually conducted with the ultimate users of a conventional IS, but is performed predominantly with the expert(s) in a KBS. More importantly, one would not expect the expert to be a user of the KBS, meaning that an important feedback mechanism for quality control during KBS development is lacking. In this section of the review of the literature, to support the arguments put forward, we will consider one issue at both the "macro" and "micro" levels. At the "macro" level, a comparison will be made of critical success factors for IS and KBS development. At the "micro" level, a comparison will be made of requirements elicitation for conventional IS compared with knowledge acquisition for KBS development.

Critical Success Factors (CSF's) for System Development

Comparing KBS and IS Using CSF's

The CSF method, popularised by John Rockhart (1979), tries to identify fundamental or critical issues and objectives, to then translate these into an action plan. The CSF method focuses upon things that are important. The outcomes of applying the method normally represent collective rather than individual views. Interviews are held with decision makers to find the few key areas of activity in which favourable results are absolutely necessary. In other words, CSF's are the things that must go well to ensure success, and so represent areas to be given special and continuous attention to bring about optimal performance. Accordingly, identifying the CSF's for successful system development of a conventional IS (or MIS) and a KBS should form a good basis to examine "macro" similarities and differences. However, before proceeding with this sort of analysis, there have been some criticisms of the CSF methodology should be dealt with first, in order not to confound the analysis.

For example, Boynton and Zmud (1984) claim that the CSF method has three principal areas of weakness. First, it has been asserted that the CSF method is difficult to use, and is therefore not appropriate for organisations whose analysts do not possess the capability to successfully apply the method. However, this is not a failing of the CSF methodology, but rather a failing of the person(s) attempting to use it, and accordingly should not invalidate the analysis outlined below. In other circumstances, the same comment could be made about non-automated IS development methodologies.

Second, the validity of the CSF method has been questioned because of possible analyst or manager bias introduced through the interview process. Nevertheless, there is evidence in the literature that both supports and refutes this assertion. For instance, evidence in support of this claim is provided by Boynton and Zmud (1984) [p. 23], who noted in one case study, that one of a firm's senior managers developed a great interest in CSF's and championed the project. The list of personal CSF's that this executive developed closely paralleled the final aggregation for the firm as a whole. In addition, personnel involved with specific operating areas tended to identify CSF's that were limited in scope but which they viewed as being corporate-wide in nature. However, a study by Munro (1983) [pp. 67 - 68] has shown that two independent CSF analyses (performed albeit by very skilled CSF analysts) yielded comparable results, thus indicating that these potential biases can be overcome. It is unlikely that these independent studies would have been consistent unless there was considerable validity to the premise of no overt introduction of bias. Notwithstanding this, the application of the CSF method here will not use interviews, and again this perceived weakness should not detract from the analysis outlined below.

Third Davis (1979) [pp. 57 - 58] raises several concerns about the CSF method being an appropriate requirements analysis methodology. Given that humans have a limited capacity to deal effectively with complexity, the CSF method may yield an information model that is simple and thought provoking but is not fully representative. In simple terms important things may be left out at the beginning, and the CSF methodology lacks a procedure which ensures that all CSF's have been identified. The analysis procedure that leads to CSF identification may be biased by the manager's or analyst's beliefs and values or by the available data. It seems that in the first two instances, the arguments are not conclusive either way, and with respect to the third, the use of the method in this case is not for requirements analysis. This weakness presents a more difficult hurdle for the analysis to surmount. However, in defence of the analysis, in many circumstances determining what is or what is not a fully representative list of CSF's is never a completely objective process, but will involve subjective opinion that is submitted to critical analysis and review. In the case of KBS CSF's at least, this analysis and review is provided.

CSF's for Conventional IS Developments

Accordingly, although the research literature does not offer a lot in this area in support of the following, we consider that an appropriate list of CSF's for the successful development of a conventional IS would be:

- full and enduring commitment by all relevant sections of the organisation to development and commissioning of the system

- exhaustive identification and enumeration of the user population
- accurate and exhaustive listing of all relevant user requirements
- selection of the most appropriate system development methodology based on the contingencies that are apparent with the development environment

CSF's for KBS Developments

It should be remembered that although the factors outlined above have relevance to KBS development, the goal of the CSF method is to find those factors that are "critical" and not merely applicable. Accordingly, with this in mind, the CSF's for a successful development of a KBS are considered to be:

- selection of an application well suited for development as a KBS
- appropriate delimitation of the knowledge domain to be represented, and competent selection of the right expert who is willing to take part in all phases of the KBS development
- selection and use of the most appropriate knowledge acquisition methodology and techniques
- selection and use of the most appropriate knowledge representation methodology to store the knowledge in the KBS
- construction of an appropriate maintenance strategy to preserve the integrity of the knowledge base as knowledge changes across time

As can be seen from the above, the CSF's for each type of development are different. Is there any support for these contentions within the research literature for the KBS CSF's? The answer seems to be "YES". For example, the importance of selecting a suitable application (the first KBS CSF) is discussed by Ching (1987), who makes the following comments (drawn from the work of Zawa (1986, 1987)). The selection process should have rigour, that is broken down into considerations of the problem, management and potential user support, and the expert, say using a checklist such as [pp. 6-16]:

"The problem:

- *The problem is an* ***acknowledged problem area****, where the existing system is inadequate. This could be due to high staff turnover, shortage of skills or expertise and the impending retirement of an expert.*
- *The implementation of the KBS will provide* ***significant benefits*** *to users, thereby gaining user commitment (for example, an improvement of working conditions).*
- *The problem has an* ***identifiable solution****, so that the success of the KBS can be monitored and the system can be tested.*
- *The solution of the* ***problem does not rely on intuition****, a large general knowledge of current world situations (unless the KBS is told by the user), since the KBS technology is incapable of containing that knowledge.*
- *The problem has a* ***definable scope*** *so that users can be satisfied with its limited bounds.*
- *The problem is* ***not too broad nor complex****, thereby decreasing the development time, easing the development process and increasing the probability of success.*

- *The problem is **not too trivial**, so it can be used to prove that the technology works. If the problem can be solved without using the system, the users will be able to by-pass the system.*
- *The problem is **not suitable for implementation using conventional programming languages**. KBS's can usually be built quickly using KBS shells, but they require more computer resources than conventional systems. If the problem can be solved using conventional methods then it is recommended to do so, but if the problem requires analysis of rules or varying degrees of variables, then a KBS can be more suitable.*
- *The problem is **not a one-off problem**, thereby warranting the implementation of the system.*
- *The system will **not be made redundant** in the foreseeable future, so that investment made in building it would not be forfeited."*

This work is also supported by Waterman (1985) and Canning (1987). In terms of selecting a suitable expert for knowledge acquisition McGraw et al. (1986) raise this issue as a very important step within the development process. That is, the selection of the expert is a CSF for KBS developments being more than it would be for traditional MIS and IS applications. In the latter there is no overriding need for potential users of the IS or MIS to be experts in their domain. All the MIS or IS has to do is satisfy their information needs. Again Ching (1987) offers the following comments regarding this CSF [p. 6-17]:

"The Expert

- ***At least one possible expert** exists in the problem area, and he/she must be willing to give his/her time for the project and be supportive and committed.*
- *The expert **is able to communicate with the knowledge engineer** and present his knowledge in a form understandable by the knowledge engineer.*
- *It would be beneficial if **the expert has some knowledge of computers**, making it easier for the expert to visualise the final system."*

With respect to the third and fourth CSF's, Khoo (1986) discusses the importance of selecting the right knowledge acquisition tools and techniques to ensure overall success of the KBS development. The importance of using the right knowledge representation tools is considered by Jamieson and Szeto (1987), stating that it is important to consider such issues as : (a) the accuracy of representing the expert's knowledge correctly; (b) the intelligibility to users, in terms of reading and understanding the knowledge represented and the problem processes employed; (c) the ability of the representation technique to incorporate additional rules; and (d) the explicitness of the knowledge representation where the expert(s) and developers can test the knowledge base and easily determine its overall integrity.

Furthermore, Canning (1985) states that the two ultimate attributes which affect the choice of knowledge representation are : (a) the power to express the expert knowledge; and (b) the simplicity to describe, update and explain the knowledge in the model. Selecting the appropriate knowledge technique for a given situation is difficult as it is generally believed that no single knowledge representation technique is optimal. Baldwin and Kasper (1986) agree with Canning's contention stating [p.16]:

> *"... The choice of a data structure (knowledge representation technique) depends on assumptions about how that data will be used"*

and that a critical problem in knowledge representation lies in the fact that the choice of a particular knowledge representation technique is not facilitated by an existing framework (Baldwin & Kasper, 1986, p. 186). Their answer is to develop a contingency approach, based on what environment the KBS is expected to operate in, namely : (a) operating, where decisions are based upon well established routines or guidelines; (b) co-ordinating, where decisions are not as structured requiring more estimation and judgment, for example budgeting, man-power planning; (c) exception, where decisions are made on an ad-hoc basis but are not significant; and (d) strategic, where decisions are made on an ad hoc basis, and are significant. The following table shows the type of knowledge representation that would be most likely within each of these scenarios (Baldwin & Kasper, 1986, p. 170):

Table 12. Knowledge Representation Techniques for Specific Situations

Decision Type	Function of System	Likely Knowledge Representation
Operating	Replace the decision maker	Production rules
Co-ordinative	Support the decision maker	Frames and semantic networks
Exception	Support the decision maker	Frames and semantic networks
Strategic	Support the decision maker	Combination of techniques

(Source: Baldwin & Kasper, 1986, p. 170)

Finally with respect to the last CSF concerning maintenance, McCallum (1985) identifies some of the major maintenance functions for a KBS: (a) changing weights on rules due to refinement of knowledge; (b) replacing knowledge that is found to be incorrect, inappropriate or inaccurate; (c) discarding knowledge; (d) updating of time dependent information; (d) adding new conclusions; (e) adding new rules; and (f) adding or replacing comments. It should be remembered that knowledge acquired and represented within the knowledge base of a KBS merely portrays a snapshot of the expertise which, as it is now frozen in time, must become obsolete with the passage of time. User requirements also become obsolete but this has a less critical impact on the operation of a conventional IS.

This raises interesting questions in terms of maintenance, for example should all knowledge encapsulated within the knowledge base have a "use-by" date, which when passed means that no further access to that portion (or indeed all) of the knowledge base can occur until some form of knowledge maintenance activity has been completed and registered. Furthermore, assessing whether or not knowledge is obsolete would be a far harder and more complex task than assessing whether user requirements have become obsolete. Finally Ching (1987), in her literature review, states that the frequency of maintenance within a KBS will be higher than a normal IS at the start of the KBS's life. However, as the KBS matures the maintenance load should approximate that of a normal IS.

In summary, it would appear from the above that the contingencies associated with these factors will make conventional IS development and KBS development different at the "macro" level.

Comparing Requirements Elicitation and Knowledge Acquisition

For these two activities, similar to what has been found with other sections of this chapter, there are similarities as well as differences between requirements elicitation and knowledge acquisition (KA). The overriding question here to be answered is whether the similarities are superficial and the differences fundamental, or vice versa. With respect to similarities, within the field of knowledge acquisition research, although researchers may label activities as purely knowledge acquisition, functionally there is very little difference between these "KA" type activities and those used for requirements elicitation in conventional system development. For example, McGraw and Harbison (1989) state, that the most widely supported view of knowledge acquisition stages consist of the following major tasks, namely: (a) initially entering knowledge; (b) reducing or avoiding erroneous knowledge; and (c) augmenting acquired knowledge. From these basic tasks the authors have developed the following framework for knowledge acquisition:

- ***Identification*** - characterising key problem aspects including participants, characteristics, resources and goals. In this phase the knowledge engineer becomes familiar with the domain.

- ***Conceptualisation*** - which involves specifying how the primary concepts and key relationships among the concepts in the domain are depicted and related by domain experts.

- ***Formalisation*** - in which the knowledge engineer maps the recognised concepts, sub-tasks, relations, and other information into formal representational mechanisms.

- ***Implementation*** - in which the representational forms of knowledge developed in the previous phase are encoded either into the language selected for the KBS development or encoded into the selected expert system shell.

- ***Validation*** - in which the system is evaluated as to the efficacy of the system's formalisation, basic assumptions, accuracy and knowledge acquisition efficiency using test scenarios and problems.

However, what McGraw and Harbison have said about knowledge acquisition could equally be applied to requirements elicitation (although the technical terms may differ slightly); and in performing the tasks outlined above, would in most circumstances ensure that an effective determination of users' requirements has been conducted within a conventional system development. In a similar vein, Johnson (1984) identified four different knowledge acquisition procedures, namely:

- ***Descriptive Methods.*** Interviews are held with the expert with the objective, through the answers given, to obtain the knowledge and skill in a given task situation. Although a representational language may be developed to ease knowledge acquisition, the major difficulty with this method centres on the problems an expert has in characterising or describing their knowledge and decision process.

- ***Observational Methods.*** The method relies on observing the expert in situ. Such techniques as protocol analysis enable the knowledge engineer to record heuristics and models used to solve problems which the descriptive method may miss. The major difficulty with this method is in trying to simulate, as closely as possible, the real world environment in which the expert works. The experimental conditions may impose a bias which can influence or alter the expert's actions and thought processes.

- ***Intuitive Methods.*** Two streams of intuitive methods exist; the "pseudo-expert" form (the knowledge engineer attempts to become familiar with the task environment and knowledge involved in building the KBS) and the "true expert" (in which the expert attempts to become a knowledge engineer for the KBS development).. In the former case the knowledge engineer then incorporates their acquired knowledge into the KBS. In the latter, the supporting software must be comprehensible to the expert, who might not have any computing background. The limitations with this approach are self-evident. Where the knowledge engineer becomes less than an a competent expert or the expert becomes less than a competent knowledge engineer, then this phase will be at best incomplete.

- ***Knowledge Acquisition Software.*** This approach attempts to automate the knowledge acquisition phase. Some software has been produced to automate the expert interviewing process. Another example, is the "Expertise Transfer System (ETS)" (Jamieson and Szeto, 1987), a package which would provide such capabilities as removing and adding rules to the knowledge base and checking for rule consistency.

Again, substituting expert with user, and knowledge acquisition with requirements elicitation and these methods would be equally applicable to either KBS or IS development. However, in terms of the detail in which these approaches ,are employed, significant differences are apparent, as requirements elicitation is concerned more with the overall structure and delivery of the conventional IS, whereas knowledge acquisition is not only concerned with this, but also with the types of data that will eventually be encoded into the knowledge base. This is considered to lead to more fundamental differences between the activities. For example, Kahn and Bauer (1989) raise the following issues that should be addressed during knowledge acquisition for a KBS development, namely [pp. 48 - 49]:

> "1. ***The scope and granularity of knowledge.*** *What knowledge is required to solve a problem; and to what level of detail? In a diagnostic application, for example, is it necessary for a system to understand how experts generate an ordered list of hypotheses to pursue or only that they do? How explicit a representation of order should there be?*
>
> 2. ***The degree of procedural regularity in the use of knowledge.*** *Is a diagnostic solution always achieved in the same way, or are there alternative techniques for the different failures that occur?*
>
> 3. ***The need for and availability of data at run time.*** *What can be considered permanent knowledge versus what must be acquired at run time? What are reasonable default assumptions? Is it reasonable, for example, to assume the possibility of extreme temperatures in Arizona in the summer or should temperature factors be elicited at run time?*
>
> 4. ***The degree of accuracy acceptable in a resulting recommendation or situation assessment.*** *Is it acceptable to terminate an analysis of a diagnostic*

> *problem with the identification of a likely cause or should all possibilities be considered?"*

These sorts of issues, while important for a KBS, would be of no importance (with the possible exception of "3" above) to a conventional IS. The implication here is that knowledge engineers may have to look further into the knowledge being acquired than systems analysts would have to with regard to user requirements. Kidd (1987) supports this proposition from another point of view [p. 8]:

> *" ... when building an expert system for practical use, the situation is complex. A medical expert, for example, does not simply carry out diagnostic reasoning. Rather, this reasoning is part of some functional role (or "modality") that he fulfils within his domain of expertise. For example, at any time he may be providing advice in his role as a consultant to his patients or be providing explanations in his role as medical tutor to a group of students.*
>
> *This point has been constantly overlooked in the development of expert systems and may largely account for their lack of success in practical applications despite their problem solving prowess (Coombs and Alty, 1984, Kidd, 1985b). As Breuker and Wielinga* [see article in Kidd (1987)] *point out, the distinction between roles (or modalities) and problem solving tasks has not even been recognised. They cite the example of the Hayes-Roth et al. list (see p. 5), where for instance "Instruction" (a modality) is included alongside "Diagnosis" (a task). This is a crucial point because changing modality of an expert system is not (as was originally thought) merely a case of adding a different modality specific front end. Clancey's unsuccessful attempts to convert*

MYCIN into a tutoring system illustrate this point most clearly. He found that a complete re-construction of the underlying knowledge base was required.

... In designing an expert system, therefore, the first step must be to decide on the appropriate modality for that system. If, as is the case in the majority of expert systems, the desired modality is one of domain consultant, then the next step should be to identify the set of problem solving tasks that are the minimal requirements of a consultant in the selected domain. One way of doing this is by recording and analysing naturally occurring consultation dialogues between experts and their clients."

If anything has emerged from the above discussion, it should be that knowledge acquisition, is far more than a simple process of sitting down with one or more experts, having a chat about their field and then encoding that knowledge into a suitable knowledge base. The process is far more complex than this simplistic view. In the first place knowledge is not monolithic, but can be broken down into the following categories: (a) factual or theoretical knowledge (Goranzon and Josefson, 1988), gained from performing such activities as empirical, normative or rationalist research; (b) situational or tacit knowledge (Klein et al., 1989), gained mainly from experience, which gives an expert the ability to make assessments and take actions in unique situations;, and (c) judgmental knowledge (Fischhoff, 1989), which is similar to tacit knowledge except that this experiential knowledge allows experts to construct suitable models of processes that incorporate either incomplete data or uncertainty. This level of detail would never be considered in terms of user requirements for a conventional IS.

Another fundamental difference between requirements elicitation and knowledge acquisition is the introduction of a new player in this phase, that is the expert. The KBS research places great importance on the expert, and a concomitant lack of attention for users, in terms of their current domain knowledge and their requirements as to what the KBS should provide. An interesting question to resolve would be whether in the development of commercial KBS's is whether the importance of these groups is the same. That is, in terms of knowledge acquisition for commercial production KBS's, can and do users play as equally an important role as the expert? Are users ever perceived to be sources of knowledge? The answer to this question should be "NO", as it would seem that if potential users of the KBS provide its knowledge, then these users should then be classed as "experts" rather than novices. If this is the case, it must bring into question the utility of continuing the KBS development, that is has a suitable problem been selected? Alternatively, is the classic situation of the greying expert still appropriate. In this scenario, the organisation is about to lose the sole repository of special or unique knowledge that the organisation can ill afford to forego. In this situation, KBS development is undertaken in order to preserve the expert's knowledge and expertise within a computer system.

An important issue concerning users during KA is their level of interaction and relationship with the expert - little and intermittent, or significant and constant. Furthermore, the KBS research literature does not seem to have properly addressed the role of users in other KBS development activities, such as their contribution to the verification and validation of the knowledge base, the testing of the user interface, consultation structure and dynamics, and so on. Kidd (1987) claims that little emphasis has been placed on the contribution of users [p. 9]:

> *" ... In any interactive expert system, the user is an active agent in the problem solving process. In the past, the design of both expert systems and intelligent tutoring systems has tended to be*

based on the assumption that the system, in some sense contains the true model of the domain and the aim is to transfer this truth in to the empty head of some passive user". However, rather it is the case that any user, no matter how little he or she knows about the domain will still be actively acquiring and organising the expert's knowledge for some purpose of his or her own. There are always two models of the domain, the system's and the user's. Neither is "true" in some objective sense, but each may be organised for different purposes."

Therefore, given we accept this argument, there appears to be a fundamental difference between KBS development and conventional IS development in this area. As a side issue, although this conclusion has been reached, we could ask whether or not it should be so. Given that KBS development has often failed to deliver an effective product to the organisation (see discussion in Chapter 1), should there be some movement closer to the type of involvement that users have in conventional IS development. That is during knowledge acquisition, should there be a similar level of the user involvement as there is within the requirements elicitation and systems analysis phases? Rather than being a last minute ad hoc exercise, if users play a more active role in the knowledge acquisition process as well as other KBS development activities, would this lead to more successful KBS developments? Whether more successful developments would result may be moot, but Kidd (1987) states that more user involvement should occur, specifically knowledge acquisition should address questions such as [pp. 9-10]:

(a) Identifying the different classes of users likely to use the system and their different needs.

(b) Analysing user requirements, for example what are the common classes of problems and questions? What advice does the user require and in what form? For example, do users have their own idea of a solution and only require a critique from the expert? Do users need to have a set of alternative solutions with pros and cons spelt out? What type of justification is required?

(c) Analysing what types of knowledge the user brings to bear on the problem solving process, which may range across the following: the user's goals within the domain; constraints on acceptable solutions, for example time, availability, cost; models of the current problem and the type of problems that the system can solve for him.

Accordingly, it would appear that similarities between knowledge acquisition and requirements elicitation are more superficial and that the differences between these two activities are more fundamental.

Integrating KBS With Other IS Products

The final issue that should be considered with respect to KBS development and conventional IS development is the level of integration that occurs between these two pieces of software. A number of questions that should be investigated and/or resolved with respect to this issue. For example, what level of integration exists between KBS's and conventional IS? Second, for KBS's with a high level of integration with conventional information systems, what impact has this had on their development - little or significant? Third, does integration have a significant impact on: (a) the type of KBS technology used; (b) the problems selected for resolution by application of KBS technology; and (c) the development approach used to construct the KBS?

As far as the research literature goes, Bowerman and Glover (1988) state that KBS development has focussed on creating stand-alone systems. However, Bowerman and Glover (1988) contend that KBS's need to be integrated with the rest of the organisation's production systems before they can begin to benefit the organisation. These authors claim that this integration could occur in a variety of ways, either through links with the organisation's MIS or if it is a manufacturer, through links with computer integrated manufacturing (CIM) systems. Bowerman and Glover (1988) claim that experience has shown that expert systems need to be coupled with accounting, real-time sensors, alarms, database management systems, graphics displays, reporting facilities, and other systems interfaces in order to be perceived by commercial organisations as productive and profitable. The authors claim that the field of process control is one of the leading CIM applications which is ripe for integration with KBS's.

There appear to be three levels of integration for a commercial production KBS's, viz:

- little, in which case the KBS operates basically as a stand alone system, although it may be installed and accessed through an organisation's local area network;

- moderate, where integration is concerned with the sharing and passing of data between two independent systems; or

- significant, where the KBS is either embedded within a conventional information system, or has been fully absorbed into the other work practices and information processing activities of its client user population.

An interesting question to resolve would be what level of integration is currently apparent within Australian commercial organisations? Bowerman and Glover (1988) would assert that the level is either little or at best moderate. These authors state that a major obstacle preventing any real progress with respect to integration of KBS to other mainstream IS products is the lack of knowledge-base interface standards. In fact Bowerman and Glover (1988) claim that although low-level hooks are available, there are no high-level interfaces and that KBS technology is in the same position as traditional computer standards of a few years ago; each individual manufacturer's products are distinct and are not designed to easily interface the products of other vendors. Accordingly, there is a need to develop standards similar to the OSI project in the area of data communications.

To conclude this section, the above discussion has displayed that the research literature considers there are significant differences evident between conventional systems development and KBS development. The discussion has indicated that these differences exist both at the "macro" level of system development as well as at the "micro" level. Furthermore, the level of integration between KBS's and other conventional IS products is low. Accordingly, these conclusions should then mean that there is little in common between the system development approaches and activities for conventional IS and knowledge based systems.

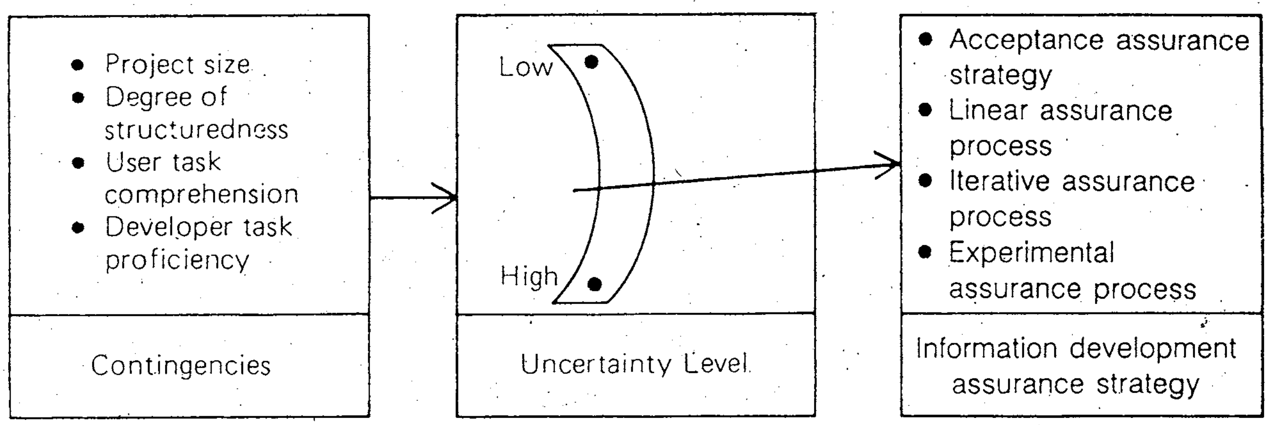
Case Study Research - Development of KBS's

Another avenue that should be explored is to determine whether the research literature in reporting on KBS developments has a focus on one particular development approach rather than on a number. In this area, it seems that the analysis should be concerned with whether or not prototyping is the best method of developing KBS's.

Uncertainty and the Mode of System Development

With respect to KBS development (considering the points already discussed above), it is argued that a common feature of all projects will be high task uncertainty and high development environment uncertainty. Accordingly, using the contingency approach as outlined by Davis and Olson (1985) leads to determining a test for this assumption. This approach is summarised in the diagram below [Davis and Olson, 1985, p. 567]:

Figure 3. Contingency Theory of System Development



(Source: Naumann et al., 1980, p. 277)

Davis and Olson (1985) outline four development approaches. Selection of a particular approach depends on the level of uncertainty that impacts on the development. Uncertainty is attributable to four factors, indicated in the figure above, namely project size, degree of structuredness, user task comprehension, and developer task proficiency. The four approaches (in terms of increasing uncertainty) are [pp. 565-566]:

(a) ***Acceptance assurance strategy.*** User statements of requirements are accepted as complete and correct. System development proceeds strictly in accordance with these requirements which do not change, and with no procedures in place to question or determine the veracity or accuracy of these requirements. Typical system developments in this category would be file conversions, reports from existing files or databases, and some simple, single-user decision models. The systems produced would typically be small, highly structured systems that are well understood by users.

(b) ***Linear assurance process.*** With higher uncertainty, this process allows for requirements to change slightly. The approach is to proceed stepwise (or phase by phase) from beginning of development to final delivery. As each phase is completed there are assurance procedures or "sign-offs", by say a steering committee, to proceed to the next stage. No iteration back to earlier phases occurs under this approach. This approach would encompass most formal life cycle methodologies. Typical system developments would be for systems with highly structured requirements, user task comprehension high, and developer task proficiency high.

(c) ***Iterative assurance process.*** Where requirements uncertainty during development is moderately high, the traditional life cycle methodologies of linear assurance are often modified to include iteration. Whenever requirements are found to be wrong or inadequate during development, the specifications are reviewed by a return to the requirements determination process with users. The sequence of activities in the life cycle can therefore be repeated as often as necessary in order to obtain the requirements and develop a system to achieve them. This approach assumes that a correct and complete specification of requirements can be obtained if sufficient iterations are used. Examples where the iterative assurance process is applicable are large multiple-user systems and application areas that are new to the user or developer organisation.

(d) ***Experimental assurance process.*** A high level of uncertainty during development may require an experimental approach in which system requirements are obtained through actual user experience with the system being developed. The normal approach adopted is to perform some sort of prototyping, in which successive prototypes endeavour to reduce uncertainty by producing closer approximations of actual requirements (that is there is a convergence on actual requirements). Users and developers identify the shortcomings of a prototype or eradicate or improve on these flaws in the prototype's next version or generation. Typical systems developed under this approach are decision support systems for upper management, interactive forecasting models, and unstructured systems to be implemented for multiple users.

Accordingly, if KBS developments are new to organisations, where users have little understanding of the technology, and developers possess insufficient skills to correctly apply the technology, then it would seem that high developer and user uncertainty would exist. Therefore out of the four approaches outlined above, the most common KBS development method should be experimental assurance. There is evidence in the KBS development case study research literature to support this contention during the 1980's (assuming of course that to develop a prototype, developers used prototyping). For example, Basili and Ramsey (1985) reported on a prototype expert system for software engineering management (Arrowsmith-P). Biggs and Selfridge (1986) discussed a prototype expert system concerning assessment of a business as a going concern (GC-X). In the same area Wilson et al. (1988) discuss another prototype expert system whose objective is to evaluate a business as a going concern. Zawa (1986, 1987) in her reports has as her first objective development of a KBS through a number of working prototypes. Finally, support for the above contention is also found in a review of KBS development within the field of auditing performed by Jamieson (1986) - for more detailed information, see Appendix 2. As indicated by the information presented in this Appendix, up until recently most KBS development within the auditing field had not passed prototype or demonstration stage.

Prototyping - Still the Preferred Development Approach?

What does the term "prototyping" mean? Although the product delivered by the process may differ across different developments, the underlying technique remains the same. Prototyping aims to deliver a complete but "cut down" version of the system, which is either broad and shallow (the prototype displays all required functionality but inferencing is only superficial) or narrow and deep (the prototype displays only a portion of the required functionality, but for that portion inferencing is far more complex). There is some case study evidence to indicate that even towards the end of the 1980's prototyping was still considered an effective way of developing KBS's. For instance, Hickman et al. (1989), although they have reservations concerning this approach to KBS development, claim that this approach is quite popular. It is the most common approach suggested by KBS shell developers, and one of the most commonly adopted by novice KBS project leaders and system developers. Hickman et al. (1989) consider it a less than effective approach, particularly when coupled with an expert system shell, as it may lead to the situation of looking for problems that suit the expert system shell, or even more dangerously, modifying the problem in order to match the shell's capabilities. Some confirmation of the prototyping approach is provided by Cupello and Mishelevich (1988), who state [p. 534]:

> *"... we propose that the most efficient way to investigate this technology is to build a demonstration prototype. This prototype is the first step in a three-stage process that is typically followed for building K/ES. ... The second stage is the full prototype phase that incorporates both breadth and depth across the problem domain, but is not in a deliverable form for daily use ... the third stage is the delivered system, useable on a day-to-day production basis."*

These authors claim that it is important to recognise the strategic importance of the demonstration prototype, which is used to demonstrate to senior management, domain experts and other users the value or the potential of the proposed KBS. However, as a note of warning, Cupello and Mishelevich (1988) indicate that development of the final product will usually involve significant product enhancement and software engineering activities, and not be just a simple extension of the demonstration prototype. Irgon et al. (1990) reported on system development for five KBS's produced by Bellcore, a company within the AT & T corporation, which were discussed earlier; in four out of the five systems, the prototyping technique played a key role in their development.

Hickman et al. (1989) classify the prototyping activity into two approaches - either a "throwaway" or "keep and enhance" approach. With the former there are normally either two or three "generations" of prototypes, namely exploratory or conceptual prototypes, experimental prototypes, and field prototypes. Conceptual and experimental prototypes are generated early in the system development. Their main objective is to improve communication and understanding between developers, experts and users. Developers gain a better understanding of the knowledge domain, and therefore a clearer idea of how to best apply KBS technology to the solution. Experts and users on the other hand gain a better idea the capabilities of the technology and some indication of how the final production version of the KBS will operate. One of advantage of prototyping therefore is that it ensures user involvement early within the development of the KBS. Development of conceptual prototypes can also be viewed a means to confirm the requirements specification for the KBS.

The experimental prototype, is also used in the early stages of the design phase, in which various design ideas are canvassed to determine the best approach to implement the final system. Remember that under this strategy, the conceptual prototype is not used as basis to develop the experimental prototype. The tool used to implement these prototypes may not be used to develop the final production version of the KBS. Development of these prototypes is often referred to as "prototyping the design". Field prototypes are versions of the KBS that are still considered to be under development but are released to users for in situ testing and further development. At times development may cease here without going to the next stage of developing a full production version of the KBS.

With the "keep and enhance" approach, Hickman et al. (1989) claim that it is often difficult to determine the difference between what a system developer states is a "rapid prototyping" approach to KBS development, and just another variation of the incremental or evolutionary system development strategy. In contrast with the strategy alluded to above, rejected prototypes are not discarded and the new version of the prototype emerges from the old. System development effort is directed towards correcting flaws and enhancing the prototype KBS's functionality, to turn it from something that was rejected into something that is acceptable. In these situations, the developer has a focus on solving current problems and issues, and may lose sight of the overall development goal. In some cases, Hickman et al. (1989) claim that considering the development activity as "rapid prototyping" may be inapplicable; and further that a perception of using a so-called "system development methodology" is merely a "figment of the KBS developer's imagination".

Accordingly as indicated above, Hickman et al. (1989) are not convinced that prototyping is the preferred system development approach, as they consider this approach has some disadvantages, namely:

- **Expectations of users**. The ease with which a small conceptual KBS prototype is developed using a shell can lead to false expectations by users, and more importantly project sponsors, about the pace of development and the delivery of the final product, as significant or insoluble implementation problems may exist in converting the final version prototype into an efficient production KBS.

- **Integration with other systems**. Where rapid prototyping is linked to expert system shell software, there may be significant problems in delivering the production version of the KBS. The required level of integration may be beyond the capabilities of the expert systems shell to deliver.

- **Inefficient development overall.** Hickman et. al (1989) claim that "rapid prototyping" in many circumstances will not be a total methodology for system development, and has inherent inefficiencies [p. 37]:

  > *"Development of prototypes is usually an unplanned, ad hoc, iterative affair - there is no underlying life-cycle model to guide and control. The direct coupling between specification, design, and implementation phases means that each iteration involves re-specification, re-design, re-implementation, and re-evaluation activities."*

  The amount of effort and expenditure of resources on re-performing development activities may render this approach commercially unacceptable.

- **Ineffective development overall.** Using prototyping, major design decisions made early in the development process, get "locked in".. If these decisions are discovered to be wrong, for example knowledge acquisition is flawed, this may compromise the whole design strategy, leading to project termination or backtracking to develop a new conceptual prototype. Backtracking is not only wasteful of resources, it also makes quality assurance, and change management procedures very difficult to undertake. Accordingly, further errors may be incorporated into the current version that were not present in the earlier version that was scrapped. System development may be perceived as working very hard, but very little actual progress is being made.

Use of prototyping is supported to some extent by Weitzel and Kerschberg (1989) who state that when the prototype has been validated, the KBS development team has four choices as to what they can do with the prototype, namely [p. 487]:

> "1) *use the prototype as the production system*
> 2) *write off the prototype as a learning experience*
> 3) *use the prototype as the basis for a "Mark-II" system*
> 4) *use the prototype as the specification for a conventional SDLC methodology"*

However, like Hickman et al. (1989), Wietzel and Kerschberg (1989) have indicated that prototyping does not form a complete KBS development methodology. That is, in order to convert the "Mark II" prototype into a production system, a complete re-development of the entire KBS is performed using a formal systems development life cycle methodology.

Prototyping is Not The Answer

In addition to the comments outlined above, Slagle et al. (1990) have stated that a major problem with expert systems is maintenance, due either to a changing domain (which often leads to re-development or replacement of the existing KBS), or that the expert system was developed using ad hoc procedures (which to a large extent can be solved by adopting a more formal and rigorous approach). Slagle et al. (1990) claim that this ad hoc approach is increasingly a poor selection given the growing number of commercial organisations that have an increased interest in using this technology. These authors also believe that prototyping (following the Hayes-Roth et al. (1983) and Waterman (1986) evolutionary models of demonstration prototype, research prototype, field prototype, and finally a production system), may be a poor choice of development methodology. Slagle et al. (1990) consider the effort expended in converting a poor KBS solution of the problem into a solution that is considered better as unnecessary effort - both by the developers and the users of the ultimate KBS [p. 30]:

> *"But modifying a prototype to produce a production system presents several difficulties - a prototype usually contains many ad hoc changes and is poorly documented and sloppily written. This type of modification often requires two to three times the effort required to develop the prototype - a significant investment."*

Their approach is slightly different, namely create and update a knowledge specification during prototyping, and then use this knowledge specification to re-implement the system when the production version is being built. The knowledge specification can be generated as part of the knowledge acquisition phase (which will also generate the first prototype system), which can then serve as the basis for the KBS through its entire life cycle - from knowledge acquisition to maintenance. As Slagle et al. (1990) state [p. 30]:

> *"Re-implementing the system from a knowledge specification is likely to take less time and money than improving the prototype, and will result in a system that is easier to maintain. Although the knowledge specification does not solve all the problems of developing a system - including performance and user interfacing - it is a powerful means for capturing knowledge and communicating among developers and experts."*

The knowledge specification must: (a) be easy for experts to read and understand for verification or validation; (b) be clear, concise and unambiguous; (c) enable easier modification of knowledge already acquired; and (d) represent knowledge in a format ready for entry into a knowledge base. Slagle et al. (1990) have adopted protocol analysis as the main source of knowledge and conceptual structures as the knowledge specification formalism (Sowa, 1984; Sowa, 1989). The methodology consists of five stages, namely:

1. ***Requirements analysis.*** Select the system and then identify: (a) the objectives, type and scope of the system; (b) the constraints of the system -including availability of experts, the project schedule, funding, computing facilities, and development tools; and (c) an initial set of test cases (which unlike the next phase emphasise testing for system acceptance) and expected results for system acceptance.

2. ***Knowledge acquisition.*** Knowledge engineers develop a basic understanding of the domain. In this phase the associate expert helps the knowledge engineer develop test cases - that is problems - to be used for collecting protocols from the primary expert. Having been given these test cases, the primary expert thinks aloud while solving the problems which are tape-recorded and then transcribed.

3. ***Knowledge specification.*** Protocols are analysed to identify expert problem-solving strategies and the results are represented using conceptual structures or other representational forms. The authors' methodology (in using conceptual structures) has six structures, namely type definitions, schemata with and without actors, implications, facts, a type lattice, and canonical graphs. To develop these structures, Slagle et al. (1990) have outlined

the following steps: (a) identify key concepts; (b) identify lines of reasoning; (c) identify implications; (d) identify facts; (d) identify relations among concepts; and (e) identify constraints of concepts and relations.

4. ***Verification.*** This phase involves: (a) check if all concepts and relations used in the conceptual structures exist in the type lattice and the type definitions; (b) for each concept and relation that is not defined, either replace it with one that is defined, or add it to the type lattices or type definitions; and (c) make sure that each graph conforms to the constraints imposed by applicable canonical graphs. If not, either the graph is incorrect, or the canonical graphs must be modified.

5. ***Validation.*** In this phase the knowledge specification must be validated against requirements. Execute the knowledge specification (that is, the prototype) with the test cases used in collecting the protocol, and compare results against the expert's analysis. Determine if the knowledge specification gives correct answers for all known validation cases. If not identify missing or incomplete knowledge, and expand/modify the initial knowledge specification. Determine if difficulties exists due to knowledge in the wrong form, and so on.

This research has been used to develop a KBS (using KEE) for an audit task. Two project teams participated in the development - one adopting the above methodology as their development vehicle and the another using the Waterman methodology of evolutionary prototyping. It was found that the former methodology was consistently better than the latter with two exceptions:

- In using conceptual structures in linear form, as opposed to those in graphical form, it was found that experts had difficulty in understanding them.

- Development of the knowledge specification obviously increased the effort required to develop expert systems. However, the authors believe (although this was not tested) that re-implementing the systems from these specifications will take less time and money than improving the prototypes.

Fox (1990) provides support for some of the contentions raised by Slagle et al. (1990). In his exploration of the facts, myths and legends associated with KBS and their development, Fox (1990) makes the following comments (amongst others) [pp. 13 - 18]:

> ***"Legend - rapid prototyping leads more quickly to final solutions.*** *Rapid prototyping has frequently been touted as an approach for constructing solutions more quickly than conventional approaches can. And this is partially true. If the problem fits an application shell ... knowledge gathered from experts can be put into the system quickly and then tested. Consequently the knowledge base can be built incrementally, with verification and validation occurring throughout the process.*
>
> ***Myth - small prototypes can be scaled up into full-scale solutions.*** *Using today's powerful knowledge engineering tools, AI engineers can construct prototypes quickly with "pretty" interfaces. This not only gives managers a false sense that the problem has been solved it also misleads AI engineers. The heart of the problem is whether the problem-solving method used*

> *in the prototype - which solves only a small portion of the problem - will scale up to solve the entire problem. ..."*

That is prototyping may only be an optimal development approach when looking at small applications, and where expert system shell software can be used not only for development of the conceptual prototype but also in the final phase of delivery of the production system. With respect to commercial KBS development, the question therefore arises, is development concerned with only small systems, and have expert system shell software been the only vehicle used to deliver production KBS's?

Other KBS Development Issues

Are Commercial Knowledge Based Systems Expert Systems?

Bobrow et al. (1986) claim that the knowledge acquisition phase of KBS development should not use materials such as textbooks. These authors state that at best textbooks could be used to provide test cases as part of the validation phase, namely [p. 892]:

> *"**Expert Knowledge Is (usually) Not in Textbooks.** One trap awaiting the unwary is the expectation that textbook knowledge is the right stuff for incorporating into an expert system. Textbooks are not bad or incorrect; the problem is the great deal of practical material - obvious to an expert - that never finds its way into textbook. Most textbook knowledge is too idealised: for applications of real interest, only an expert knows the messy but necessary details of real problems and the unpublished rules of thumb."*

Furthermore, these authors claim that textbooks are often designed as merely reservoirs of knowledge, a convenient place to store and then look for facts about a particular knowledge domain. These stores of knowledge may not provide anything in terms of the best strategy to adopt when solving a problem in this knowledge domain, or how to structure knowledge, both characteristics often seen as essential to typing a human being as an expert. For example, prior to the development of DENDRAL [p. 892]:

> " ... 'chemistry books gave examples of the interpretation of mass spectral data, but never said how to systematically enumerate the set of possible molecules. In fact, developing DENDRAL required the construction of some sophisticated mathematical theories not previously known." '

Two questions arise from the above:

(a) what is the goal of KBS development within commercial organisations - "expert" system or merely "knowledge based" systems; and

(b) what are the primary knowledge sources used in the development of these systems - manuals and other written material, or more sophisticated and "intellectual" sources such as those required in the development of DENDRAL.

Turban (1990) has characterised expert systems as those which operate at a level of performance equivalent to recognised experts within a particular field of intellectual endeavour, whilst knowledge based systems do not have this stringent requirement. Fox (1990) claims that in not all cases is KBS development focussed on delivering an "expert system", namely [pp. 13-18]:

> *"**Myth - All expert systems are expert systems.** Contrary to popular belief, there are few pure expert systems. Once an expert's knowledge has been extracted, AI engineers usually identify problem aspects that can be better solved through other problem-solving techniques. As a result, the final system tends to combine expertise (search guided by the expert's knowledge) with other forms of search (unrelated to how the expert solves the problem, but using a large amount of domain knowledge)."*

This discussion leads to some interesting issues that should be considered further. First, within a single commercial organisation, and also across different organisations, what is the principle driving force of the KBS development. The first thing that springs to mind, is that development should (would?) only occur if the organisation perceives some commercial advantage in undertaking the task. That is, the driving force of KBS development has a focus on economic considerations, for example does the system provide a positive return on the investment to be made in the KBS development, or does the KBS provide more benefits to the organisation than it costs. Alternatively, is KBS development undertaken where the economic considerations, although important, are less significant, for example there is a problem that the organisation requires solved and there appear to be no viable alternatives other than to develop a KBS. In the first case, questions of whether the KBS is demonstrating expert level performance or not may be totally irrelevant. Accordingly, if Fox (1990) is correct then the requirements of Brobrow et al. (1986) can be relaxed. That is, for systems which are required to perform at less than the expert level, knowledge acquisition may come from "textbooks" or other forms of documentation generated by the organisation. Alternatively, if Fox (1990) is found to be incorrect, does that still mean that knowledge acquisition for KBS's that must operate at the expert level of performance cannot be obtained from written material?

Verification/Validation and Maintenance of Production Systems

These two issues can be considered linked to a large extent. For example, there may be an inverse relationship between the amount of verification and validation performed and the subsequent amount of KBS maintenance. Obviously, considerations other than mere quantity must also come into play such as the effectiveness of verification and validation, that is these activities are not merely concerned with "proving up" the knowledge base, but looking at all aspects of the system such as the user interface, structure and conduct of consultation sessions. Given that all verification and validation processes have been completed, then it should be expected that, on average, the incidence of maintenance on the production KBS should be reduced. In terms of evaluation techniques, O'Leary et al. (1990) have proposed a three stage process in which a third party controls and directs the validation procedure, namely:

- ***Face Validity:*** is concerned with a step-by-step code review, in which the third party validator deals directly with the expert; the system designer acts as a consultant. In reviewing the code, the validator and the expert focus on the expert's reasoning processes and knowledge. The results of face validity should act as a feedback mechanism to the system designer for prototype refinement, redesign and re-implementation.

- ***Sub-system Validity:*** is concerned with identifying and examining the revised prototype's assumptions and critical procedures, focussing on the prototype's details and specifics, to try and identify areas that may need further development or revision. The first step is to divide the code into modules, which is more difficult for a KBS than for a conventional IS. For example a KBS's module will not be a collection of program statements, but could be a rule, or a set of rules with common inputs, a piece of logic, or a set of procedures that generate specific outputs. As a result of performing this activity overlooked design elements may be discovered, as well as feedback to the system designer for further prototype refinement or even re-performance of earlier phases in the KBS development (such as knowledge acquisition or knowledge conceptualisation).

- ***Input-Output Comparison.*** The ultimate test of the prototype is its ability to supply responses comparable to the expert for certain combinations of events. Numerous techniques are available, such as spectral analysis and other goodness of fit tests, which are highly quantitative. Other techniques such as the Turing test, are qualitative and use subjective comparisons. The second group of techniques while lacking mathematical precision are less demanding and more flexible. Again the system designer would use the results of this phase of the validation process to revise the prototype, or re-perform earlier phases of the KBS development.

Validation of the above paradigm was performed by O'Leary et al. (1990) on the development of an expert system (SCII) that controlled traffic signalling at isolated intersections. Again given the limited validation performed with respect to the procedure outlined above, what degree of similarity exists with respect to the verification and validation procedures adopted by development teams for commercial production KBS's.

Maintenance procedures for commercial production systems is one area that Rook and Croghan (1989) consider will require a lot of attention as KBS development moves from a predominantly laboratory or research type environment into the commercial environment. In the former environment such operational issues are rarely addressed. Rook and Croghan claim there are quite different forces driving development in each environment, that is [p. 587]:

> *"... the ultimate goal of a KBS developed for a laboratory environment is to demonstrate a KBS technology application to a domain of interest in order to advance the KBS technology as well as to provide a foundation for additional technology development work."*

However, within an operational environment [p. 587]:

> *"... the primary goal of [a] KBS is to function as a powerful, problem solving system whose interface inspires confidence and acceptance. The KBS at this end of the continuum must operate day-to-day over a prolonged period of time with minimal maintenance."*

How do commercial organisations ensure that the systems being delivered satisfy these criteria of performance and minimal maintenance. With respect to the maintenance issue, Fox (1990) indicates that another myth is often associated with the maintenance of KBS's, namely [pp. 13-18]:

> *"**Myth - AI systems are easy to maintain.** Using rules as a programming language provides programmers with a high degree of program decomposability; that is rules are separate knowledge*

> *chunks that uniquely define the context of their applicability. To the extent that we use them in this manner, we can add or remove rules independently of other rules in the system, thereby simplifying maintenance.*
>
> *Building rule-based systems differs from this ideal. Various problem-solving methods (including iteration) require that rule implementing these methods have knowledge of other rules, which breaks the independence assumption and makes the rule base harder to maintain. The much-heralded XCON system has reached its maintainability limit (about 10,000 rules). The complexity of rule interactions at this level exceeds maintainer abilities. To reduce the complexity of interactions, attempts have been made to modularise the rule base."*

However, Bachant and Soloway (1989) appear to contradict the above assertion, as the system is still in operation and has undergone significant expansion, both to it's rule base and overall functionality. Accordingly, another interesting question to resolve concerning the development and delivery of commercial production KBS's is the whether maintenance issues are addressed during development, and what sort of maintenance procedures are in place once systems go into production.

Application of Object-Oriented Concepts to KBS Development

From its origins in the work of the Xerox Palo Alto Research Center (PARC) and the programming language Smalltalk, the term "object oriented" seems to have broadened dramatically, so that nowadays there are object-oriented environments, object-oriented applications, object-oriented databases, architectures, and user interfaces, and object-oriented specification, analysis and design methods (Cox, 1990). Accordingly, what does the term "object oriented" mean? The simple answer to this question is that it is not clear - it means different things to different people. For example, McGregor and Korson (1990) state that object-oriented is founded on five basic concepts, namely objects, classes, inheritance, polymorphism, and dynamic binding, with the first three of these concepts appearing within high level analysis and design development phases, and the last two appearing within low-level design and implementation development phases. To a large extent, Duff and Howard (1990) agree with the above statement, however they maintain that the most important features which distinguish object-oriented programming languages from traditional programming languages, are: encapsulation, polymorphism, and inheritance.

How do "object oriented" concepts impact on KBS development? It appears that this impact will occur from two different directions. The first of these can be loosely described as a "technology impact", and the second will arise from the work being done in object-oriented systems development methods, strategies and approaches. With respect to the first issue, the use of different representational formalisms is an example. For instance, although production rules may still be the predominant vehicle for knowledge representation, the knowledge representation using frames embodies many object oriented concepts. Duff and Howard (1990) state because of encapsulation or information hiding, an object will contain both data as well as procedures by which other objects may gain access to that object's data. Accordingly, objects can be seen as "bundles" of data and code. Frames as a knowledge representation formalism contain slots which can themselves contain not only data, or "instances" of an object, but also rules, and procedures. Therefore frames meet two important requirements for objects.

Furthermore, in many frame based formalisms of knowledge representation there can exist relationships between frames, the most important relation being "a kind of". This relation enables the frame system to infer values for unfilled slots by means of inheritance, another central concept within the object-oriented paradigm. If frame X is "a kind of" frame Y, and a slot in X is not filled, then one way of filling that slot is to inherit a value or a procedure from frame Y. With respect to inheritance, another interesting issue to resolve would be whether in KBS implementations inheritance is either single or multiple. Duff and Howard (1990) claim that most OOP languages, including Actor, Smalltalk, Objective-C, and Object-Pascal, allow only single-ancestor inheritance. On the other hand, Meyer (1990) provides the following example of multiple inheritance from the Eiffel libraries [p. 72]:

> *"Inheritance in Eiffel is multiple: a class may have any number of parents. This is necessary whenever a simple tree-structured hierarchy would not provide a satisfactory classification. Many libraries use this possibility. For example, class POPUP_MENU in the Graphics Library inherits both from MENU and from POPUP (describing "pop-up" objects)."*

Another term commonly associated with the object-oriented paradigm is re-use. Meyer (1990) states that with respect to the Eiffel language and associated libraries, that the re-use issue approach is concerned with [p. 71]:

> *"The object-oriented approach to reusability begins with the premise that practical resuable components should be organised around objects (data structures) rather than functions (action structures). This leads to the fundamental modular construct of object-oriented programming ... which is the class, a module built around a data abstraction. A class is a model for a set of data structures."*

An interesting question to resolve would be where within the KBS technology will re-use be most important. For example, would it be an easier matter to re-use inferencing strategies than knowledge bases? Or will re-use focus more on the way a complete KBS solves a particular type of problem, that is a market develops in which KBS's are sold that address common or generic problems found within organisations, say within the asset acquisition area? Or will object-oriented open up new horizons for KBS technology in which problems never attempted before may now be tackled. For example, with respect to this last question, Agha (1990) states that there are three common patterns of parallelism in practice (Athas and Boden, 1989; Fox et al., 1988). First, pipeline concurrency involves the enumeration of potential solutions and the concurrent testing of these solutions as they are enumerated (which has many similarities to the frame-based abduction or hypothesis-and-test cycles of inferencing). Second, divide and conquer concurrency involves the concurrent elaboration of different sub-problems and the joining of their solutions in order to obtain a solution to the overall problem. In divide and conquer concurrency, there is no interaction between the procedures solving the sub-problems. A third pattern can be characterised as co-operative problem solving which involves a dynamic complex interconnection network. As each object carries out its own computational process, it may communicate with other objects, for example, to share the intermediate results it has computed.

Agha (1990) gives the example pipeline concurrency as being the prime sieve. Instead of a linear pipeline with its inevitable bottleneck, a concurrent one in the form of a tree is implemented, with the numbers sent to different identically behaving objects, each testing for divisibility by a given (low) prime, and then merging the results. Divide and conquer concurrency algorithms can often be expressed as functions. In co-operative problem solving concurrency, intermediate results are stored in objects and shared by passing messages between objects. An example of cooperative problem solving is blackboard systems which allows collaboration between agents through a shared work space. In an object based system, the blackboard and the agents may be represented as systems of objects.

The other perspective is the impact of object oriented development methodologies either will have or already has had on how KBS are developed. A number of expert system shell products stated that they have an "object oriented" development environment. Will object oriented KBS development occur only through the use of such software. Alternatively, will less specific work related to object-oriented design methodologies ultimately be used within KBS development. Will these design methodologies follow the approach developed by Henderson-Sellers and Edwards (1990), with its origins in the structured programming and development paradigm, or are they more akin to the Wirfs-Brock and Johnson (1990) model, which adopts a responsibility driven approach to object oriented design?

Development Methodologies and KBS Development

The previous sections of this literature review looked at what commercial organisations will use KBS's for, what structure or components may be included in a knowledge based system, and how these systems are developed. In terms of development approaches, it appears that this field is showing signs of increasing maturity with a movement away from the more ad hoc "rapid prototyping" approach to something with more rigour and discipline. This increasing rigour and discipline should be evident in the use of KBS system development methodologies by KBS development projects.

Accordingly, the next sections of this literature review will consider the application of systems development methodologies to the construction and delivery of production KBS's. There seem to be many questions to resolve in this area, of which the following are but a few, namely:

- Are system development methodologies necessary to the successful development of KBS's?

- Are there products available that could be considered "KBS development methodologies", such as KADS and ES/SDEM? The research literature indicates that both of these methodologies are not merely academic constructs but have been used to develop commercial production KBS's.

- Are KBS development methodologies significantly different to those used for conventional systems development? A comparison will be made between these two methodologies and some of the more mature products that are available for conventional systems development, such as SSADM, ISAC, and Boehm's (1988) spiral methodology.

- Have Australian commercial organisations taken any of these methodologies on board to assist them in building KBS's? Alternatively, have these organisations built their own development methodologies or modified existing methodologies, either proprietary or in-house, to assist them in the delivery of these systems?

However, before meaningful answers can be given to these questions, a better understanding of what constitutes a methodology should be undertaken. As will be evident from the discussion that unfolds below, the first problem in this area is the lack of precision both in the definition of the term "methodology" and its subsequent application. For example, there have been many different products within the KBS field that have the term methodology or method applied to them, but only cover a very limited portion of the entire development process. Should anything that has the label methodology be automatically considered one, or should more stringent analysis and application of appropriate criteria be employed to something before it can earn the label "methodology"? Before answering these questions, another should be attempted, that is how important is the application of a methodology to the successful development of an information system?

The Importance of a "Methodology"

Is a methodology necessary for the successful development of a conventional information system or for a KBS? Alternatively, is a development methodology only a sufficient condition for successful systems development? That is, employing one should ensure completion of a viable system, but not having one does not guarantee that system development will be unsuccessful. With regard to KBS development, in agreement with earlier comments by Hickman et al. (1989), Slagle et al. (1990) and others, de Greef and Brueker (1992) claim that you can't just "prototype out" an effective KBS - it is too complex a system, namely [p. 89]:

> *"It is difficult to develop an operational knowledge based system. Even if one succeeds in modelling and implementing some expertise using some AI paradigm, it may well turn out that the intended users cannot use it or have no use for it in their everyday work. Introducing operational knowledge based systems (KBS) has subtle, but further reaching consequences than automation by conventional systems in professional organisations.*

Conventional systems are largely viewed as semi-passive tools under control and command of the user, whereas a KBS may take the role of intelligent, active agent. Therefore "careful" specification of how the user and the artificial agent should and can co-operate becomes far more decisive in knowledge engineering than it was thus far in software engineering. ..."

Hickman et al. (1989) agree with the above arguments, that complexity and subtlety are inherent features of KBS's. Therefore, given that KBS's are complex systems, then there is more likelihood that viable KBS's would require a development methodology than vice versa. On the other hand, Hickman et al. (1989) also state that where organisations are endeavour to apply the technology for the first time, they usually build stand-alone small scale systems, with an inferencing pattern based on some form of diagnosis, and a development environment that uses an expert system shell. If organisations are still in this "first wave commercial KBS's" condition then it may be that systems development methodologies are considered less important to ensure successful delivery of a production KBS.

If this is the case, commercial organisations (even those within the public sector) may trade off the additional overhead and cost of following a methodological approach against the chance that the product finally delivery does not have as much overall efficacy as it should have had. Whether this will be a sensible trade off is a moot point (Hickman et al. 1989, p. ??):

"Expert systems have tended only to model the expert inferences and not the co-operative aspect of problem solving. The result has often been that such systems are unusable since they simply do not fit into the environment traditionally occupied by the human expert."

Although there has not been a need for a methodology and development has successfully deployed a KBS product, the overall result may be something less than optimal. Depending on how bad the final product is, this may defeat the whole purpose of undertaking a KBS development. More importantly, it may sour the organisation's attitude to the use of this technology in future development situations, where choosing to construct a KBS is the best approach. The devil's advocate would then ask: what are the chances of producing a less than optimal product? Are there other factors that would more than compensate for not following a KBS development methodology?

Taking a more general approach to this question of importance of methodologies, if the amount of effort is any indication of importance to IS or KBS development, then the answer seems to be a resounding "YES". The assumption here is that all this intellectual interest and effort cannot really have been a waste of time!) There are a plethora of system development methodologies (either proprietary or general purpose) available to develop conventional systems. For example, Olle et al. (1991) provides an extensive (but not exhaustive) list of system development methodologies (32, in all). Boehm (1988), in a historical review of the evolution associated with system development methodologies, provides two insights as to why these things are important. First, they can be important, even if system developers are not aware of their importance. During the 1950's "code-and-fix" and other purely ad hoc approaches to system development were being replaced [p. 62]:

> *"As early as 1956, experience on large systems led to the recognition of a non-methodological approach (such as "code-and-fix"), and led to the stagewise model which stipulated software be developed in successive stages, viz. operational plan, operational specification, coding specifications, coding, parameter testing, assembly testing, shakedown, evaluation."*

Further work and refinement on this model led to the Waterfall model which emerged around 1970. It advanced the art in the sense that feedback could occur back to earlier stages rather than merely sequential progression through the stages. Waterfall enjoyed wide acceptance in government and industry as a standard. However, it did have drawbacks in terms of the effort required to conform with the methodology and it lack of suitability with more advanced technology projects. McCracken and Jackson (1982) have claimed that technology also impacted on development methodologies, which led to the emergence of models such as the evolutionary model, which incorporated rapid development tools such as prototyping. In a similar vein, Boehm (1988) describes the transform model, which takes advantage of technology advances such as CASE tools.

System development methodologies, if they are not important, then they have attracted a lot of misguided concern and expenditure of effort. The second insight that Boehm provides, is that system development methodologies may be more important for some people than for others - for instance a novice project manager may think them more important than an experienced project manager. This is because, as Boehm alludes, the methodology provides help with questions the novice project manager may ask more often, such as: (a) what shall we do next?; (b) how long shall we continue to do it?, and many others [p. 63].

With respect to development of knowledge based systems, as already alluded to in Chapter 1, Guida and Tasso (1989) maintain that application of development methodologies to knowledge based systems is a necessary requirement of this technology being accepted within commercial organisations. Rook and Croghan (1989) also point out that a large amount of KBS development to date has occurred within a "laboratory environment" and very few operational KBS exist outside of this environment. Lack of a development methodology within a "laboratory environment" may be all right, but not within the commercial environment. For instance, Rook and Croghan claim [p.587]:

> *"... the ultimate goal of a KBS developed for a laboratory environment is to demonstrate a KBS technology application to a domain of interest in order to advance the KBS technology as well as to provide a foundation for additional technology development work"*

However, within an operational environment, these authors state [p.587]:

> *"... the primary goal of [a] KBS is to function as a powerful, problem solving system whose interface inspires confidence and acceptance. The KBS at this end of the continuum must operate day-to-day over a prolonged period of time with minimal maintenance. ...*
> *Can we effectively transfer a laboratory-based KBS prototype into a field-deployable one through additional KBS design and development ?"*

These authors conclude that answering this question is not simply "YES" or "NO". There will be some aspects of KBS development such as knowledge base architectures, and the inference mechanism that possibly will not change with the transition, while other aspects such as knowledge representation, both in terms of the content and the way the knowledge is stored, that may have to be altered substantially. However, their contention is not supported by Zawa (1987) who states in her reports that the major aim of developing the Personal Loan Assistant (PLA) was as a technology demonstration. That is, one of the major objectives behind developing the PLA KBS was to introduce this new technology into the State Bank of NSW in order to determine its feasibility before extending the use of KBS's into other areas of the organisation. If the focus within commercial organisations is still on the introduction of the technology through technology demonstrations, then use of a development methodology or not becomes an irrelevant question. However, even within small scale developments, it is considered that it would still be appropriate to find answers for the following questions:

- Are methodologies required to complete system development and produce robust, high-quality systems?

- What overhead does the adoption of a system development methodology place on the overall development effort?

- Does a methodology significantly improve communications with and understanding by end users of the system development processes?

- How critical is using a methodology to system development success, that is completing system development, or can other more critical factors be recognised across system developments?

With respect to this last question, Bubenko (1986) offers the following interesting insight [p. 298-299]:

> *"Methodologies always prescribe tasks to be performed and their sequence [or priority]. However, very few methodologies offer guidelines how to perform the various tasks and, more important, how to determine the quality of the design and analysis product. For example: what characterises a "good" [human] activity*

> *diagram or a conceptual schema? We should realise that a design always will have an artistic component and that not everything can be "prescribed". But the situation today is largely that almost no criteria exist and the quality of a design is totally dependent on the competence of the designer to the extent that one sometimes wonders about the utility of the methodology at all."*

So even if methodologies are of assistance to the novice project manager, there are often plenty of "holes" within methodologies that require innate skill and insight to ensure system development success. It would appear from the above that the importance of using a methodology to develop KBS's is very much unresolved.

Research Issues: System Development Methodologies

Another question that needs to be resolved is to determine what the term "methodology" actually means. It would appear that as has been found in other sections of this literature review, the answer will not be just a simple definition. Rather than being a situation where a product is either a methodology or not (fits all the criteria in the definition or not), it may be more appropriate to evaluate these products on a continuum where individual products are considered to have more or less of the attributes of a methodology. For example, at what point along this continuum can it be said that a technique which performs many of the activities within system development becomes a methodology?

Other issues outside of the purely definitional ones may also be more apposite. For example, can system development methodologies be nested inside of another methodology, or do these things stand independently or one another? If the former assertion is true, then when a methodology is nested inside another, is this nested methodology still perceived to be a methodology, or does it become a lesser thing, say a mere technique. The classic example of this situation is of course prototyping, which is often nested within system development methodologies, such as Boehm's spiral; but can also be used as the primary development activity to develop complete systems.

The continuum model appears to be the better approach, as although different methodologies have many things in common, there has not been one dominant methodology applied by different individuals or by different organisations to the activity of system development. Bubenko (1986) in his review of development methodologies over the last thirty years claims, [p. 296]:

> *"It is a reasonable estimate that up to now hundreds of more or less similar methodologies have been published. In practice, probably tens of thousands of more or less different approaches are being used."*

Floyd (1986) also supports the notion of a plethora of development methodologies, with another confusing dimension in that each methodology has its own underlying philosophy, its own conceptual framework, and its own claims of general and superior applicability to this activity when compared to other development methodologies. The academic and research community has not been tardy in the development of tools, techniques and whole system development methodologies that the creators consider applicable to commercial developments. However, with respect to this point, Bubenko (1986) claims [p. 296]:

> *"Among the methodologies published, the great majority are developed in academic research environments. A very small fraction of these methodologies have ever been applied to practical cases of a realistic size and complexity. The acceptance of "academic" methods in practice is low, and in general, the rate of transfer of research results and "know-how" from scientific research to industry is embarrassing slow."*

Avison and Fitzgerald (1988) also refer to a number of different classes or categories of methodologies, namely: (a) commercial methodologies (that have evolved from practice); (b) theoretical methodologies (which often started life as research projects in universities or research institutions); and (c) blended methodologies (which combine a number of theoretical bases, such as SSADM which combines entity modelling techniques with data flow diagramming techniques). Is this wealth of different products for conventional systems development also evident in the KBS field? Furthermore, are the conclusions reached by Bubenko for conventional system developments also applicable to knowledge based systems development? Some people may argue that the objective of a lot of research is not to produce commercially viable products but has a focus of trying to gain a better understanding of the thing being researched. However, an "academic" methodology that has no validation in practice would seem to be the same as a disaster recovery or contingency plan that has not been tested, that is it represents a theory more than a plan. In terms of methodologies, should it be a necessary requirement that before something gains this label, it has been applied in a practical situation and found to be effective, that is more than just a mere theoretical exercise or thought experiment?

What Do We Mean By The Term "Methodology"?

In many situations to determine what a term means, the first port of call is the dictionary. However in this case, as was alluded to above, dictionary definitions don't provide a lot of help. Nevertheless, it is a good starting point, and the Oxford English Dictionary (1989) offers the following definitions for the words that often appear with respect to constructing information systems:

- **Method:** Pursuit of knowledge, mode of investigation. The dictionary then goes on to provide a wider definition of this term, namely: a way of doing anything, especially according to a defined and regular plan; a mode of procedure in any activity or business; a scheme, a plan of action.

- **Methodology:** The science of method, "methodics"; a treatise or dissertation on method; a systematic classification in the natural sciences; the study of the direction and implications of empirical research, or the suitability of the techniques employed in it. Interestingly, the dictionary then goes to say: "**In some contexts weakened to mean little more than "method"**.

Therefore, should we be talking about system development methodologies or system development methods? Looking at these dictionary definitions, it would seem that the latter is more appropriate. Obviously, in terms of system development, the word "methodology" is used in its weakened form, that is what is meant is a method of developing a system. But this is just so much casuistry; irrespective of which term is correct, from a practical sense these definitions do not provide a better understanding of this concept, as they understandably lack precision.

From a theoretical point of view, Avison and Fitzgerald (1988) provide the following definition [p. 4]:

> *"A methodology is a collection of procedures, techniques, tools, and documentation aids which will help the systems developers in their efforts to implement a new information system. A methodology represents a way to develop information systems systematically. A methodology will consist of phases, themselves consisting of sub-phases, which will guide the system developers in the choice of the techniques that might be appropriate at each stage of the project and also help them plan, manage, control and evaluate information systems projects."*

This definition provides great assistance in obtaining a better idea of this term. First, a methodology is not just a loose collection of best practices built up on an ad hoc basis through experience in system development projects, but has had more consideration in their construction. That is, there is a difference between a project manager adopting a methodical approach to system development, by carefully applying best practices garnered from experience, and a project manager developing an information system using a methodology (which by definition will be methodical). Second, to be complete in a practical sense, a methodology has to be more that a good set of ideas, however clearly expressed (McDonald, 1986). For example, Information Engineering is not only carefully documented and explained, but has additional support, namely training courses, interactive video tape material and detailed case studies. These aids help to ensure that the methodology is used effectively in systems development.

The above supports the contention that the term "methodology" is one that embraces a breadth and depth of function and meaning that is beyond the capability of simple dictionary definition to adequately describe. It would appear that to be a methodology, this product should have the following features:

- The SDM should produce an end result which in this environment is an information system, and accordingly should cover most of the development process.
- As a plan, an SDM should present a framework within which the system development can be adequately estimated, planned and managed, for example use Ghannt charts, critical path method, identification of milestones, definition of deliverables and structuring of reporting arrangements should all be documented.
- As an assistant, an SDM provide advice, particularly to the novice project manager, on selection of appropriate tools and techniques that will expedite the technical side of the system development, for example data flow diagrams, entity relationship models found within structured systems development methodologies.
- As a trainer or teacher, an SDM ensures users understand all processes contained within it, and the relationships between these tasks or processes through clear documentation as well as additional support, such as training courses, interactive video tape material, and detailed case studies.

Moving from the theoretical to the practical, what semantic meaning have practitioners in the field of IS development given to this term? Hackathorn and Karimi (1988) offer the following comments on method, methodology, tools and techniques [pp. 207]:

> *"... a method should be a solid basis for explaining its approach, major issues,, relationships among variables, and expected outcomes. Having such a conceptual basis, the method has more consistency and stability in the* [software development] *industry. On the other hand, the practical side of depth dimension focuses on tools for actually performing the method, considering issues of useability and efficiency.*

The "depth dimension" referred to above basically consists of three levels nested within one another, that is a methodology contains techniques which themselves contain tools. For example, the goal of structured design (Yourdon and Constantine, 1979) is to develop program modules that have three important properties: (a) modules are relatively independent; (b) existing dependencies can be easily understood; and (c) there are no hidden interactions between modules. Hackathorn and Karimi (1988) claim that structured design provides a set of techniques for producing modules with these properties, but there are no automated tools for creating these modules. Accordingly, the absence of tools or even techniques from a "candidate" methodology may not prevent it from being considered one. One thing that should be resolved is whether methodologies can be distinguished from other things that should possess labels such as "procedures", "tools", and "techniques", which are considered to be less than actual methodologies.

Distinguishing Between "Tools", "Techniques" and Methodologies

From the above, tools and techniques will feature in most methodologies, and many will be common to different methodologies, but by themselves they should not be considered a methodology in their own right. Tools and techniques are normally associated with particular tasks or activities to be performed within the system development. Accordingly answers to the following questions are necessary. Within a development methodology, what constitutes a tool? What is a technique? What is the relationship (if any) between tools and techniques? Can a tool or technique be both part of a larger methodology, as well as a methodology in their own right?

Again, in this situation, dictionary definitions of these terms seems to be of little assistance, namely (Oxford English Dictionary (1989) :

- **Technique:** Manner of artistic execution or performance in relation to formal or practical details (as distinct from general effect, expression, or sentiment); the mechanical or formal part of an art, especially of any of the fine arts. The dictionary then goes on to say: "*Loosely* - a skilful or efficient means of achieving a purpose; a characteristic way of proceeding."

- **Tool:** Anything used in the manner of a tool; a thing (concrete or abstract) with which some operation is performed; a means of effecting something; an instrument.

A CASE tool which is applied regularly to generate code within a system development could, in these circumstances, be perceived as a "characteristic way of proceeding". On the other hand techniques can be either concrete or abstract things with which "some operation is performed" in the system development. If these definitions are appropriate then these terms are to a large extent interchangeable. That is, there is no boundary between what would be considered merely a tool or a technique used in system developments, and methodologies. However, the research literature does not adopt this assumption. For example, Hackathorn and Karimi (1988) define a technique as [p. 208]:

> *" ... 'a procedure for accomplishing a desired outcome'. In particular, a technique specifies the steps in performing the IE activities, as well as the necessary inputs and results from each step. A technique deals with the logical way of "how" to do an activity and represents knowledge more than actual products."*

whereas on the other hand a tool is defined as [p. 208]:

> *" ... 'an instrument for performing a procedure'. In particular, a tool is some tangible aid ... The objective of using a tool is to produce a deliverable."*

Avison and Fitzgerald (1988) agree with this assertion that there is a fundamental difference between a tool and techniques, and offer the following example [p. 9]:

> *"A non computer-orientated example may help. Two techniques used in the making of meringues are (1) separate the whites of eggs from the yokes and (2) beat the whites. The methodology may recommend the use of tools in these processes, for example an egg separator and a whisker. In this text, tools are usually automated, that is, computer tools, usually software to help the development of an information system. Indeed, some of these have been designed specifically to support activities in a particular methodology. Others are more general purpose and are used in a number of methodologies."*

To summarise, it would appear that techniques are more general specifications of how to do things, and so may contain one or many tools, whereas as tools are detailed specifications of how to do things. The detail of how to proceed in a tool (which at times will be encapsulated within a piece of computer software) will mean that the overall application of the tool is no different across different system developments. That is, developers are constrained by the tool, which should be extremely stable across system developments and system developers. On the other hand, since a technique provides overall guidance in how to perform or compete a task, and does not specify precisely how to complete the task, application of the technique could vary slightly between different systems developments and across different system developers.

The research literature concerning KBS development, indicates that tools and techniques are available to assist in the development process, particularly during the knowledge acquisition (KA) phase of system development. For example, Marcus (1988) has reviewed six different knowledge acquisition tools MORE (Kahn et al. 1985), MOLE (Eshelman, 1987), SALT (Marcus, 1987), KNACK (Klinker et al., 1987), SIZZLE (Marcus, 1988), and RIME (Bachant et al., 1984). As Marcus (1988) indicates the tool can relieve system developers of some of the manual work required to construct a KBS. For instance, the tool may not only automate knowledge acquisition and knowledge representation (depending on the sophistication of the tool), but the KA tool can also be used to develop a well chosen set of sample problems and thus help to assess the validity of the KBS. However, tools in being specific do not have the more general application of a technique: a tool will not have the capability to handle broad, general or undirected questions during KA such as "How do you perform your job?". Furthermore, each of the tools mentioned above can operate effectively in a limited problem solving area, for example KNACK is specialised for tasks that require a report as their output, whereas SIZZLE builds expert systems that perform quantitative sizing tasks.

In order to provide a better understanding of the difference between tools and techniques, the area of knowledge elicitation and acquisition has also produced a number of techniques to assist KBS developers with this task. For example, Klein et al. (1989) have outlined a technique known as "Critical Decision Method" (CDM). Here the perennial terminology problem emerges once again: should this artefact be more appropriately labelled the "Critical Decision Technique" rather than "Critical Decision **Method**". On the other hand, it would appear that this artefact has many of the characteristics or criteria that we would associate with a methodology. For instance, CDM has an underlying philosophy in that it is based on the work of Flanagan (1954]); CDM it tries to identify non-routine (or "critical") events, and how these are coped with by the expert(s). Furthermore, CDM also has a raison d'etre, in that it allows KA to be performed more effectively. By concentrating on non-routine cases, CDM will not only incorporate general knowledge that is widely known, but also far deeper knowledge into the KBS. Klein et al. (1989) also indicate that the technique has broad application within KA, covering problem domains such as how to expert pilots fly aeroplanes to how expert fire fighters put out fires in office buildings. Finally, in terms of structure, CDM consists of the following steps [pp. 465-467]:

- **Step 1 - Select non-routine incidents** and probe for information that goes beyond general knowledge and procedures required to perform routine tasks.
- **Step 2 - Obtain unstructured incident account**, an uninterrupted account by the "expert" of the event with interjections only to gain clarification.
- **Step 3 - Construct incident timeline**, that is reconstruct the account provided in step 2 into a timeline establishing the sequence and duration of each event, which can be either objectively verifiable or subjective thoughts and perceptions.

- **Step 4 - Decision point identification**, from the previous step, specific decisions should have been identified which require further probing, for example a point at which a number of courses of action could have been taken and the "expert" selected one of them.
- **Step 5 - Decision point probing**, at each decision branch, probes about options should be asked, both for those actually considered as well as those that existed but were not considered.

Should this artefact be labelled a technique or a methodology in its own right? We consider that the former rather than the latter term is the most appropriate for two reasons. First, although the artefact has broad application in KA, in terms of other phases of the system development it offers little (in fact it offers nothing). Therefore without regard to producing an end result, this technique does not in itself produce a KBS. A methodology should have a broader spread than merely one phase of the system development process. Second, Klein et al. (1989) claim that it is very difficult to validate the above theory, for two reasons:

(a) being a semi-structured interview method, the exact circumstances can never be recreated; and

(b) once a person is interviewed, the interviewee's memory for the event will alter to some unknown degree.

That is, an underlying reason for existence may be hard to justify, and a methodology should be able to justify its existence.

Another technique that can be used to improve knowledge acquisition is the Knowledge Acquisition Activity Matrix (KAAM) (Rook and Croghan, 1989). These authors also call their approach a methodology, and like CDM this artefact also has many similarities to a methodology. For example, KAAM structures is activities around a conceptual knowledge acquisition framework which draws from the work of Hall (1969) and Sage (1977). This framework seeks to address questions such as:

- What are the goals of KA in the requirements analysis phase of KBS development?
- What are constraints to KA during the KBS development?
- Are specific KA steps inappropriate for certain KBS development phases?
- How does one know what the differing role of knowledge acquisition is throughout the system development life cycle?

KAAM also has a reason for existence as Rook and Croghan (1989) claim that KA will be performed more thoroughly because KAAM: (a) identifies specific steps or tasks involved in each phase of the KBS development process; (b) specifies goals of each knowledge acquisition step; and (c) identifies constraints or barriers to successful knowledge acquisition activities. Finally, the artefact has structure in that it consists of the following phases [pp. 590-591]:

Knowledge Framework Specification: specification of the knowledge domain or the subject of the knowledge, for example knowledge about the configuration of a computer system.

Knowledge Resource Identification: specification of knowledge resources or sources of expertise related to the knowledge framework, such as specific human experts or documentation.

Macro-knowledge Extraction: extraction, analysis, and integration of knowledge about the previously identified knowledge framework, for example macro-knowledge would consist of general domain descriptions or identification of major information processing nodes.

Knowledge Partitioning: partitioning of high-level conceptual macro-knowledge into knowledge modules, providing a framework for the detailed knowledge that will eventually comprise the KBS's knowledge bases.

Micro-knowledge Extraction: identification, acquisition, and specification of detailed knowledge such as heuristics, algorithm designs, and evidential reasoning schemes.

Knowledge Formalisation: translation of micro-knowledge into specific knowledge representation schemes, for example rules, frames, object-oriented representations, ready for software encoding.

Knowledge Encoding: software implementation of the knowledge base data structures resulting from the knowledge formalisation step, for example the pseudo-language rules are encoded in the KBS-specific rule representation languages.

Again, as with the CDM artefact, the question could be raised as to whether or not KAAM should be considered a methodology. As with CDM, it is considered that the above is a technique, for the same initial reason, that is a lack of coverage across the entire system development process. One conclusion that flows from the above discussion of tools and techniques is that both of these would reside on the system development continuum - where at one end there is complete and undirected adhocracy, while at the other is use of a system development methodology.

It would seem that techniques are closer to the methodology end of the continuum than are tools, and between the two techniques discussed above, that KAAM is closer to an actual methodology than CDM.

Methodologies: More Than Just Collections of Artefacts

The above discussion hopefully provides a better idea of what a methodology should contain and how the contents should be applied. However, in addition to "what" and "how", the methodology should also provide assistance on "when" and "where". With respect to this point, it is worth returning to the definition offered by Avison and Fitzgerald (1988) for further consideration. First, it is important to realise that the definition must be seen holistically, that is a methodology should have all of the characteristics outlined above and not just some, or a majority of them. In other words individually these conditions are necessary conditions but are not sufficient. For example, taking merely the first sentence, all project managers and other participants would have a collection of "procedures, techniques, and tools" - some of which may be stored on magnetic media, or documented in some other fashion.

However, mere possession of these things in some written form should not mean that a person or organisation has a systems development methodology. Using the above definition, the answer to this question is "NO" as inferred in the second sentence of the definition, that is the collection of best practices has to be applied "systematically". Because, other than the discipline of the individual project manager, there is nothing about a collection of best practices which would ensure that they will be applied consistently by the same project manager across different developments projects, or by different project managers across different system development. More importantly, as was discussed previously, care must be taken to differentiate between a methodical application of procedures and practices, as compared to the application of a methodology's procedures and practices.

On the other hand, systematic application of procedures and practices is a necessary condition. A methodology should not contain only a list of items ("what" to do), and instructions on using these items ("how" to do) that guide systems developers, but also guidance on the overall structure of the development effort and a framework within which tasks will be performed should be provided.

Unfortunately the Avison and Fitzgerald (1989) definition does not provide us with any guidance on how much system or framework is to be provided along with the collection of best practices to allow the artefact to be considered a methodology; or how many times the methodology has to be applied before it is considered to have been used "systematically". Is once enough, or does it have to be applied many times? In the KBS area, have organisations had the opportunity to develop and apply their "KBS development methodology" more than once? If it has been applied only once, does used "systematically" mean that there is a threshold in terms of the proportion of activities carried out according to the methodology, that is all the early activities and phases of the system development, more than half of the activities and phases of the development, that occur within a system development, for the whole of a system development, and so on. In terms of gaining a clearer idea of what "systematically" means, Avison and Fitzgerald (1988) provide the following objectives for system development methodologies [p. 5]:

1. Accurately record requirements for an information system.
2. Provide a systematic method of development. A methodology will usually consist *of phases,* themselves consisting of *sub-phases,* which will guide systems developers in choosing

appropriate techniques suitable for completing tasks at that stage of the project. A methodology should help developers plan, manage, control, and evaluate development of the IS.

3. Provide an information system within an appropriate time limit and at an acceptable cost.
4. Produce a system which is well documented and easy to maintain.
5. Provide an indication of any changes which need to be made as early as possible in the development process.
6. Provide a system which is liked by those people affected by that system.

There appear to be two underlying notions associated with this set of objectives. One is that the term "systematic" has connotations of rigour and discipline, for example "accurately record requirements", "provide an information systems within an appropriate time limit", produce a system that is "well documented". The other notion is decomposition, that a large problem such as developing an information system should be broken down into smaller problems which are assumed (because they are smaller than the original problem) to be more manageable. Accordingly, "systematic" appears to mean rigour, discipline and a logical decomposition of the broader task. Even the CDM and KAAM techniques discussed above, were disaggregated into smaller work units or tasks.

Hickman et al. (1988) support the above ideas, but give additional content to the notions of rigour and discipline. A development methodology should ensure that down stream changes to a system are minimised. Anybody who has practical experience in the development of conventional systems will testify to the fact that the effect of changes is greater (in terms of over time and over budget) the further one is into the development of a system.

Hickman et al. (1988) claim that the structure or framework of the methodology should minimise these problems in three ways, namely:

- **Disaggregation.** The overall system development process is divided into simpler, separate, but interdependent phases.

- **Exhaustive.** The methodology should be exhaustive in its contents, and there are mechanisms to ensure that all activities are performed. Floyd (1986) describes this as "coverage", that is the methodology should support all tasks relevant to a method's view of systems development, and considers that Jackson System Development methodology (JSD) and System Analysis/System Design (SA/SD) to both have coverage.

- **A framework for development.** The methodology provides guidance on when and where activities should be completed. For example, it would not be correct to think about issues of integration of the KBS with other systems, at the end of the design phase of the development - this activity should more correctly be tackled much earlier on.

- **Control mechanism.** Rules and guidelines for traversing the life cycle to ensure that phases have been completed and that consistency of effort between phases with the same development, and across different developments occurs. These rules and the framework above also allow interested parties to measure various factors, such as the progress of system development, and therefore activities to be invoked to correct problems.

- **A toolbox.** A standard set of tools, and techniques to support the developer in his task. What has proved to be a good procedure in earlier KBS developments is absorbed into the methodology and then applied to later developments. When this point is coupled with the proceeding one, essentially we should get better at developing KBS's the more KBS's we develop.

Even the mere possession of a framework or ordering of tasks may not be enough to really be considered "systematic". Floyd (1986) states that the framework should have "coherence", that is the guidelines and other contents within the methodology should be related to each other in a convincing manner, usually based on an overall strategy. For example, Floyd (1986) considers the Jackson System Development methodology (JSD, McNeile, 1986), and Structured Analysis and Design Technique (SADT) to be coherent, whereas Systems Analysis/Systems Design (SA/SD) to be incoherent as there are only weak connections between its Systems Analysis and Systems Design components.

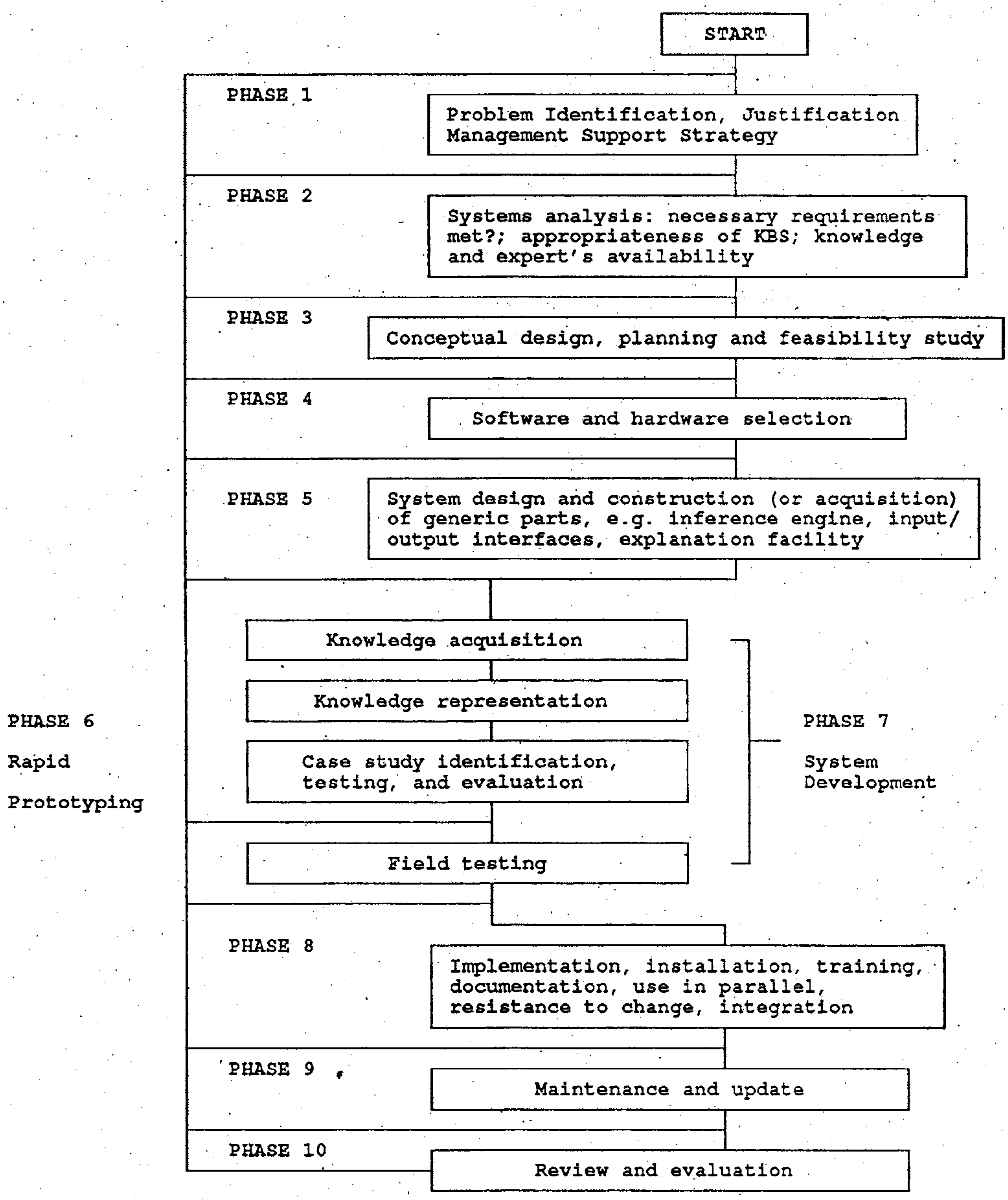
Accordingly, there seems to be a synthesis or common ground coming from the research in this area. First, an artefact should only be considered a methodology when not only does it prescribe (Floyd (1986) also stated that a methodology should be prescriptive) the "what" and "how" in terms of procedures, tools and techniques, but also the "when" and "where" these techniques, tools, and be used procedures are to. The combination of the "what" and "how" with the "when" and "where" should be in some sort of framework that has adequate coverage (either most of the system development process or more stringently all of the system development) and is coherent (the order has meaning, possibly explicated in the rules and control mechanisms incorporated into the framework).

How Systematic Should a Methodology Be?: An Example

Therefore, in terms of how systematic should a methodology be, the answer appears to be rigour, discipline, disaggregation, coverage and coherence. That is, more than looking at the system development process, defining boundaries on between tasks that allow the larger process to be broken down into smaller one, and then structuring and ordering of these sub-processes, is required, an underlying logic must exist. In this section, these ideas are shown to have weight when used to analyse what purports to be a KBS development "methodology" to that has emerged from the research community.

Turban (1990) has outlined a development methodology, which follows essentially a structured approach in the early phases of the KBS development, and a more iterative or prototyping approach during the system design phase, as indicated by the diagram shown below [p. 645]:

Figure 4. Schematic of a KBS System Development Methodology



(Source: Truban, 1990, p. 645)

Obviously, on the surface, the above diagram indicates that this methodology has rigour, discipline, disaggregation, and coverage. That is, disaggregation is there because the process has been divided into ten phases. Rigour and discipline are there in that certain tasks must be completed before others.

Coverage is evident in that the whole of the development process has been delineated from the initial processes of KBS development to maintenance and review of the production system. However, this methodology lacks coherence, and a methodology should be more than a mere arrangement of activities and tasks. For example, some activities that are outlined later in the development process should be included in earlier phases. For example, phases 2 and 3 appear to have some overlap in the fact the phase 2 considers the "appropriateness of KBS", while phase 3 is the "Feasibility study", in which one would need to consider technological feasibility. Furthermore, in order to effectively complete either phase 2 or phase 3 a better understanding of the problem should be gained, which would mean that some knowledge acquisition and knowledge representation (phase 6 or phase 7), should be completed. There is no link of these later phases with the earlier ones.

There are also problems with later phases in this "KBS development methodology". For instance, there is no distinction (since the activities are the same) between phase 6 "rapid prototyping and phase 7 "Systems development". With respect to phase 4 "Hardware and software selection", this phase may either not occur (required to use existing hardware and/or software), or may occur later in the system development, say after the first or second iteration of the rapid prototyping/systems design phases. With respect to phase 5, if the development environment is an expert system shell, then "System design and construction (or acquisition) of generic parts" such as the inference engine would not occur. Furthermore, to construct the user interface and the explanation facility prior to the rapid prototyping and system build tasks (phases 6 and 7) would seem illogical at the least. Accordingly, it would seem that more work on this "methodology" is required before it qualifies as a suitable candidate. That is, on the system development continuum, the Turban approach to building KBS's would be closer to the methodology end than the completely ad hoc end, but is still not quite there.

Methodologies: A Basis in a Philosophy of Development

It is considered that order and system by themselves are still only necessary conditions to identifying a systems development methodology and not both necessary and sufficient. Are there additional things that should be considered?. Avison and Fitzgerald (1988) also state that a systems development methodology should have, as a foundation to the methodology, some basis in the theory of systems development. For example, it could be based on a process oriented theory of system development such as the ISAC (Information Systems Analysis and Change) methodology, and SSADM (Structured Systems Analysis and Design Methodology). Alternatively, this theory could be based on a data oriented view of systems development, such as is the case for the Information Engineering (IE) development methodology.

Floyd (1986) agrees with this contention, stating that methodologies should have an underlying theory or basis. This theory can be derived from the world of mathematics or logic, such as the JSD methodology, or alternatively on some world view or understanding what system development is, how it should be modelled, such as SADT which perceives system development being based on a process of communication and co-operation. In this fuzzy area, Floyd (1986) claims [p. 31]:

> *"... our understanding of the nature of system development as a whole is haphazard at present and tends to be based on opinions and individual experiences rather than on systematic empirical researches. In my view, the lack of a suitable theory about systems development as a whole explains many shortcomings in the existing methods."*

Finally, in addition to being based on some coherent theory of system development, a system development methodology should have possess a certain amount of logic in its own right, that is it should possess a certain amount of raison d'etre which justifies its existence. That is, the methodology should in some fashion provide a better way of developing information systems than using nothing at all, say a purely evolutionary ad hoc approach. If a methodology cannot do this, then the purpose for its existence must be severely questioned.

Assessing a "Methodology" - as a Methodology

Some research has been conducted on how to perform this assessment. Floyd (1986) claims that assessment is hard because there is no ultimate definition of "method". Avison and Fitzgerald (1988) claim assessment is hard because of the variation how the product is communicated - sometimes as a detailed manual which covers all of the development process and requires an army of personnel to perform all the specified tasks, while others are merely a short pamphlet, which may not even have any specified tasks. The continuum approach to assessment of a methodology provides answers to both of the above. For instance, a continuum approach is analog rather than digital - an artefact can be almost a methodology rather than being one or not. With this approach an accurate definition of what is or what is not a methodology has reduced relevance and importance to the assessment process. With respect the second point, it would seem reasonable to place the short pamphlet explanation more towards the ad hoc end of the continuum and the more detailed manual at the other. As was indicated above, tools, techniques as well as the development approach outlined by Turban could all be placed on this continuum.

Other research efforts have been directed towards developing checklists of what methodologies should contain. In this vein, Maddison (1983) and Catchpole (1987) have defined quite extensive lists of features that development methodologies should possess. However, large checklists have two problems. First, they become subjective in determining what is and what is not a methodology, unless all features must exist in the product - in which case the possibility that no candidates pass the test is obviously increased. Second, less precision is available to compare and contrast between two artefacts that have been subjected to the checklist and comparisons are looking more at detail than the larger picture. Even Maddison (1983) saw this flaw, and so reduced his checklist six key questions. A similar process was performed by Avison and Fitzgerald (1988) who took Maddison's 100 features and condensed these down to the following headings [p. 278]:

(a) Does the methodology cover all aspects of the systems analysis and design process from planning to implementation?
(b) Are the steps well defined?
(c) Is the methodology data or process orientated?
(d) How are the results at each stage expressed?
(e) To what *types* of applications is it suited?
(f) Does it aim to be scientific or behavioural?
(g) Is a computer solution assumed? What are the other assumptions made?
(h) Who plays the major role - the analyst or the user?
(i) What built-in controls are there to evaluate the success of each stage?
(j) Is the methodology simply an attempt to link a number of techniques and tools or does it have its own philosophical base?

The checklist approach has also been adopted by Bubenko (1986) and Floyd (1986) who used the initial work of Mathiassen (1982). With respect to these other checklists, Bubenko (1986) has offered the following criteria, [pp. 305-306]:

> *" ... we could assess a methodology by examining to what extent it includes facilities for*
>
> 1. *tutoring the methodology*
> 2. *supporting the "early phases" of information systems development by providing languages and concepts to define and describe organisational goals, problems, activities, interactions, etc.*
> 3. *guiding the user and/or designer through the development process, providing facilities for analysing designs, suggesting improvements, giving criteria for "good designs", etc.*
> 4. *supporting the management and the control of the design process*
> 5. *model management: maintaining the large number of, formal and informal, specifications and models on various aspects of the UoD* [**Universe of discourse**]
> 6. *Modelling support: assisting the users in conceptualising a particular UoD, assisting in analysing the models for completeness and consistency, assisting in model design, redesign/restructuring, and in model integration."*

An Assessment Framework for Development Methodologies

What can be deduced from the discussion in this section? Is it too hard to determine what is or what is not a systems development methodology? The lack of a common underlying philosophy, and the consequent large variety of methodologies means that there is great difficulty in defining the term, and difficulty in making comparisons. Large checklists such as those proposed by Maddison (1983) and Catchpole (1987) do not provide any greater insight into determining what is and what is not a methodology as well as being cumbersome vehicles for comparisons between methodologies. In terms of checklists of features, the reduced checklist approach proposed by Avison and Fitzgerald (1989) seems more reasonable. Furthermore, if a reduced checklist approach is married to the system development continuum then the need to determine what is and what is not a methodology becomes less of an issue.

Some products will just have more features of a methodology than others. In fact, techniques such as prototyping, CDM, and KAAM could be subjected to this process to determine where they lie on the system development continuum as well as "methodologies". Furthermore, combining both of these features constitutes a model which delineates a reasonable assessment mechanism. Although this assessment process may not definitively determine what is or is not a system development methodology, it will at least allow these artefacts to be positioned with reasonable accuracy on the system development continuum. The checklist feature of this assessment process incorporates the following major characteristics that a methodology should possess:

- ***WHY:*** The methodology should be based on a theory or philosophy that provides a foundation and a common thread to the methodology.

The methodology should justify its existence, that is either explicitly or implicitly showing that it offers something substantially better than merely performing a collection of ad hoc tasks. Some form of validation or verification of this claim should also exist.

- ***WHAT:*** The methodology has an end result which in most cases should be an information system (or a knowledge based system). That is, the methodology should cover all aspects of the development process from planning to implementation.

  The methodology should contain aids, for example techniques and tools that can assist both novice and experienced participants in performing systems development tasks. In addition, descriptions of activities to be undertaken in the development should be provided.

  The methodology should provide guidelines on management practices associated with systems development, such as duties of system development personnel, reporting requirements, milestone deliverables, and guidelines on other operational or management tasks, such as feasibility assessment.

- ***WHEN:*** The methodology should possess a framework or hierarchy, based on the underlying philosophy that decomposes the overall development into a number of phases or stages, which in turn may themselves be decomposed into activities and tasks within activities.

The methodology's framework should cover the whole of the system development process, or as a minimum cover all critical phases within the system development.

There should be an underlying logic or coherence to the framework of tasks adopted by the methodology.

The framework should, in some fashion, indicate critical stages of development, such as points of major review, authorisation to proceed with development, and so on.

- ***HOW:*** The methodology should clearly describe all system development tasks, such as necessary inputs and required outputs from the task.

  For each tool and technique outlined, the methodology should provide information on how best to apply this tool or technique to particular development environments.

  The methodology should be clearly documented, and where possible other forms of support, such as training courses, case studies and so on, should be provided.

- ***IF:*** The methodology should provide control in terms of outlining the interdependencies between tasks, conditions where certain techniques or tools may be less effective, and a list of trigger events and solutions to previous problems encountered in using the methodology.

Weighting and Scoring of Assessment Characteristics

As stated above, this assessment framework is used to place a methodology on the system development continuum. The approach will use a novel and simple scoring procedure for the characteristics shown above. Accordingly, two issues need to be addressed, namely: (a) what weight should each characteristic have in relation to all the others; and (b) how is a particular characteristic to be scored across different candidates - in order to ensure consistency. The following comments are not seen to provide the final answer as it is considered that this area presents a rich source of further research. In other words, the development of the assessment framework and the weighting and scoring processes covered in this report are seen as a beginning to further research, discussion and debate rather than a final answer to the vexed question of assessment and comparison of system development methodologies. Notwithstanding this reservation, the following comments are still considered apposite in the weighting scheme outlined below. First, any decision made initially will necessarily be largely subjective, although the assessment framework will be validated using five system development practices (discussed in the next section) in order to determine the viability of the framework, the weighting of characteristics, and the scoring processes. It is considered that across time, as the assessment framework is applied to a larger number and broader range of "products" (methodologies), the results of these assessments should lead to a greater refinement of the weighting scheme and the scoring process which would lead to an evolution of the overall assessment procedure into something more objective than what it is now.

Second, in terms of assigning the initial weights all of the characteristics mentioned are considered important, and so one characteristic should not dominate the overall assessment process. Third, as all methodologies are assessed under the same framework an absolutely accurate weight for a characteristic is not critical, however there should be a firm and reasoned argument to support the relative weight given to that characteristic.

Finally, the scoring process initially will rely on a comparison between the different products; however, future research should be conducted to develop more objective procedures so that individual candidates can be scored on their own merits rather than in comparison with what other methodologies possess or lack. Accordingly, with these comments in mind, the weighting for each assessment characteristic is as follows:

- **WHY (25 units)** in which: (a) an underlying philosophy is scored out of 15 units; and (b) a raison d'etre is scored out of 10 units.
- **WHAT (30 units)** in which: (a) producing an end result is scored out of 10 units; (b) provision of adequate tools and techniques is scored out of 15 units; and (c) provision of management guidelines and other practices is scored out of 5 units;
- **WHEN (35 units)** in which: (a) the framework's coverage is scored out of 15 units; (b) the framework's coherence is scored out of 15 units; and (c) indication of critical phases/stages is scored out of 5 units;
- **HOW (25 units)** in which: (a) documentation of the phases, activities and tasks is scored out of 10 units; (b) clear description of the application of tools and techniques is scored out of 10 units; and (c) provision of additional training and support material is scored out of 5 units.

- **IF (10 units)** in which documentation of the control mechanisms and quality assurance procedures is scored out of 10 units.

A total score of 125 would indicate that the approach is a methodology without any reservation, while a score closer to zero would indicate the complete opposite. The obvious question to ask is why not score out of 100? The answer to this question is that rather than adopt a top down approach in which 100 units are allocated across different characteristics, it was felt more important to get the correct relativity between different characteristics in the assessment framework. In fact if a score of out 100 was considered more appropriate then a simple standardisation or normalisation technique could be applied to the final figure(s) - divide the final figure by 125 and then multiply the result by 100. Finally it was considered this approach offered more flexibility if further characteristics were added to the assessment framework at a later point in time, particularly with respect to re-scaling the relativities between characteristics as a result of adding a new characteristic.

Justification for the Assessment Weights

In relative terms the most important characteristics were considered to be WHAT needs to be done and WHEN it should be done. A methodology may lack a discernible philosophical basis but if a particular candidate does not possess either of these two characteristics, then some doubt should exist on its ability to perform as a methodology. Furthermore, it is considered that WHAT is not as important as WHEN, as there may be a number of different processes that could be used, but it is more important that one of these processes is performed at a particular time rather than at another time. For example, it is important that knowledge acquisition for a KBS be performed at a particular point in the systems development process, but can be accomplished using a number of different techniques.

On the other hand, WHY is considered important in terms of setting the methodology's "tone", and providing a basic foundation for its structure. However, methodologies do not completely rely on an underlying philosophy to establish a structure which assists the system development process. For instance, SSADM has the technique of entity life histories (which are discussed in the section below) that do not have any direct relation to its underlying philosophy of decomposition (which can be seen to underpin the analysis phase of SSADM). Similar comments apply to the raison d'etre of the methodology. That is, while it is important for a methodology to provide a better development environment than nothing at all, some of the reasons for a methodology to exist will emerge from the WHAT, WHEN and HOW. For instance, ISAC has no overt justification of its existence, other than it has been successfully applied to a number of system developments.

Equivalent weightings for HOW and WHY were made on the basis that both were important but not fundamentally important. In some cases the only HOW information required is a mere description of the technique to be applied rather than detailed documentation. The equivalence for these characteristics was based on the third reason, that is there was little relative difference and no requirement for have an absolutely accurate weighting for each characteristic.

With respect to the last characteristic, this was considered least important as to some extent it was contingent upon the design of the methodology. For example, ISAC has very little overt discussion of IF within its methodology, as this characteristic is "absorbed" into the basic structure of the methodology. However, the SSADM methodology with its dual focus of system development - SSADM possesses both a process and data orientation - means that this methodology requires a lot of cross checking and control. It was interesting that in a number of the other methodologies discussed below, this characteristic raised little or no mention. However, concern about IF raises an interesting point; that is, the assessment framework should be not considered fixed across time but should develop as issues emerge within the IS field. At the moment there appears to be a great interest on quality assurance within IS development. For example, both the International Standards Association (with its ISO 9000, ISO 9001, and ISO 9000-3 standards) and the Australian Standards Association (with its AS3563 standard) have published guidelines on quality assurance for software development. Accordingly, if in the future there is greater emphasis of this characteristic within systems development process, then there should also be a concomitant adjustment of the weight for the "IF" characteristic in the assessment framework.

Another advantage of the above method that comparisons of different approaches can be made without the need to compare actual contents between the different development approaches. However, these scores should not be viewed as indicating some quantitative value of the approach, as the individual scores are determined on a judgemental basis rather than following a purely objective process. Again, with the application of the framework to more candidates and further research to refine and enhance this framework, the weighting scheme and the scoring process, it may be possible to move from a qualitative to a more quantitative approach.

Comparing System Development Methodologies (SDM's)

The last section of the literature review tries to validate the above approach by applying it to three conventional IS development methodologies, namely ISAC (Lundeberg, 1981) and SSADM (Downs et al., 1988; and Cutts, 1988) - both of which would be considered fairly mature development methodologies, as well as Boehm's (1988) spiral development methodology - which is a more recent and therefore less mature methodology. In addition, two KBS development methodologies will be assessed using this framework, namely KADS (Hickman et al., 1988; Weilinga et al., 1992; and Linster and Musen, 1992), and ES/SDEM (Matsumoto, 1989; Hayes, 1990) - both of which are fairly recent products and therefore considered to be more immature than either ISAC or SSADM. Although, this analysis may not provide the definitive answer on whether or not KADS and ES/SDEM are true KBS development methodologies, the analysis should at least determine where KADS, and ES/SDEM lie on the system development continuum in relation to these other products - commonly regarded as system development methodologies.

WHY: An Underlying Philosophy

ISAC is rather sparse in this area, being more a practical, work oriented methodology. However, the following comments indicate that its system development philosophy is that system development should be decomposed into smaller tasks, and that development should proceed sequentially: change analysis should be undertaken and completed before activity studies is commenced; and information analysis provides "an exact basis on which data system design" can be performed. The underlying approach of ISAC is that solutions to current problems need to be derived in order to advance the development process. Iteration is not addressed, and accordingly using the Davis and Olson (1985) framework, ISAC would be best suited for developments where uncertainty in the development environment is low, which would accordingly limit its applicability to KBS developments where uncertainty is often found to be quite high. Lundeberg (1981) provides some support for these contentions about ISAC's philosophical basis, namely [pp. 15-16]:

> *"If you want to analyse complex problems, the complex problems must be subdivided into sub-problems. ... A problem has to be formulated before you start to analyse and solve it. ... Change analysis has to be performed before information systems development is performed. ... Problem-oriented work must be performed before data-oriented work."*

Downs et al. (1988) describes SSADM as prescriptive, that is it dictates to system developers how the systems development effort is to be conducted, and reductionist in that SSADM breaks a development project down into phases, stages, steps and tasks, inputs and outputs - which are nested within each other. This philosophy reveals itself in what SSADM perceives to be a methodology, that is it is something that "describes how a project is organised, the order in which the many jobs are to be done, and the interfaces between them". Furthermore, the structure should be comprehensive and logically coherent, and methodologies should also describe how some jobs are to be performed, providing concrete and intellectual tools. Another underlying structure is SSADM's three different views of data. The first view is a model (an entity relationship data model) of the organisation's information base using the logical data structuring technique (described below) which describes the stable information on which the organisation and its information system are based. The second view is represented by data flow diagrams (DFD's) which show data flows into, out of, and around the information system, as well as processes which transform it, entities external to the system which communicate with it, and the storage of data within a system. The third view of the data marries the LDST data models with the processes defined in the DFD's, using entity life histories (ELH's).

Boehm's (1988) spiral model is built on the assumption that systems development (particularly within commercial organisations) is a risk driven process, and that system development should not be seen as a sequential process but rather an iterative one in which similar tasks and activities are conducted at different points within the system development. Even SSADM implicitly agrees with this assumption as the tasks that are incorporated within the feasibility phase of systems development are repeated again during the analysis phase - such as the drawing of DFD's, the conversion of physical DFD's into logical ones and so on.

Within the knowledge base field, KADS (Knowledge Acquisition and Design Structuring) would be seen as a fairly mature product (having its origins in Esprit Project 12 which started in 1983, and has been claimed as a potential standard for KBS development (Neale, 1989). (As a side issue, an interesting question therefore to resolve is the level of recognition and influence of KADS on KBS developments within Australian commercial organisations.) The fundamental principle behind KADS is that KBS development involves modelling knowledge at a series of different levels of abstraction. At the highest level, the epistemological level, four different "layers" of knowledge are distinguished in the KADS model of expertise. The domain layer includes concepts relations, and structures. The inference layer describes the domain layer in terms of the inferences that can be made on the basis of the knowledge it contains. The task layer applies the knowledge from the two lower layers in an ordered way to achieve a goal or complete a task. Finally, the strategic layer selects appropriate tasks for problem-solving and plans their execution.

This approach means that the ultimate product produced by KADS is not a strict emulation of the expert's thought processes, but rather the development of a KBS which possesses the relevant expertise to satisfy the organisation's problem, addressing other issues such as the user interface of the KBS, links to other conventional systems, or DBMS's. Part of the underlying structure for

KADS has drawn from the work of Brachman (1979). Neale (1989) also states that strong links exist between KADS and more conventional development methodologies such as the waterfall model (Barthelemy et al., 1987) with emphasis placed on a rigorous, extensively documented analysis phase, to be completed before any design and implementation - which is in stark contrast the "rapid prototyping" approach. Work has been done by Taylor et al. (1989) to adapt KADS to Boehm's (1988) spiral model.

Part of the impetus for the development of ES/SDEM (Software Development Engineering Methodology for Expert Systems) which was first published by Fujitsu in 1987, was the release in 1985 of a general purpose expert system shell by Fujitsu, namely ESHELL (Expert SHELL - based on AGE developed at Stanford University), which provides blackboard, rule and frame representations. ES/SDEM's underlying philosophy is fairly sparse, as it was predominantly a synthesis from practical experience (in ESHELL developments), recognising similarities in the problem solving processes, data structures, and the development process employed, which were systemised into "a structured expert systems development methodology". As a result of practical experience it was found that incremental development based on rapid prototyping was more effective in building expert systems than a waterfall approach used in conventional information systems development, because the problem solving process was undefined in the early stages of ES development - an interesting contrast to the approach taken by KADS!

Accordingly, all methodologies appear to have some form of philosophy or foundation to their approach, which provides structure and a common thread. It would appear that SSADM, spiral and KADS have a richer philosophical basis than ISAC or ES/SDEM, and so the following scores (maximum of 15 units) for this feature are appropriate: ISAC (8 units), SSADM (10 units), spiral (10 units), KADS (11 units), and ES/SDEM (7 units).

WHY: A Raison d'Etre

ISAC (Lundeberg, 1981) claims to have the following advantages:

1. Information systems are developed when there is a real need for them, via change analysis.
2. Information systems are developed only when they give positive contributions to the activities of the organisation in some way, via activity studies.
3. Information systems are developed in such a manner that the users understand what they contain and perform, via information analysis.
4. Information systems are developed in such a way that they are possible to change when needed and that they are not restricted only to certain computer equipment, via data system design.
5. Information systems are developed in such a way that they are effective and adapted to the technical equipment used, via equipment adaptation.

SSADM, similar to ISAC, indicates that its reason for existence is somewhat self-evident. It is the UK government's standard method for performing the systems analysis and design stages for an information technology (IT) development project. Furthermore SSADM, through its structural and procedural standards, is intended to enhance the abilities of system developers, but to provide assistance that allows system developers to concentrate on one task at a time in the knowledge that the task being performed is integrated with all the other tasks in the project by the methodology. Boehm (1988) claims that the spiral development methodology offers the following advantages:

- It incorporates prototyping as a risk reduction option at any stage of development, In fact, prototyping and reuse risk analyses were often used in the process of going from detailed design into code.

- It accommodates re-works or go backs to earlier stages as more attractive alternatives are identified or as new risk issues need resolution.

- It fosters development of specifications, as these documents, being less formal and exhaustive, take less resources to produce and so developers are more included to produce them. The specifications are also better because they defer detailed elaboration of low risk software elements until the high-risk elements of the design are stabilised.

This risk driven approach allows the spiral model to accommodate different system development approaches (such as specification oriented, prototype oriented, or simulation oriented). The perceived magnitude of the risk will often guide the selection in which the system developer matches the strategy with the perceived effectiveness of the strategy in coping with those risks. In a similar way, risk management considerations can determine the amount of time and effort that should be devoted to project activities as planning, configuration management, quality assurance, formal verification and testing. In particular risk driven specifications can have varying degrees of completeness, formality, and granularity, depending on the relative risks of doing too little or too much specification.

With respect to KADS, Harmon (1991) claims that KADS represents a movement from vague prescriptions of KBS development processes to a more systematic specification that is suitable to address large scale KBS development efforts. Weilinga et al. (1992) justify use of the methodology mainly because knowledge engineering is now driven through a modelling procedure. Weilinga et al. (1992) claim that this feature of KADS allows knowledge engineering to be decomposed, and that this divide and conquer strategy means system developers can concentrate on simpler tasks rather than addressing the complexity of knowledge engineering as a whole. Furthermore, knowledge acquisition can be structured around the KADS four layer model of expertise, and KADS provides a library of generic models as templates to support knowledge acquisition. Neale (1989) professes that the KADS methodology adopts a more traditional top-down approach to knowledge engineering, in which an "interpretation model" is selected as a template to guide knowledge acquisition. When this is fleshed out, via a process of incremental refinement, the resulting conceptual model is said to capture at a high level of abstraction - the epistemological level - both the underlying expertise and the co-operation between the problem solver and the user in order to complete the tasks in question. That is, the resulting system incorporates deep knowledge in addition to shallow knowledge.

However, Weilinga et al. (1992) also declare that KADS has a number of weaknesses, namely: (a) four layer model of expertise needs to have a less theoretical and a more practical bias - particularly differentiating task knowledge from domain knowledge; (b) the precise meaning and use of knowledge sources is ambiguous; (c) the library of interpretation models is incomplete and those models that do exist also need extension and or revision; (d) KADS does not provide enough support for transforming conceptual models of knowledge into operational ones; and (e) KADS provides little support for the modelling process.

With respect to ES/SDEM, Matsumoto (1989) claims that this methodology makes KBS development easier because it, like KADS, provides a generalised problem solving model (the EXPERT MODEL) which imposes structure on knowledge acquisition in particular and knowledge engineering in general. Furthermore, ES/SDEM although it uses prototyping as its main development vehicle, the methodology adopts a top down system development approach that starts with drafting a rough EXPERT MODEL by analysing the problem and progresses through a number of phases to the delivery of the final system. Similar to the spiral model, iteration is a fundamental part of the development process which cycles through conceptualisation, structuring, detailing and verification. ES/SDEM again similar to KADS offers models of several problem solving processes, and detailed work sheets for system documentation.

However, KBS developers using ES/SDEM have indicated that overall the methodology is a bit sparse and should be fleshed out more in terms of tools and techniques. Furthermore, ES/SDEM has a limited range of applicability with respect to KBS developments, being tied specifically to Fujitsu ESHELL developments. In this case, the argument must be against ES/SDEM for not have applicability to different KBS developments, which is not the same as saying that a methodology is less justified for existing merely because it does not cater for a variety of development environments, that is that the methodology is generic in terms of system development.

Boehm has made similar comments with respect to the spiral model, that is spiral is more suited to an in-house development environment, as software developers in this situation have greater flexibility and freedom to accommodate stage-by-stage commitments, to defer commitments to specific options, to establish mini-spirals to resolve critical path items, to adjust levels of effort, or to accommodate such practices as prototyping, evolutionary development, or design-to-cost. The spiral model may not have the same applicability in the world of contract software acquisition, where flexibility and freedom are traded off against accountability and control. ISAC and SSADM with an emphasis on sequential progression through the system development would appear to be more applicable to developments of conventional transaction processing systems rather than advanced technology projects. Downs et al. (1988) have also indicated that SSADM is probably more appropriate as a system development methodology for those organisations that have reached the third stage of Nolan's (1974) model (where there is a need to control the provision of IT services). At this time the need for a more professional approach to development is widely accepted and effort can be expended in achieving it. Neale (1989) also states that KADS may be more suited to large projects, say in excess of 12 months expected development time, as the methodology does incur substantial overheads.

However, lack of broad applicability is not seen as a relevant factor in assessing whether a product is a methodology or not, although lack of specific applicability would be, such as a KBS development methodology that is not applicable to different KBS developments. Given the range of information systems that can be developed, from transaction processing systems running on mainframes, to powerful executive information systems running on microcomputer, it would be rather unrealistic that a single development methodology could cater for all the different contingencies associated with these development environments. As Davis and Olson (1985) have indicated even one contingency, uncertainty, can give rise to four different system development approaches. Accordingly, for this reason, lack of generic applicability is not seen as a necessary condition for a product to be considered a system development methodology.

Therefore, all of these methodologies appear to provide justification for their existence. For instance, there is evidence (excepting the spiral model) that all have been used in successful developments. Subjectively, it would appear that ISAC and SSADM have the most justification, followed by KADS, the spiral model and ES/SDEM. Based on the above discussion it would seem the following scores (out of a maximum of 10 units) for this feature are appropriate: ISAC (7 units), SSADM (6 units), spiral (4 units), KADS (6 units), and ES/SDEM (3 units).

WHAT: An End Result

ISAC, given that all stages or major activities are completed (outlined below), will definitely deliver a completed system. However, the methodology as stated above, seems to be limited to particular system development environments - typically development of transaction processing systems. If used outside of these environments, ISAC may be inefficient development vehicle which could translate into incomplete developments.

Similarly, both SSADM and the spiral model are complete methodologies. Given that all phases, stages, and steps are completed, the methodology covers system development from the initial planning processes to the delivery of the production system, with appropriate manuals and other documentation. With respect to the spiral model, there may be a tendency to terminate system development with delivery of the final prototype rather than re-engineer the final prototype (using a formal systems development life cycle methodology - such as SSADM) into a production system. Therefore, in this respect the spiral model is less likely (although not unlikely) to complete, in the sense of producing a production system, than are ISAC or SSADM.

KADS on the other hand has a focus on knowledge engineering and accordingly does not cover the entire development process. Neale (1989) states that important activities such as verification and validation of knowledge, testing, maintenance, integration and development of a business oriented set of interpretation models are necessary. Furthermore, Neale (1989) comments that these activities are of great importance in terms of KADS being accepted by commercial organisations as an appropriate KBS development methodology. Difficulty in arriving at an end result may also occur in transforming the interpretation model into the final product. Hickman et al. (1989) acknowledge that it may not be possible to recognise an underlying generic task (such as heuristic classification) or combination of tasks in the application domain. This difficulty may lead to incomplete systems development, and is not an issue with either ISAC or SSADM. Finally, although some work has been done by Taylor et al. (1989), KADS at present does not have a complete and comprehensive set of output documents to support practical KBS project management.

In addition, there is other evidence that KADS is not a complete methodology. For example, although Weilinga et al. (1992) report that KADS has been used in 40 to 50 commercial KBS developments, they also state that these projects did not use KADS exclusively. Furthermore, KADS has been incorporated into a product called Structured Knowledge Engineering (SKE) marketed by a Dutch company, Bolesian Systems. Other companies such as Arthur Andersen Consulting have also incorporated KADS into their own methodology. Accordingly, the methodology appears to have a limited functionality in delivering a final product.

ES/SDEM, by virtue of its origins is regarded as a complete methodology. However, like KADS, it is considered that ES/SDEM has a number of drawbacks that, although it is complete, the methodology may not deliver a final system. First, ES/SDEM assumes development within the ESHELL environment. Second, there seems to be an underlying assumption that experienced personnel are used in the KBS development, as there is little in the way of documentation of development activities and other support materials for the novice KBS developer to use.

Accordingly, to a greater or lesser extent it is considered that all of the methodologies should provide an end result. Subjectively, it would appear that ISAC and SSADM provide the greatest guarantee, followed by the spiral model, ES/SDEM and KADS. Based on the above discussion, the following scores (out of a maximum of 10 units) for this feature are appropriate: ISAC (8 units), SSADM (8 units), spiral (6 units), KADS (3 units), and ES/SDEM (4 units).

WHAT: Tools and Techniques

The ISAC and SSADM methodologies are self-documenting, that is the techniques and tools described in the methodology can be used to present and explain in more detail the overall application of the methodology. This interesting feature means that to some extent these methodologies are recursive. ISAC has a large range of tools and techniques are available covering all development activities. For example, activity graphs (A-graphs) are used in change analysis and activity studies, I-graphs and C-graphs used in information analysis, D-graphs used in data system design, and E-graphs used equipment adaptation. Care has also been taken in consideration of the links between different development tools, for example I-graphs are derived from A-graphs, and where these graphs cannot display all information, these data are contained in associated text pages linked to the appropriate graph, or property tables.

In terms of techniques, ISAC's structure is geared to the Hackathorn and Karimi (1988) approach in which tools are nested within techniques which are themselves nested within major development activities or stages. For example, within activity studies ISAC outlines the technique to be employed to complete the activity of partitioning the target development area, namely [p. 127]:

> *"Classification of information sub-systems is performed into four groups:*
>
> - *Information systems that are impossible to formalise, for example including qualified decisions, informal contacts, and know-how. These information systems must necessarily be manual.*
> - *Formalisable information systems that are naturally manual ... The manual tasks can be performed after given rules, for example standardised telephone calls and mail procedures.*
> - *Automatable information systems with calculations that can be performed after given rules. How these work tasks should be performed (manually or by using computers or other technical equipment) is decided during the data system design.*
> - *Automatable information systems with only message transport in time (storing of messages) or space (switching of messages). This form of message transport is easy to formalise. Again how it should be performed is decided during data system design."*

Within SSADM tools are not differentiated from techniques, however some could be identified more as tools, for example data flow diagramming, rather than all being classified into a single category. In contrast to ISAC which takes a rigorous approach with respect to techniques outlined, SSADM adopts two different approaches to the techniques used with it's development framework. The technique is merely referred to - with the assumption that the project manager or system developer "knows" how to use the technique; or the technique is outlined in more detail. For example, within the development step "Initiate feasibility study" (contained with the problem definition stage of the feasibility phase), SSADM lists the following techniques as applicable to the tasks to be completed in this step [p. 16]:

*"**Techniques:***

- *Data flow diagramming*
- *Estimating*
- *Planning*
- *Logical data structuring technique"*

No further detail is provided with respect to "Estimating" and "Planning" while on the "Logical data structuring technique", SSADM provides the following information, presumably because this technique has greater importance to the development process, namely [p. 84]:

"Technique Description: the steps of LDST are:

1. *Identify the entities within the system.*
2. *Investigate and record entity inter-relationships to create the LDST grid chart.*
3. *Convert the LDST grid chart into a LDS.*

   Steps 2 and 3 tend to be omitted by experienced developers who generate an LDS as their first step.

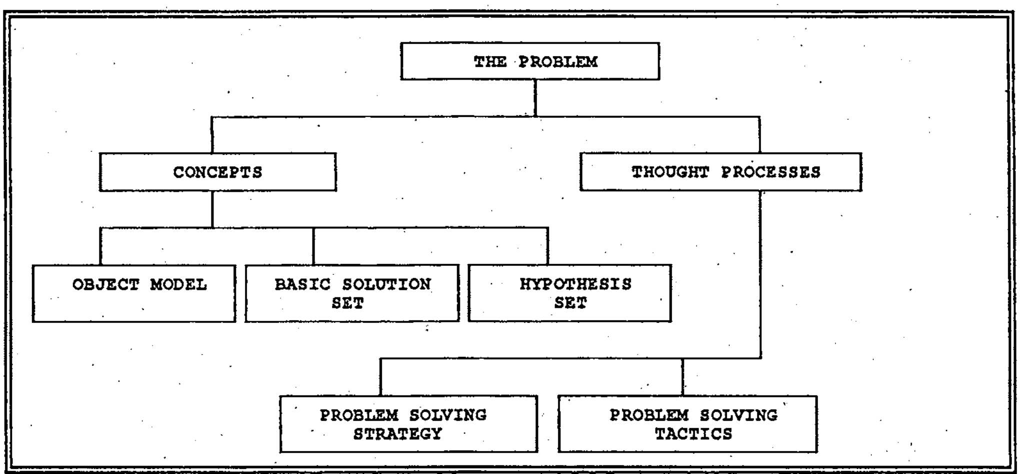
4. *Validate the LDS against the DFD's.*
5. *Rationalise the structure.*
6. *Re-validate."*

In terms of tools and techniques, KADS, ES/SDEM and the spiral model are less specific than ISAC and SSADM, and do not have as large and as varied a range in their repositories. For example, both spiral and ES/SDEM's principal tool or technique is prototyping, but little has developed from this basic technique in terms of additional tools and techniques that support or enhance prototyping overall. It appears that an underlying assumption is that experienced people will be using the methodology and so will bring to it their own tools and techniques. However, lack of specificity such as this will leave these methodologies open to interpretation and therefore impact upon the consistency of application across different development projects and across different development teams.

Some tools have been produced to assist a developer using KADS, namely Keats (Motta et al., 1989) and Shelley (Anjewierden and Toussaint, 1992). Shelley, which is also discussed by Wielinga et al. (1992), is a different type of tool to those described for ISAC and SSADM. First, this computer-based tool supports different activities across different parts of the KBS development life cycle, rather than a specific activity. In fact a more apposite description of Shelley may be a toolbox, since it contains a number of development products: (a) a domain text editor; (b) a concept editor; (c) an interpretation model library; and (d) an inference structure editor. Whether the term toolbox is more appropriate may be a moot point as the underlying emphasis is to constrain the activities of the system developer and therefore higher consistency of development effort across different projects.

Similar comments can also be made with respect to ES/SDEM - a lack of appropriate tools and techniques. ES/SDEM also has a tool similar to SHELLEY, called SAKAS (Semi-Automatic Knowledge Acquisition Support System) which automates a lot of the development activities undertaken by the knowledge engineer. For example, SAKAS contains the EXPERT MODEL and reusable program parts related to the different problem solving paradigms (discussed below). Furthermore, ES/SDEM's EXPERT MODEL can be seen as an overall structuring technique which guides knowledge acquisition. The EXPERT MODEL consists of (Hayes, 1990, p. 11):

Figure 5. ES/SDEM Expert Model



(Source: Hayes, 1990, p. 11)

This technique or model has some similarities with the KADS four layer model of expertise. Both models are based on decomposition, which allows knowledge acquisition to be a more structured task in terms of addressing each component of the knowledge individually. For example, the problem solving strategy component looks at the way an expert solves problems (similar to the strategy layer in the KADS methodology; and problem solving tactics, which are more detailed rules and procedures used by experts in problem solving has similarities to the KADS task layer.

In addition ES/SDEM also provides other guides or tools to complete particular activities within the development process. For example, in terms of determining whether or not a particular development project is suitable for KBS development, ES/SDEM offers the following "schedule", which the methodology states should be used as a guide only (Hayes, 1990, pp. 28-32):

Figure 6. ES/SDEM Problem Selection Checklist

Selecting an Application?

1. Is the problem solvable using expert systems technology?
   - (a) Does the given problem mainly require reasoning for its solution? (YES)
   - (b) Are there a few people with specialised knowledge spending time helping others to solve this problem? (YES)
   - (c) Was the competence of the recognised specialist primarily acquired through solving many problems in this area? (YES)
   - (d) Does solving this problem require a lot of commonsense? (NO)

2. Is it feasible to implement an expert system?
   - (a) Is an expert available, willing and able to assist with building an expert system? (YES)
   - (b) If multiple experts must be used to define the system, is there general agreement among them or is there one who is the obvious final authority? (YES)
   - (c) Are adequate organisational resources committed to the project? (YES)
   - (d) Are the ultimate end users of the developed expert systems willing and committed to using it? (YES)
   - (e) Is the knowledge necessary to build the system readily available? (YES)
   - (f) When solving the problem, are the facts to be dealt with uncertain or how they should be applied uncertain? (NO)
   - (g) Can the problems be broken down into smaller problems? (YES)
   - (h) Can the expert solve the problem in between 3 minutes and 3 hours? (YES)
   - (i) Is a knowledge engineer available? (YES)

3. Will the benefits exceed the costs?
   - (a) Are there tangible monetary benefits likely to accrue to the organisation through developing an expert system? (YES)
   - (b) Though the benefits may be tangible, does the development of an expert system address a significant operational or strategic objective? (YES)
   - (c) Would not developing an expert system expose the organisation to

(Source: Hayes, 1990, pp. 28-32)

The answers obtained from this schedule are then be used in the following worksheet (Hayes, 1990, p. 33):

Figure 7. ES/SDEM Problem Selection Summary Sheet

Selecting an Application: Summary

SCORES

	Sheet	Total	Weight		Score
1.			x 1	=	______
2.			x 2	=	______
				TOTAL SCORE:	______

MANDATORY REQUIREMENTS

Sheet	(Y/N)
1.	____________
2.	____________
3.	____________

Have all sheets Y/N summaries been answered "Y"? ______

(Select application with greatest score and all Y/N summaries responded "Y".

0 <= score <= 117; minimum score is 65.

(Source: Hayes, 1990, p. 33)

From the above discussion, it would appear that ISAC and SSADM offer the most in terms of tools and techniques (possibly due to the fact that these methodologies are the most mature), followed by ES/SDEM, KADS, and then spiral. Accordingly, the following scores (out of a maximum of 15 units) for this feature are considered appropriate: ISAC (13 units), SSADM (12 units), ES/SDEM (9 units), KADS (8 units) and spiral (6 units).

WHAT: Management Practices

In this area ISAC provides little guidance, other than the overall methodology itself. ISAC provides no project management facilities, and in fact states that care should be taken in selecting a project management tool - that is the tool should not perform anything other than project management. In a similar fashion, SSADM except for the detail contained within the development framework itself provides little guidance in project management issues. For example, "estimating" and "planning" are conducted in the step "Initiate feasibility study" but there is little additional detail on how these activities are to be performed. Other than this, SSADM only has general comments on project management such as the following [pp. 148-149]:

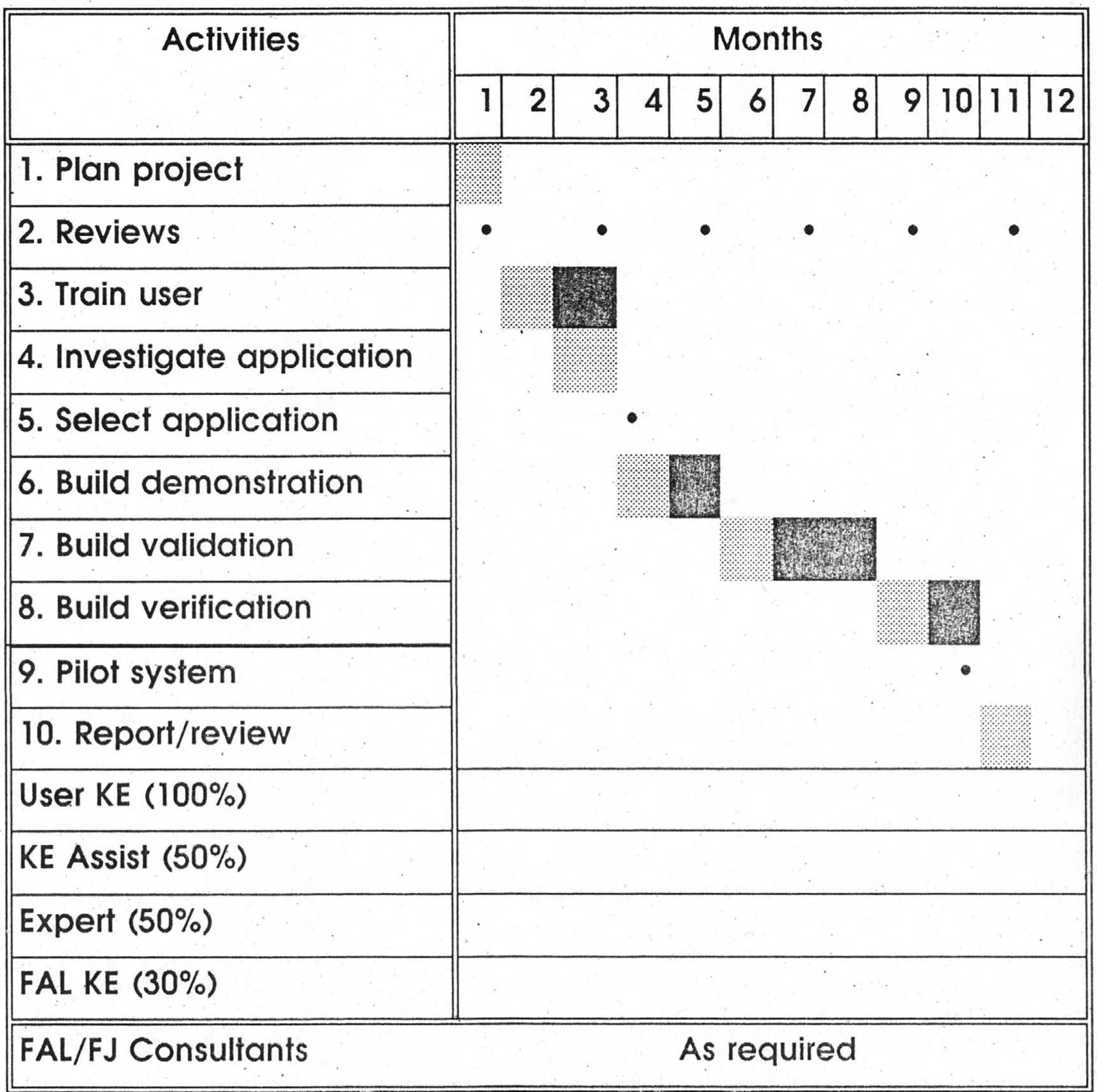
> *"Each organisation will have it own means of managing systems development. This may be based upon experience gained over many years, the political reality of that organisation, or possibly some explicit project management methodology which may be accompanied by some software tools. ... Managing project activity - may be seen as a set of jobs which need to be done. They may be listed as: (a) estimating the work load and resources available; (b) planning; (c) implementation of the plan; (d) monitoring progress; and (e) evaluating and controlling the project."*

In contrast to ISAC and SSADM, Boehm (1988) considers that the spiral model needs further elaboration with respect to this particular characteristic, in order to ensure that all participants in the system development are working consistently. For example, the spiral methodology should be "fleshed out" to include more detailed definitions of spiral model specifications and milestones, the nature and objectives of spiral model reviews, techniques for estimating and synchronising schedules, and the nature of spiral model status indicators and cost-versus-progress track procedures.

The spiral model also places a great deal of reliance on the ability of software developers to identify and manage sources of project risk. For example, a risk driven specification is very people dependent; a design produced by an expert may be implemented by non-experts. In this case, the expert, who does not need a great deal of detailed documentation, must produce enough additional documentation to keep the non-experts from going astray. With a conventional document driven approach, the requirement to carry all aspects of the specification to a uniform level of detail eliminates some potential problems and permits adequate review of some aspects by inexperienced reviewers.

For KBS developments, KADS offers very little, if anything at all, in regard to appropriate management practices that will support the methodology. On the other hand, project management is addressed in ES/SDEM, probably because it was a methodology developed from practical experience. For example, Hayes (1990) provides the following project management metrics on ES/SDEM developments [p. 5]:

Figure 8. ES/SDEM Project Management



(Source: Hayes, 1990, p. 5)

From the above discussion, it would appear that all methodologies, with the exception of ES/SDEM offer little in the way of management practices. It appears that in this area, the assumption is that these activities are external to the system development process and will be controlled using other procedures and processes that do not rely on the methodology used to development the system. Accordingly, the following scores (out of a maximum of 5 units) for this feature are considered appropriate: ES/SDEM (4 units), SSADM (2 units), ISAC (2 units), KADS (1 unit) and spiral (1 unit).

WHEN: Coverage of the Development Process

ISAC states that is it not a "phased" development methodology, but it does have structure as development is decomposed into the following activities:

1. **Change Analysis.** Change analysis precedes information systems development, as a decision not to develop an information system but merely change enhance manual processes maybe made.
2. **Activity studies.** Activity studies is concerned with delimiting future information systems by undertaking a detailed study of the activities that support the proposed information system.
3. **Information analysis.** Information analysis performs analysis and design activities to describe what the future information system will do. The description is used: (a) as a means of communication between different affected interest groups in order to facilitate the discussion of information contents; and (b) as a basis for data system design.
4. **Data system design.** Data system design creates equipment-independent data system solutions to the specified information sub-systems. The design should be made with regard to chosen processing philosophy but independent of specific physical equipment. Final decisions are made as to what parts within the systems are to be manual and what parts are to be automated.
5. **Equipment adaptation.** Equipment adaptation determines the equipment used and adapts the equipment independent data system model to this choice. This phase is concerned with the automated parts, and manual side routines that support these automated parts.
6. **Realisation.** In realisation, the information system is built according to the designed information systems models. Realisation consists of program production, file establishment, design of manual instructions, system test, and production of user and systems documentation.

As can be seen from the above, ISAC's coverage of the development process is extremely high. SSADM is similar to ISAC in that it disaggregates the development process into the following framework:

- Feasibility phase, which consists of two stages:
  - problem definition
  - project definition
- Analysis phase, which consists of three stages:
  - analysis of the current system
  - specification of the required system
  - select service level for the new system
- Design phase, which consists of three stages:
  - detailed data design
  - detailed process design
  - physical design control

Furthermore, SSADM stages can be broken down into steps, for example the problem definition stage of the feasibility phase has the following steps:

1. Initiate feasibility study
2. Create system overview
3. Create overview data structure
4. Consolidate problem/requirement list
5. Review problem definition

and within the step "Initiate feasibility study" there are the following tasks to complete (Downs, et al. 1988, p. 16):

Figure 9. SSADM - Task List for Initiation of Feasibility Study

Step 010 Initiate feasibility study

Tasks

1.	**Establish base constraints for study.**
2.	**Create initial problem/requirements list.**
3.	**Create initial Level-1 data flow diagram and overview logical data structure.**
4.	**Identify major areas for investigation.**
5.	**Estimate the amount of time required for each step of feasibility and allocate a proportion of the time available.**
6.	**Create project plan.**
7.	**Review plan with project board.**
8.	**Establish a formal quality assurance group.**
9.	**Brief project team on details of the plan.**

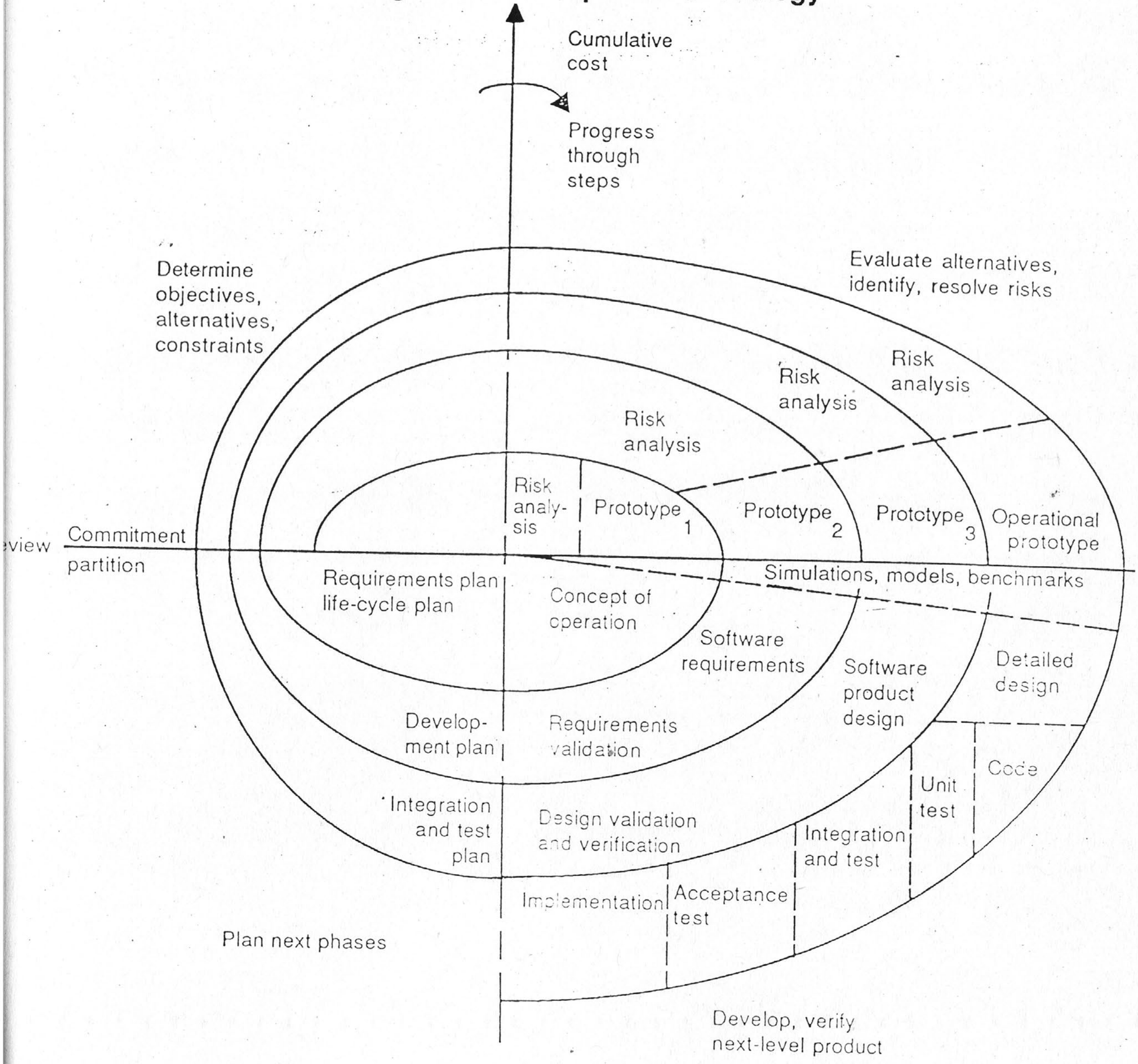
(Source: Downs, et al. 1988, p. 16)

At first glance, there seems to be nothing new, most project managers (particularly experienced ones) would perform these steps anyway, and so would not need SSADM. However while this may be true, SSADM because of the detail provided, can be used as a checklist to ensure that everything is done. Therefore SSADM provides a mechanism for greater consistency across different development projects, as well as minimising the potential for human error, that is in this particular development the project manager just forgot to do something. SSADM has also thought about interdependencies between tasks, and so relieves project managers from having to worry about these. For example, within the problem definition stage of the feasibility phase is the step "Consolidate Problem/Requirement List" which is there to [p. 20]:

> *"The problem/requirement list created in step 010 will be this stage need to have the new problems and requirements identified in subsequent steps added to it. Often this will lead to duplication and inconsistency of entries. In this step every entry on the list must be reviewed and if it is to remain, allocated a priority status (e.g. mandatory or desirable)."*

The spiral methodology is different to ISAC and SSADM in that there is not an underlying assumption of sequential progress (although SSADM has a limited "lateral iteration" capability) through the system development, but rather has an iterative progression in which similar tasks are performed many times. The framework of this methodology is best explained by a diagram, namely [p. 64]:

Figure 10. The Spiral Methodology



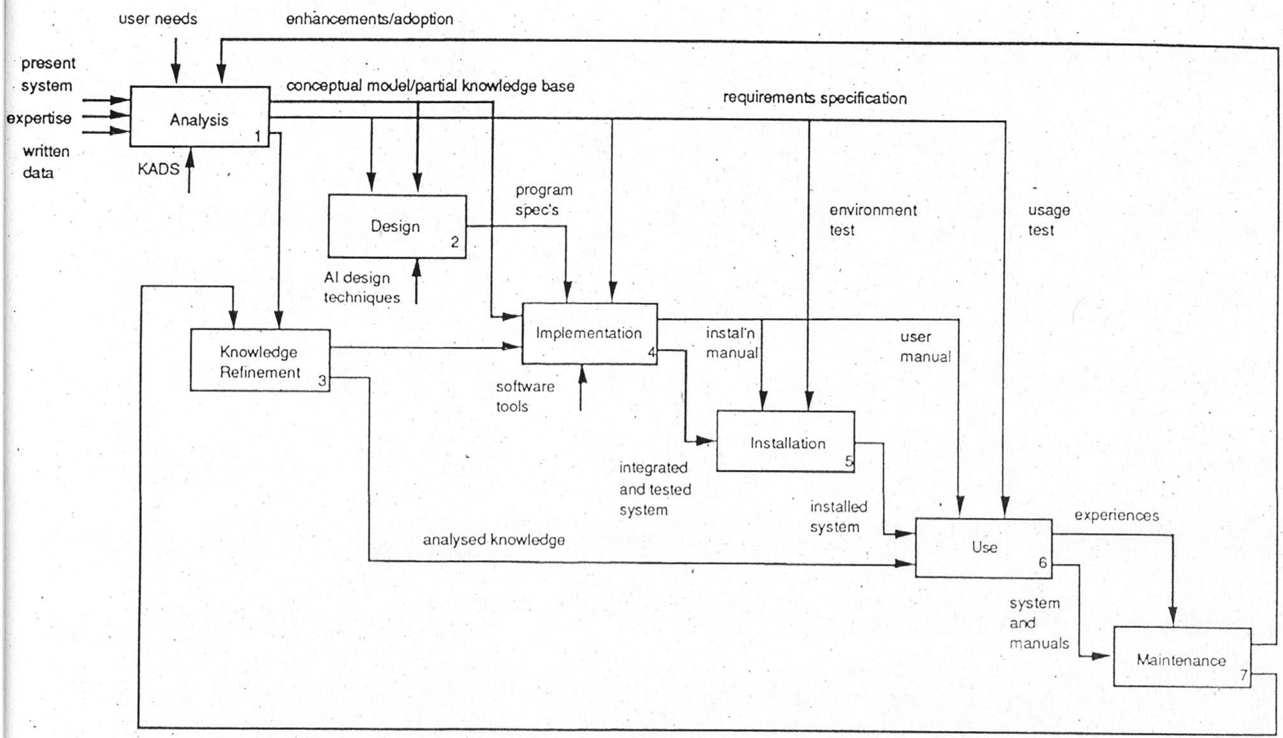
(Source: Boehm, 1988, p.64)

This model of system development has three components, namely a radial dimension, an angular dimension, and a complete cycle. The radial dimension (no matter what direction, left, right, up or down) indicates increments of cost in arriving at the current point in the system development. The angular dimension is indicative of the progress through a particular cycle in the system development. Each cycle of the spiral represents a progression of the system development that addresses the same sequence of steps, for each portion of the product and for each of its levels of elaboration, from an overall concept of operation document down to the coding of each individual program.

Each cycle of the spiral begins with an analysis of the objectives for the current cycle, the alternatives that will complete this part of the system development (say develop using consultants or develop in-house, use 3GL or 4GL), and the constraints under which this part of the system development will operate (such as costs or time constraints). Evaluation of objectives against constraints should indicate project development risks and a strategy should be formulated to resolve these risks, until at some point all outstanding risks are resolved and development can then proceed from that point on using a basic waterfall approach. Accordingly, unlike Davis and Olson (1985), Boehm is saying that the level of uncertainty can change during the development process, and accordingly the methodology should adjust for this change in uncertainty, rather than selecting one approach at the beginning of the development and sticking with it.

Hickman et al. (1989) claim that KADS is based on a modified form of the waterfall life-cycle model, in which the system development framework consists of a number of phases (with at least analysis, design, and implementation). Within each phase is a set of activities to complete which will also generate a number of products such as design documents, reports, and so on which can serve as inputs into other phases. Hickman et al. also claim that KADS identifies key review and decision points. The figure shown below presents the KADS framework (Hickman et al., 1989) [p. 26]:

Figure 11. The KADS Top-Level Life-cycle Model



(Source: Hickman et al., 1989, p. 26)

The framework for KADS is quite detailed in terms of specification. For example, during the analysis phase the primary objective is to develop or select a suitable interpretation model. To complete this task KBS developers perform external analysis (which is similar to requirements analysis for a conventional systems development) as well as internal analysis (which covers knowledge acquisition). Finally, the following documents should be created by completing the analysis phase, namely (Hickman et. al, 1989, p. 30):

- **Project Document** - project management information;
- **Requirements Document** - qualitative and quantitative expectation of the system and environmental constraints;
- **Model Document** - conceptual model of expertise and model of co-operation;
- **Feasibility Document** - estimation of feasibility from the external (business case) and internal (conceptual modelling) viewpoints;
- **Support Document** - miscellaneous project information.

In ES/DEM's methodology, system development is phased as well as based on a prototyping approach, as shown below (Hayes, 1990, p. 93):

Figure 12. ES/SDEM Development Phases

SURVEY	PLANNING	DEVELOPMENT			COMMISSIONING
		STAGE 0	STAGE 1	STAGE 2	
		PROOF OF CONCEPT	PROOF OF SOLUTION	PROOF OF USE	
				DETAILING	VALIDATION
			STRUCTURING	DETAILING	VALIDATION
		CONCEPTUAL-ISATION	STRUCTURING	DETAILING	VALIDATION

(Source: Hayes, 1990, p. 93)

Furthermore, ES/SDEM also decomposes these major phases into smaller work units. For example, Matsumoto (1989) provides the following dissection of the "Survey" phase of KBS development, in addition to the above [p. 3]:

"1. *Survey (identification of potential projects)*
- *organise surveying staff*
- *select problems to be solved using the checklist discussed above.*

- *understand the feature of the problems*
- *identify whether there is a project to solve them or not*
- *estimate the benefits to the users or the company by solving them with a computer*
- *decide on a theme*
- *establish project objectives"*

A similar decomposition occurs for each of the other phases outlined in the methodology, namely planning, prototyping, evaluation, publication, delivery, and maintenance.

From the above discussion, it would appear that all of the above methodologies have devoted considerable effort to delineating suitable frameworks that provide structure to the system development process. In terms of level of detail it would appear that ISAC and SSADM have possess the greater amount (again possible due to the fact that these methodologies are quite mature), with ES/SDEM, KADS, and spiral about the same in what they have to offer. Accordingly, the following scores (out of a maximum of 15 units) for this feature are considered appropriate: SSADM (13 units), ISAC (12 units), ES/SDEM (11 units), KADS (11 units) and spiral (10 units).

WHEN: Coherence of the Development Framework

The fact that ISAC is a sequential development approach is indicative that this methodology has high coherence. Furthermore, within the methodology this sequential progression is supported by the fact that some tasks are required to be completed before system development can progress to the next stage. This is supported by the fact that the methodology has been used in a large number of system developments, particularly within Sweden. Similar comments would apply to SSADM as has been mentioned for ISAC. Again the focus on SSADM is very much sequential progression. SSADM is slightly different to ISAC in that there are a number of similar tasks performed at different stages in the development, for example data flow diagramming. SSADM, to ensure that duplication of development effort is minimised, addresses this issue by detailing different products flowing from the technique, and that the latest iteration of the same technique uses the output of the technique from the earlier stage. For example, although data flow diagramming occurs in the feasibility and analysis phases, system developers create only high level DFD's in feasibility, whereas lower level DFD's (based on these high level ones) are created during the analysis phase. Coherence within SSADM is also enhanced but the validation and checking procedures incorporated into the overall development framework. The spiral model is considered to have less coherence than ISAC or SSADM, for two main reasons, namely lack of delineation of development tasks and products, and ability of the methodology to adopt different development structures.

In regard to the former reason, the methodology does not provide enough information regarding the differences between the specific analysis and design goals for the products "Prototype 1", "Prototype 2", and "Prototype 3". In terms of tasks, there is no delineation between the activity "Risk analysis" which is performed three times prior to the development of these prototypes. If no significant difference exists between these products and processes, then the methodological framework may be questioned. Is it really a spiral or just a series of precessions performed along a linear development path?

The second reason is that the spiral model, given different contingency factors can become similar to other methodologies. Boehm (1988) indicated that the spiral model can be considered a chameleon, for under the following conditions, it becomes equivalent to other system development models, namely:

- If there is low risk in areas such as getting the user interface right or meeting performance requirements, but high risk in predicability or control of schedule and costs, then the spiral model approximates the waterfall model.
- If requirements are stable but the software must be error free, then these risks considerations drive the spiral model to resemble the two-leg model of precise specification and formal deductive program development.
- If costs and schedule are predicable and easily controlled but has high risk in areas such as user interface or decision support requirements, then these risk considerations mean the spiral model approximates an evolutionary development model.
- If automated software generation capabilities are available, then the spiral model accommodates them either as options for rapid prototyping or for application of the transform model, depending on the risk considerations involved.

At what levels of risk do system developers change the development approach? In this regard the spiral model is silent, and so in terms of coherence appears to have a major flaw.

Coherence of the KADS methodology is considered higher than for the spiral model, but less than ISAC or SSADM. As shown in the figure above, KADS is similar to ISAC, sequential progression through the system development cycle. However, KADS does not have the proper interfaces between major phases, such as: (a) when does analysis finish and design begin?; and what things does analysis hand over to design? Similar comments are applicable to ES/SDEM. Other than the functional goal of the different prototypes what interface exists to transform the "Stage 0" prototype into the "Stage 1" prototype, or the "Stage 1" prototype into the "Stage 2" prototype?

From the above discussion, it would appear that all of the above methodologies have not devoted as much effort to the coherence of their frameworks as was evident with the coverage of the framework. It would appear that ISAC and SSADM possess greater coherence (possibly due to the fact that these methodologies are quite mature), with ES/SDEM, KADS, and spiral having around the same level of coherence. Accordingly, the following scores (out of a maximum of 15 units) for this feature are considered appropriate: SSADM (12 units), ISAC (11 units), ES/SDEM (9 units), KADS (9 units) and spiral (8 units).

WHEN: Indication of Critical Processes

ISAC is very much a holistic methodology in which development could be terminated at any point. Furthermore, the high interdependence between stages within the methodology means that no process or activity stands out as being more critical than others. From another point of view, sequential progression would indicate that the initial process is the most critical, and ISAC does devote a lot of attention to "Change analysis" - not only in terms of should be done, but also other factors that may impact of this phase such as identifying favourable circumstances in which "Change analysis" will produce an optimal product or decision concerning whether further development of an information system should occur or not. Similar comments would apply to the SSADM methodology (again a methodology that relies on sequential progression) with perhaps the only difference that quality assurance and checking activities (discussed below) are given enhanced importance when compared to other development activities and tasks.

Obviously, the spiral model has indicated that risk identification, analysis and resolution - wherever they occur in the development process - are the critical processes. However, although Boehm (1988) provides some comments on how risk analysis will impact on the next iteration or traverse of the development cycle, there is little comment on how this critical activity is to be completed. For such an important process it would be expected that more detailed documentation should exist. With respect to KADS, there is some indication that analysis is a critical phase (for similar reasons that "Change analysis" is important or critical within ISAC) given the level of attention it receives in the outline of the methodology. Similar comments would also apply to ES/SDEM and the prototyping activities used in the "Development" phase of the methodology. However, with respect to the last two methodologies, these indications are more implicit rather than explicit.

From the above discussion, it would appear that all of the methodologies either explicitly or implicitly have addressed this issue, with some more than others. Accordingly, the following scores (out of a maximum of 5 units) for this feature are considered appropriate: SSADM (3 units), ISAC (3 units), ES/SDEM (2 units), KADS (2 units) and spiral (1 unit).

HOW: Phases, Activities and Tasks

ISAC provides a lot of guidance in terms of how major stages are to be performed, as well as the tasks and activities found within that stage. For example, in change analysis ISAC specifies that for this stage to be completed effectively, people from different interest groups must participate in change analysis [p. 64]. ISAC also describes the steps required to move from one stage to another, namely to move from activity studies to information analysis, a system developer would perform the following steps [p. 198]:

"1. *Extract input and output information sets from the A-graph of the sub-system and put these in a first outline of an overview I-graph (information precedence graph).*

2. *Study the text pages of the A-graphs in order to start making a more precise description of the input and output information sets.*
3. *Refine the outline of the overview I-graph by means of precedence analysis. There will be one overview I-graph for each sub-system. These are distinguished by different prefixes (one or two letters) before the reference codes."*

Finally, ISAC provides detailed guidelines on how particular processes are to be completed both within a particular stage of development and also to ensure a smooth progression of the system development between major phases or stages. ISAC also specifies what deliverables should be produced for each stage, for example at the end of change analysis the following documents should have been created [p. 120]: (a) A-graphs; (b) text pages to A-graphs; (c) property tables; (d) tables of needs for changes; (e) list of interest groups; and (f) list of "unsolved" problems.

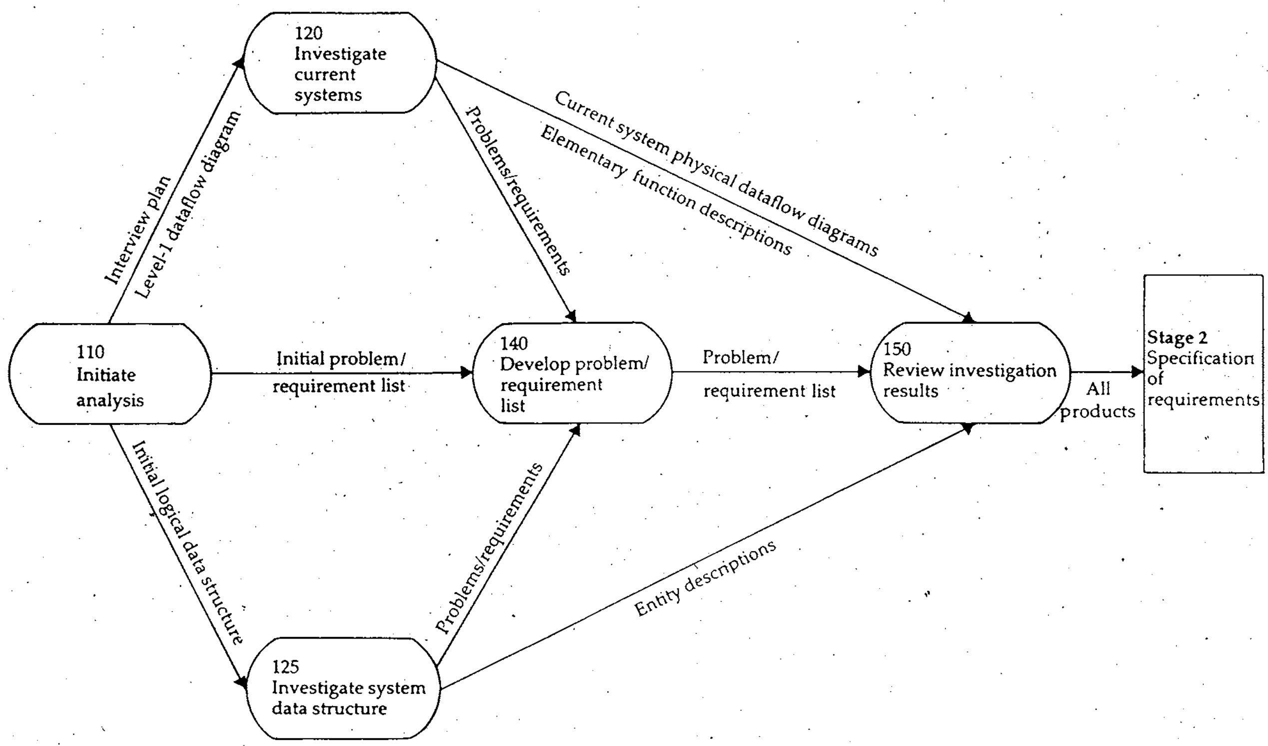
SSADM in terms of "HOW" at this level provides general information regarding the objectives of each phase. For example, SSADM states that the objective of the feasibility stage is to examine the business case for, and technical feasibility of, the project. With respect to the feasibility phase, SSADM also provides guidelines on where a feasibility study is inappropriate, namely [p. 13]:

"• *Not desirable:*
  - *small projects with limited budgets and time scales*
  - *where the project is being developed as part of a larger strategic plan, provided the strategic planning exercise has covered many of the aspects, and taken the decisions, required of a feasibility study*

- *where development work has already been done without using SSADM*
- *Desirable:*
  - *for projects with high costs and long time scales*
  - *for sensitive projects where justification is in doubt."*

Furthermore, the sort of information is available at more specific layers within the SSADM framework such as at stages, steps and even tasks. For example, SSADM provides the following information concerning the first stage of the "Analysis phase" of system development, namely "Analysis of Systems Operations and Current Problems", "this stage is an analysis of the current system and involves ensuring that the system is correctly documented in DFD and LDS format." [p. 28]

Figure 13. Stage 1 - Analysis of Systems Operations & Current Problems



(Source: Downs, et al. 1988, p. 28)

Information is also presented at the next layer within the development framework as well. For example, the following information concerns the first step of this stage, namely "Step 110 - Initiate Analysis" [p. 29]:

> *"Some or all, of the activities may have been done as part of the feasibility phase. If this is the case the effort required for this step will be small. Additionally, if a project management system is used in conjunction with SSADM, some of these activities will fall within its scope. The techniques of DFD and LDST are used here to produce a "Level-1" and an "Overview" respectively. Users must fully understand what is required of them: this may require provision of SSADM overview training. A problem/requirement list must be created or revised from information provided in the analysis requirement document and from initial discussions. Importantly, QA procedures must be set in motion."*

For the spiral model, Boehm (1988) states that four fundamental questions arise in terms of progression through the development process, namely:

- How does the spiral ever get started?
- How do you get off the spiral when it is appropriate to terminate a project early?
- Why does the spiral end so abruptly?
- What happens to software enhancement (or maintenance)?

However, in comparison with ISAC and SSADM, this feature of the methodology only receives somewhat superficial treatment. For example, the answer to the fourth question is a bald assertion that the spiral model applies equally well to the development of enhancement efforts, without any detailed prescription of how it would be applied within a maintenance project. Furthermore, the same approach is taken with the first question, where it is asserted that the spiral gets started "by a hypothesis that a particular operational mission (or set of missions) could be improved by a software effort" [p. 66]. Boehm (1988) claims that development can cease at any time, when the above hypothesis is proved invalid. However, the spiral model is silent in terms of what makes up a particular operational mission and how much or how little improvement triggers continued development or termination. Probably a good logical approach but in reality in order for this methodology to be used effectively the procedure would require more detail. The spiral obviously ends with the installation of new software, and success will be analysed by testing the development hypothesis against the real the effect the installed software has on the operational mission. Again the procedure is superficial in the sense that no factors to be used in the assessment are defined and no performance metrics for these measures outlined.

The KADS methodology is somewhat similar to the spiral model, both of which have a more of an "academic" flavour. There is a lot of information on the theoretical background of the methodology, but little "HOW" information to assist in the practical application of the methodology. For example, Linster and Musen (1992) claim that KADS "is not a cookbook for building expert systems" [p. 82], and that the methodology offers little guidance at a practical level, for example there is little to tells a system developer how to construct a conceptual model or how to transform a conceptual model into a design model. Accordingly, these authors claim that the heavy reliance on the knowledge engineer's skills could mean inadvertent elimination of relevant knowledge, or the incorporation of irrelevant knowledge! That is, there is a lack of rigour and discipline associated with this very important process within the KBS development.

Hickman et al. (1989) state that KBS design in KADS follows a top-down approach which is broken down into the following major stages: (a) functional description, a decomposition of the system into a hierarchical structure) which is created from the requirements specification and conceptual model; (b) selection of methods such as inferencing strategies, search algorithms and other more conventional methods that interact with the functional blocks of the functional description; and (c) physical description which is the set of physical modules specifying how to achieve the requirements of the functional layer. As can be seen from these comments, although on the surface system developers have an idea of what to do, more detail is required for KADS to be effective in a practical situation.

With respect to other system development phases, such as installation and maintenance (see the above figure), KADS states that procedures are similar to those of conventional IS developments, for example decisions on production hardware and software should not be made until the design phase is completed. Whether or not most commercial organisations have this luxury is probably a moot point. KADS does consider maintenance of specific KBS components to be slightly different to normal systems maintenance - this activity is known as knowledge refinement and is completed using a rapid prototyping approach (since the system now exists). Again the sort of information provided at this phase is that system developers should determine the scope of the phase, elicit and formalise the knowledge - too superficial to be considered effective "HOW" information. Finally, the central feature of the KADS methodology is the four layer model of expertise. Again KADS is more concerned with "WHAT" the model is rather than prescribing "HOW" it is to be applied within the KBS development. For example, Hickman et al. (1989) provide the following comments on the model of expertise's strategy layer [p. 51]:

"Strategy Layer. *At the strategy layer, the strategic reasoning about the problem solving process is represented. The knowledge that resides here allows the system to make plans. It enables the execution of the problem solving to be monitored and if a deadlock occurs a replanning of the problem solving procedure may occur, that is a repair mechanism is available. This is equivalent to choosing an alternative task structure. Thus, it captures the circumstances under which task structures may be applied during problem solving."*

ES/SDEM, primarily due to the circumstances of its creation has a more practical emphasis than KADS. However, it also suffers from a lack of detail in the "HOW" information provided. For example, the most important of phase of the methodology is the "Development phase", which follows a prototyping approach of the "keep and enhance" category rather than the "throw-away" type of approach. Although prototyping is essentially a technique, within ES/SDEM it also forms actual development phases, and so is open to the criticism of using a pure prototyping approach that was discussed earlier in this chapter. Notwithstanding these comments, "HOW" information is provided as indicated by the following information on evolving the prototype (ES/SDEM also outlines prototyping as a technique, which is discussed below):

- **Stage 0: Demonstration prototype system.** KA done by interviews and case studies, to confirm that the problem has been recognised correctly. The prototype developed is an executable systems which can only solve limited problems. This prototype is reviewed by the expert. Time taken in this stage is from 2 weeks to 1 month on average.
- **Stage 1: Validation prototype system.** The previous prototype is used to drive the knowledge acquisition in this stage. The

prototype produced at the end of this stage is reviewed by the manager. If it is not producing "good" results then the project should be cancelled. This stage should not take longer than 3 months (or what was defined during the ES/SDEM planning process).

- **Stage 2: Certification prototype system.** The previous prototype is used to drive the knowledge acquisition in this stage. The prototype produced at the end of this stage is reviewed by the expert again to determine its problem solving capability. With the agreement of the expert, the kernel part of the knowledge based is certified and fixed.

From the above discussion, it would appear that all of the methodologies have to varying degrees provided "HOW" information regarding the phases, activities and tasks of the methodology. It would appear from the above discussion that ISAC and SSADM have a greater amount of this feature than either ES/SDEM, KADS or the spiral model - again possibly due to exposure in practical development situations, and the maturity of the product. Accordingly, the following scores (out of a maximum of 10 units) are considered appropriate with respect to this feature: SSADM (8 units), ISAC (8 units), ES/SDEM (6 units), KADS (3 units) and spiral (2 units).

HOW: Tools and Techniques

In terms of tools and techniques, ISAC provides detailed "HOW" information on the tools and techniques used within the methodology. For example, A-graphs (activity graphs) which are used in change analysis and information studies, have the following information on usage of this tool [pp. 94-103]:

"An A-graph or activity graph is a picture of some activity, which consists of:

- *Sets. Sets of different kinds, for example, persons, material objects, and messages.*
- *Activities. Activities involving persons and other resources with the purpose of transforming input sets into desired results (output sets)*
- *Flows. Flows of real sets (persons/objects) and messages that connect different activities. [p. 94] ...*

Text pages - there is not much space for much in the symbols of A-graphs. Therefore the A-graphs have to be supplemented by text pages, which describe the input sets to the A-graph, the sets and activities inside the frame of the graph, and the output sets from the A-graph. [p. 103] ...

A-graphs can be supplemented with qualitative and quantitative information in the form of property tables in order to obtain a description of specific activities. [p.105]"

SSADM, also provides detailed guidelines on how particular techniques are to be completed, namely [p. 78]:

"Current Logical DFD's.

Converting current physical DFD's to logical DFD's in termed logicalisation, by performing the following steps"

1. *Rationalise data stores: remove duplication and redundancy using the LDS as a guide.*
2. *Combine duplicate processes.*
3. *Remove processes and data stores concerned only with the scheduling of the physical system.*
4. *If the current systems is a computer system then removal*

and review of the human computer interface is required to remove physical considerations.

5. *Replace form names with logical data descriptions. These are taken from the data dictionary.*
6. *Ensure that only required data are passed to a process. This may involve creating additional by pass data flows.*
7. *Break down large physical data stores that correspond to several entities in the data mode, creating additional data stores if necessary.*
8. *As a result of the above tasks, the DFD set may require levelling to produce a functional rather than departmental hierarchy."*

In addition, over and above explanation of tools and techniques, SSADM also provides, mainly as side comments, help with particular aspects of applying the methodology, most of which appear to be gained through experience, for example[p. 79]:

> *"Elementary Function Descriptions.*
> *Elementary function descriptions are almost always documented using narrative. However, decision tables or decision trees are sometimes required if a particular function contains complex logic. These descriptions should never be more than half an A4 page in length. If they are, the processes need further sub-division. Decision tables and decision trees are more widely used later in the development cycle during the production of process outlines."*

"HOW" information on tools and techniques for the spiral model is almost entirely non-existent. For example, the major technique associated with the spiral model is prototyping, and even at a very high level, spiral does not provide any indication whether this technique uses a "keep-and-enhance" or "throw-away" approach. In addition, although prototyping is a dominant technique within the early analysis and design phases, the spiral model does not present additional information on how this technique is to be applied. In fact it appears that spiral assumes anybody using the methodology will know what to do. However, as Hickman et al. (1989) have stated (see above), this technique has a number of variations and therefore people's knowledge and application of the technique may vary considerably across different developments with different results. Similarly another important activity to the spiral model is risk analysis/assessment. Again with respect to this important technique, the spiral methodology offers little in terms of practical advice and instruction.

Boehm (1988) also indicates that the methodology is deficient in this area by stating the current version of the spiral models needs additional guidelines and checklists to identify the most likely sources of project risk and procedures which help to identify the most effective risk resolution techniques for each source of risk. These additions would help novice project managers rather than experienced ones, but even for the latter category would ensure greater consistency in the application of this development methodology.

In terms of "HOW" with respect to tools and techniques, the KADS methodology is quite sparse; there are very few tools and techniques within the methodology and therefore a limited amount of explanation. As has already been stated, Linster and Musen (1992) say that KADS is no "cookbook for building expert systems", offering little guidance concerning how to construct a conceptual model and how to transform that conceptual model into a design model. In agreement with Neale (1989) these authors state that although KADS offers interpretation models (which contain concepts, structures and relations at the domain layer, inference structures or knowledge sources at the inference layer, and goals and tasks for the task layer), KADS does not give direction on how to use these models with a given problem domain. Furthermore, Linster and Musen (1992) state that KADS is not prescriptive enough, so that it is often difficult to transform conceptual models into design models. Although KADS offers considerable freedom in helping the developer build a conceptual model, the technique does not assist the developer in overcoming the bias and information loss that can occur during the model transformation phases. Model transformation is still a poorly understood process that relies heavily on the knowledge engineer's intuition and programming skills. Nevertheless, these authors claim the potential is there by making explicit the intermediate stages between the initial conceptual model and the ultimate program code which they consider to be a valuable step toward more principled approach to knowledge engineering. Therefore, in terms of tools and techniques, KADS is less specific than ISAC and SSADM but probably on a par with Boehm's (1988) spiral.

There are indications that "HOW" information for ES/SDEM, is also sparse. For instance users of ES/SDEM have claimed that it is a good guide to building expert systems and has produced good results with practical applications. However, users of ES/SDEM have also indicated they need more than just a manual methodology. For ES/SDEM as with the spiral model, the assumption appears to be that experienced Fujitsu development staff will be used in KBS developments and so little explanation is provided within the methodology. For example, Matsumoto (1989) in outlining the methodology offers the following [p. 99]:

> *"When undertaking KA activities, it is important to acquire only the necessary, but sufficiently enough, knowledge to solve a given problem. Though sufficiency is not easily proved, it is*

possible to avoid unnecessary KA by obtaining only that knowledge needed to produce the required solution. Therefore ES/SDEM proposes, as a first step, the goal of solution definition, which encompasses the external forms of the solution, as well as the design of the internal data structures that hold sufficient information to express those external forms."

As far as knowledge acquisition is concerned the above may provide a good guideline, but it could hardly be called a comprehensive road map for this activity. However, a more comprehensive treatment from a practical rather than a theoretical point of view is given to prototyping. For instance ES/SDEM offers the following prescriptive guidelines on using prototyping as a technique within the "Development phase" of the methodology (Matsumoto, 1989) [p. 100]:

" ... ES/SDEM employs a four phase prototyping approach, which can be described as original form prototyping:

i) CONCEPTUALISATION - results in an initial skeleton design of the application based on the Expert Model.

ii) STRUCTURING defines the structure of the problem solving process, as well as of the domain specific data

iii) DETAILING adds knowledge to the structured design in the form of individual rules and frames as appropriate to the development environment

iv) VALIDATION establishes the correctness of the solutions derived from the expert system through, for example, the execution of tests cases

The type of prototyping is a "keep-and-enhance" approach as ES/SDEM indicates that the prototype evolves as it progresses through the prototyping phases outlined above. Furthermore, again due to the practical emphasis of this KBS methodology, ES/SDEM also provides information on a number of development tools associated with both the Expert Model and the prototyping approach used in the "Development phase" of the methodology. First, it provides within the methodology four paradigms of problem solving processes, or inferencing strategies, namely (Matsumoto, 1989) [pp. 6-7]:

> *"There are some typical mechanisms for the PSP and these have been incorporated into ES/SDEM, namely:*
>
> - ***Branching** - each node of the search tree is represented by a decision table. Branching between these tables produces a decision. this method is applicable for classification type problems, like diagnosis.*
> - ***Generate and test** - reach a solution by repeatedly generating and test interim solutions, which is applicable to time series type problems like scheduling.*
> - ***Focus and adjust** - focus on sub-optional parts of an object and adjust it according to some preferred criteria to optimise that object, this method is applicable when tackling design type problems.*
> - ***Scoring** - use some standards or points when making a selection; by scoring objects based on individual standards, and weighting them by some factor, we can select the most preferable case. . This method is applicable for selection type problems."*

With respect to the knowledge base, ES/SDEM provides an "Object Model" tool, namely (Matsumoto, 1989) [p. 9]:

"OM is often represented by frames or objects in object oriented programming.

OM - extract nouns related to the problem, and structure the nouns focusing on entities in the real world; generalise the nouns to ideal concepts

- *define attributes to each object, that is attributes should be defined so that they can be read as:*

  [<ATTRIBUTE> of <OBJECT> is <VALUE>]

- *structure relationship between objects by class and sub-class - the implying relation is represented by a hierarchical structure, and sub-classes can inherit information from their super classes*

  is-a: different abstraction level with the same data structure that has an implied relation

  part-of: the same abstraction level with a different data structure that has a part-of relationship

- *add functions or rule sets - adding functions or rule sets to the objects, we can give the OM intelligence. Here intelligence means the facility to return an appropriate answer when referenced form outside the OM. This process is repeated during prototyping."*

These techniques and tools have been developed as a result of development and empirical studies, which Matsumoto (1989) claims reduces the effort involved in knowledge acquisition, particularly with regard to the discovery and design of control structures within the KBS.

From the above discussion, it would appear that all of the methodologies have to varying degrees provided "HOW" information regarding the techniques and tools used in system development. It would appear from the above discussion that ISAC and SSADM have a greater amount of this feature than ES/SDEM, which has more than KADS or the spiral model, given the greater practical experience and maturity of the first two and the practical origins of the third. Accordingly, the following scores (out of a maximum of 10 units) are considered appropriate with respect to this feature: ISAC (8 units), SSADM (7 units), ES/SDEM (7 units), KADS (4 units) and spiral (1 unit).

HOW: Additional Training and Support Material

ISAC and SSADM both have substantial amounts of additional support material. For example, Lundeberg (1981) discusses a number of case studies outlining the methodology, and claims that training courses in ISAC's information analysis have been available in Sweden since 1969, with training modules being updated as the methodology evolved. In fact Lundeberg (1981) considers the "HOW" information and training support available often means people mechanically apply the methods without asking themselves what the results should be used for. People became disappointed when the methods did not automatically solve their problems! Similar comments would apply to SSADM, with Cutts (1987) offering a substantial case study in the methodology. The fundamental aims of SSADM were that it should be self-checking, used tried and tested techniques, be tailorable, and be teachable.

With respect to KADS, there is additional training and support where KADS has been subsumed into a proprietary methodology such as Structured Knowledge Engineering marketed by Bolesian a Dutch organisation. Similarly, since ES/SDEM is a proprietary product there is training material available (Hayes, 1990). However, in both cases it is considered that this material is not quite as extensive as it is for ISAC and SSADM. The spiral model seems to have nothing to offer in this area.

Accordingly, the following scores (out of a maximum of 5 units) are considered appropriate with respect to this feature: ISAC (5 units), SSADM (4 units), ES/SDEM (3 units), KADS (1 unit) and spiral (0 units).

IF: Quality Assurance and Control

The ISAC methodology, having been created in the 1960's and undergone many enhancements and changes since its creation, is considered a quite mature methodology. Furthermore, the philosophy followed by ISAC, is that where situations arise in which ISAC has been found wanting, these "problems" will be addressed and subsumed into the methodology, rather than the methodology having a separate "**IF**" section which handles exceptions and unusual situations. Accordingly, because of this and the fact that the methodology has been applied in many different systems developments, there is little to perceive in terms of this characteristic.

Notwithstanding this, ISAC does offer some guidelines or "advice". For instance with respect to Change Analysis, ISAC recommends that before change analysis begins options on the change strategy should be finalised, which has high agreement with the current needs and circumstances of the organisation, and the more process oriented the chosen change strategy is, the more likely the proposed change will be accepted. As a general warning concerning use of the methodology, ISAC claims that [p. 326]:

> *"The worst danger in using ISAC approach involves applications in which people mechanically apply the methods without asking themselves what the results should be used for. People have become disappointed when the methods did not automatically solve the problems they thought they had. People should try to form a realistic opinion of what the ISAC approach can contribute and what they themselves must contribute."*

SSADM has extensive cross checking facilities built into the methodology to enhance quality assurance in the development. For example, the methodology indicates that although decomposition of a development into smaller tasks improves development efficiency and effectiveness, it also could lead to duplication and inconsistency. Accordingly, SSADM includes a number of cross checking features (often referred to as lateral iteration). For example, the data model developed using the logical data structuring technique (LDST) - essentially a top down process - is checked against the data designs developed by relational data analysis (RDA), which is a bottom up technique. Another illustration of cross checking is in the development of the processing model. Functions are modelled with top down data flow diagramming. From DFD's and the information in the data model, the effects of system events on the data are described with entity life history (ELH) models. The view developed by this technique is used to check that the DFD's are complete, and as a basis for the development of process outlines.

With regard to Boehm's (1988) spiral model, an important feature is that a review process is conducted at the end of each cycle. Who is to conduct this review is less well defined, but in content the review covers all products developed during the previous cycle, including plans for the next cycle and the resources required to carry them out. The review's major objective is to ensure that all concerned parties are mutually committed to the approach for the next phase. The review-and-commitment step (also called risk analysis in the figure above) may range from an individual walk-through of the design of a single programmer's component to a major requirements review involving developer, customer, user, and maintenance organisation.

With respect to the KBS development methodologies it appears that this feature is not as evident as for the conventional development methodologies. For example, for KADS, Hickman et al. (1989) claim that the methodology includes review and decision points at various stages of development. However, as indicated in the above figure, there is no formal verification and validation stage, which is also commented on by Neale (1989), and review and control activities within the development phases are hard to find. ES/SDEM indicates that review milestones should occur at the end of each major development stage, namely survey, planning, development and commissioning. In addition there are validation processes invoked at the end of each prototyping cycle (Stage), Stage 1, and Stage 2).

Accordingly, the following scores (out of a maximum of 10 units) are considered appropriate with respect to this feature: spiral (9 units), SSADM (8 units), ES/SDEM (7 units), ISAC (4 units) and KADS (2 units).

The System Development Continuum

The final step in this analysis is to accumulate the scores and determine where on the system development continuum each of the above products would lie. The scores for each of the features discussed above are summarised in the following table:

Table 13. Assessment Scores: Five System Development Methodologies

Feature	ISAC	SSADM	Spiral	KADS	ES/SDEM
WHY: Underlying philosophy (15)	8	10	10	11	7
WHY: Raison d'Etre (10)	7	6	4	6	3
WHAT: An End Result (10)	8	8	6	3	4
WHAT: Tools and Techniques (15)	13	12	6	8	9
WHAT: Management Practices (5)	2	2	1	1	4
WHEN: Coverage (15)	12	13	10	11	11
WHEN: Coherence (15)	11	12	8	9	9
WHEN: Critical Processes (5)	3	3	1	2	2
HOW: Phases, etc. (10)	8	8	2	3	6
HOW: Tools and Techniques (10)	8	7	1	4	7
HOW: Additional training (5)	5	4	0	1	3
IF: Review & Quality Assurance (10)	4	8	9	2	7
TOTAL (125)	89.00	93.00	58.00	61.00	72.00

In conclusion, even though these methodologies have taken very different approaches with respect to systems development, this analysis has provided some insight into how much of a methodology each product when compared with the others. It is interesting that the analysis has not indicated that any of the above could be considered a methodology without any reservation (gaining a maximum score of 125). The validation of the assessment framework for systems development methodologies is considered to have shown it as an effective instrument in determining whether a development approach is a methodology or not. Clearly ISAC and SSADM can be considered methodologies, KADS and ES/SDEM to a lesser degree, whereas the spiral model is only marginally a real world systems development methodology.

Chapter Summary

This literature review has been divided into a three major sections. The first section was concerned with an evaluation of the research literature that focussed on the deployment of KBS's within commercial organisations. The objective of the review for this section was to resolve (within the context of the published research) two major issues concerning problems solved using KBS technology and components that comprise KBS solutions to those problems. The major questions to resolve were:

- Are the types of problems addressed by KBS technology similar across different KBS developments either within a single organisation, or spanning different KBS developments by different organisations? The research literature seems to indicate that diversity is more prevalent than commonality.
- A similar analysis was employed in connection with the second question, namely are the major components of KBS's (such as knowledge representations stored in the knowledge base, inferencing strategies that comprise the inference engine and the user interface) similar across different KBS developments either within a single organisation, or spanning different KBS developments by different organisations? Again with respect to this issue, the research literature seems to indicate that diversity is more prevalent than commonality.

Accordingly, it would appear appropriate for the case study to either confirm or deny the above suppositions within the Australian context. The next major section of this literature review looked at case study and other research concerning the development approach taken to deliver KBS technology. The analysis here was structured on the premise that either similar development approaches could occur across different problems with different KBS's developed, or there was no commonality. The research literature indicated that a number of different development approaches could be used (Lehner and Adelman, 1989). However, it was questioned whether the actual differences in a practical environment between these approaches was more superficial than fundamental. Furthermore, there was a body of research literature to indicate that one particular development approach dominated KBS development, namely prototyping. In spite of this, dominance does not indicate superiority of the technique over other development approaches. In fact a number of researchers have seriously questioned whether this development approach is an optimal one (Hickman et al. 1989, Wietzel and Kerchberg, 1989; and Slagle et al., 1990). Some analysis was also performed to determine the adequacy of using conventional IS development approaches and techniques for KBS development.

Therefore the research project should consider at least two issues emerging from this section of the literature review. First, is prototyping the predominant development approach used by commercial organisations in Australia to build KBS's? Furthermore, irrespective of the development approach, do commercial KBS developments within Australia reflect the development strategies outlined in the research? For example, with respect to prototyping, what form of prototyping is used - "throw-away" or "keep-and-enhance", or some other form entirely different to these two. Second, what similarities in terms of activities, milestones, deliverable, development tools and techniques and documentation are there (if any) between conventional IS developments and KBS developments?

The final major section of the literature review investigated the characteristics of systems development methodologies, and whether or not products exist that could be considered KBS system development methodologies. The research literature raised a number of difficulties or potential problems which the case study research would have to address if meaningful conclusions are to be reached. First, how important are these products to the success of a KBS development? Is there a need for KBS development methodologies? Obviously given the amount of research effort with respect to conventional systems development, it would appear that methodologies are important to the construction of conventional systems, in particular transaction processing systems (using "standard" development methodologies such as Information Systems Analysis and Change (ISAC) and Structured Systems Analysis and Design Method (SSADM). However, KBS's are not transaction processing oriented systems, and accordingly, it may be invalid to extend the reasoning from conventional developments to KBS developments. Furthermore, if the case study research determines that KBS developments have completed in the sense of creating viable production KBS's, then the importance of these products should be questioned. It may be that commercial organisations will not devote a lot of effort to something with doubtful overall benefit.

The second issue of concern is that for KBS development, as yet the research literature does not suggest an overriding KBS development paradigm, unlike conventional systems which have followed to a large extent the structured system development methodology (or some variant of it) as outlined by Yourdon and Constantine (1979). Whether the emerging object oriented design and development paradigm subsumes both conventional systems development and KBS development into one is an issue that is beyond the scope of this present research project.

The third issue concerns a lack of rigour in determining what is meant by a system development methodology. A variety of definitions exist (such as those offered by Avison and Fitzgerald (1989); Bubenko (1986); and Hackathorn and Karimi(1988)) which although have similarities cannot be considered equivalent. Even the Oxford English Dictionary (1989) is of little assistance. Accordingly, an assessment process used to determine what is, or what is not, a systems development methodology is confounded at the very beginning by not having a secure definitional foundation. This has given rise to the phenomenon that over the years many different sorts of things have been labelled methodologies which do not have a lot of characteristics in common.

For instance, the classic example is the prototyping approach to system development. Often this technique is incorporated as a component of a larger system development approach, while at other times systems are developed using this technique alone. The question therefore arises; is prototyping is a system development methodology when used by itself, and a development technique when incorporated into something else? Another complication is the structure of the assessment process itself. The work of Floyd (1986) and the commentary by Avison and Fitzgerald (1989) on the research performed by Maddison (1983) and Catchpole (1987) highlighted the implausibility of using a digital rather than an analogue assessment approach. That is rather than deciding what is and what is not a system development methodology (SDM) it may be more appropriate to consider these products as residing on a continuum - at one end being considered an SDM without reservation, and at the other being considered unlike an SDM without reservation. With this technique system development products such as prototyping and knowledge acquisition techniques such as the Knowledge Acquisition Activity Matrix (KAAM) and the Critical Decision Method (CDM) can have a number of similarities with systems development methodologies without actually being classified as a methodology in their own right. With respect to the system development products mentioned above the major feature which prevents them from being considered SDM's in their own right is that they offer only partial rather than a complete guide in the overall development process.

Finally, even if an analogue type assessment approach is applied, what are the items of assessment. Yourdon and Constantine (1979) state that while reductionism is an effective underlying philosophy on which to base a methodology, this is not equivalent to mere disaggregation [p. 18]:

> *"Of course many designers have made attempts to "Chop" a system into manageably small pieces; unfortunately, they have often found that implementation time increased rather than decreased. The key frequently lies in the second part of our stipulation above: The parts of the original problem must be solvable separately. In many computer systems, we find that this is not so: in order to implement part A of the solution to the problem, we have to know something about part B ... and in order to solve part B, we have to know something about part C."*

The analysis of the Turban system development approach is a validation of these thoughts. That is part of the art in creating a system development methodology is to perceive and resolve the interdependence of the component parts in relation to the larger problem being addressed. On the other hand, does the methodology go down to the level of detail in which tasks are assigned to particular levels of staff - for example senior systems analysts are involved in requirements determination? Or should this aspect be left to the discretion of the project leader of the system development?

After consideration of a number of different issues discussed in the research literature, an assessment framework was developed and linked to the system development methodology continuum in the following way:

- **WHY (25 units)** in which: (a) an underlying philosophy is scored out of 15 units; and (b) a raison d'etre is scored out of 10 units;

- **WHAT (30 units)** in which: (a) producing an end result is scored out of 10 units; (b) provision of adequate tools and techniques is scored out of 15 units; and (c) provision of management guidelines and other practices is scored out of 5 units;
- **WHEN (35 units)** in which: (a) the framework's coverage is scored out of 15 units; (b) the framework's coherence is scored out of 15 units; and (c) indication of critical phases/stages is scored out of 5 units;
- **HOW (25 units)** in which: (a) documentation of the phases, activities and tasks is scored out of 10 units; (b) clear description of the application of tools and techniques is scored out of 10 units; and (c) provision of additional training and support material is scored out of 5 units.
- **IF (10 units)** in which documentation of the control mechanisms and quality assurance procedures is scored out of 10 units.

This approach was validated by using the assessment framework on three conventional development methodologies (namely ISAC, SSADM, and Boehm's (1988) spiral model of system development) as well as two KBS development methodologies (namely KADS, and ES/SDEM). As a result of this validation, the approach will be used to assess the development procedures and processes used by commercial organisations to determine whether KBS development occurred within a methodological environment or not.

In conclusion this literature review has supported the contention that there has been research interest (without what could be considered definitive answers) in all four major areas outlined in the first chapter of this report, namely:

(a) problems selected by commercial organisations to use KBS technology;

(b) the type of KBS technology applied in commercial KBS developments;

(c) the development approach used by commercial organisations in their KBS developments; and

(d) whether or not the development approach could be considered a system development methodology.

These areas of interest and the specific research questions associated with each area of interest will be discussed in more detail in the following chapter.

Chapter 3: Research Questions & Methodology

Chapter Outline
Research Questions
Research Methodology

Research Questions

In the first chapter of this project report, the following areas were identified as being important to determining how commercial organisations develop KBS's:

KBS Problems in Commercial Organisations

As also shown in the first chapter, in order to gain greater depth and consistency between organisations, the answer this rather broad question will be gained through the answers to the related questions shown below:

Question 1.1: **What are the major features of the problems being solved by KBS technology? For example, are these problems associated with mainstream issues for the organisation, or alternatively are they more peripheral to that organisation's core activities?**

Question 1.2: **How are these problems identified by commercial organisations as requiring a KBS solution? Are KBS developments initiated by the MIS or IT area, user areas, or another part of the organisation?**

Question 1.3: **Are the problems long-term or short-term, that is does the KBS development address problems with long lives, or alternatively are these systems developed to address issues of limited life?**

Answers to these three questions should provide greater insight into the level of commitment the organisation has to the use of KBS technology. For organisations with a high level of commitment to the technology, typical answers to these questions would be:

Answer 1.1: Problems are selected that relate to mainstream activities, for example for a university the problem addressed is determination of graduand status for a final year student, or the KBS solution has a significant impact on the operations of the organisation in question.

Answer 1.2: The MIS or IT area has an active involvement in identifying potential situations suitable for KBS development.

Answer 1.3: KBS developments are related to projects with extended lives.

On the other hand problems that have little relation to the organisation's core activities, were identified by user areas, and are only considered to be short-term problems would indicate a lower level of commitment to the technology by the organisation.

Question 1.4: **Are there similarities between the problems solved using KBS technology for different organisations?**

As already discussed in previous chapters of this report there will be obvious functional differences evident from the different environments within which organisations find themselves, for example organisations involved in transport and distribution will have problems that are markedly different at a functional level from a public sector administration organisation. Consequently, the comparison will occur at a broader level. A number of different comparisons could be made, such as the Hayes-Roth et al. (1983) classification discussed in Chapter 1: Introduction. However, the principal comparison will be on the level of decision making the KBS is targeted, and the level of compulsion directed at users of the commercial KBS. This analysis should provide valuable information that could assist in improving systems development procedures for commercial KBS's. If the typical user of a commercial KBS is operational or clerical staff, then further research may unfold better techniques to involve these users in the overall development process. If as a result of completing other comparisons it was found that the majority of commercial KBS's are all directed at diagnosis type problems, then the KADS development strategy, with its library of generic interpretation models, may offer a significantly better development strategy than the one used currently.

Use of KBS Technology in Commercial KBS's

In a similar fashion to the first question, the answer to this question will be derived from answers to the subsidiary questions outlined below:

Question 2.1: **How was KBS technology "introduced" into the organisation? Was it an initiative of the MIS or IT area, or some other area of the organisation?**

This question like others discussed earlier would again give insight into the level of commitment that an organisation has for KBS technology. If the MIS or IT area has introduced the technology, then more than likely this area will promote the use of the technology and therefore the organisation as a whole will have a higher level of commitment. On the other hand, if a user area is championing the technology, there may be an adverse reaction from MIS (which sees the initiative being taken away from it) and accordingly this would lead to the organisation having a lower level of commitment.

Question 2.2: **What designs are employed in developing the major components of a commercial KBS? For example, are most KBS's built with knowledge represent as production rules, and forward or backward chaining for the inferencing strategy?**

A similar problem exists for any analysis performed here as it did in terms of determining similarities in the types of problems addressed by commercial KBS's. At a fine level of detail, there will be obvious differences such as the type of knowledge stored within the knowledge base. Accordingly, a similar strategy to that used for Question 1.4 will be employed here, that is the comparison will be abstracted to a higher level. For example, rather than comparing the actual contents of the knowledge base, the comparison will be abstracted to the type of formalism used to represent the knowledge within the knowledge base. The other problem in performing the analysis is what to compare. At too fine a level of detail, there may be features in one KBS that are absent in another and so comparison is at best difficult and misleading or at worst impossible and contradictory. Accordingly, comparisons here will be confined to commonly found components of a KBS, namely the knowledge representation used in the knowledge base, the inferencing strategy encapsulated in the inference engine, and characteristics of the user interface.

This sort of analysis is also considered to provide useful information with which to improve KBS development processes. As an illustration, if the comparison found that the majority of commercial KBS's represented knowledge using production rules then this fact would mean that KBS development methodologies could be tailored. That is, the methodology would only require a fairly restricted portfolio of knowledge acquisition tools and techniques to complete this phase of KBS development. The analysis may also give an insight into the applicability of proprietary development methodologies such as KADS. If most KBS developments produce systems that contain production rules, use forward or backward chaining, and have no explanation facilities, then there is a higher chance of creating a generic KBS development methodology than if the reverse was the case. Furthermore, development of such traditional KBS's would indicate that commercial organisations have adopted a quite conservative attitude with respect to the introduction of this technology.

Question 2.3: **Are knowledge based system shells the major delivery environment for commercial knowledge based systems? Alternatively, are KBS's coded systems written in either artificial intelligence languages such as LISP or PROLOG, or a more conventional language such as C (or C++)?**

This question is deemed to be important because of the impact on the development process. Is expert system shell software powerful enough to be more than a rapid prototyping development tool? If it was found that few commercial KBS's were developed and then delivered within an expert system shell, this would indicate that this software is not suitable for this purpose and could lead to wasted development effort in trying to convert the final version prototype into a production system. Furthermore, the answer to this question has KBS development methodology implications. That is, if the analysis found that few production systems are written in expert system shells, then a new and important checkpoint or milestone should be introduced into the KBS development methodology. That is, similar to feasibility studies carried out at the beginning of the project, there should be an assessment of the expert system shell's capacity to deliver a production system.

Question 2.4: **What has been the goal in using KBS technology? For example, are commercial KBS's seen by their developers and/or users to be experts, colleagues, or merely assistants?**

To some extent this question would provide insight into the maturity of this technology as a development vehicle. For instance, if most commercial KBS's are judged to be assistants rather than experts, this would signify either that organisations are still experimenting with the technology, or it would mean that for commercial organisations there are more opportunities to develop these types of systems. If the latter point is true then this would have implications for some important KBS development phases, particularly knowledge acquisition and verification processes, as well as the need to build comprehensive explanation facilities. Another issue to resolve is whether the perceived objectives for commercial KBS development evident within the research literature match up with those of actual commercial developments. Rook and Croghan (1989) have stated that the major objectives of KBS development within a commercial environment would be that [p. 587]:

> *"[a] KBS is to function as a powerful, problem solving system whose interface inspires confidence and acceptance ... [which] ... must operate day-to-day over a prolonged period of time with minimal maintenance".*

This would indicate that commercial KBS's would be more at the colleague and expert level rather than at the assistant level.

Question 2.5: **How are the final production versions of these KBS's used? For example, are they stand alone systems that run on microcomputers with a simple user-oriented interactive interface? Alternatively, are they highly integrated systems operating on a range of hardware platforms?**

The type of production environment for the KBS is also an indication of the importance of the technology. If in most cases KBS's are considered to be stand alone systems with little integration at present to other systems within the organisation, then this would indicate lesser importance than if the production KBS's had tight links to other systems within the organisation.

The Processes of Development for a Commercial KBS

These last two areas of interest form the core of the research project, and should provide greater insight into KBS development processes than would be available from the other survey style investigations.

Question 3.1: **What area within the organisation undertakes the KBS development? Is it the MIS or IT area, other areas of the organisation, or by persons outside of the organisation?**

Question 3.2: **What is the attitude of various areas with an organisation to KBS development - supportive, non-committal, obstructive?**

It is considered that there should be some form of linkage between these two questions. As an illustration, if the area undertaking the KBS development is not the MIS or IT area then this part of the organisation may not have the same level of enthusiasm and commitment to successful development as it would if the situation was reversed. Furthermore, did the attitude (either positive or negative) of other areas within the organisation have an impact on the KBS development? Finally, does ownership of the production KBS reside with the project sponsor or with the area that undertakes the KBS development?

Question 3.3: **What are the typical milestones of a KBS development? Are these milestones similar across different KBS developments by different organisations?**

Question 3.4: **What sort of documentation is generated from a KBS development? Are there any similarities in the documentation across different KBS developments by different organisations?**

Question 3.5: **What type of personnel are used to develop KBS? For example, do most KBS development personnel have a conventional IS background?**

These questions should provide an indication of two things. First, the applicability of generic development approaches would be enhanced if similar milestones and similar documentation was reported across different KBS developments. Furthermore, analysis of the data obtained in answer to these questions should provide some evidence on the level of similarity between KBS developments and conventional IS developments. Finally, what type of system development documentation from a management perspective is produced, for example formal reports regarding progress of system development, presentations or submissions?

It is considered that Question 3.5 is important to determine what impact do personnel have on the KBS development process. If KBS development teams are predominantly made up of personnel with previous IS experience, does this lead to more conventional development approaches than teams where this is not the case. Additionally, what impact on the development process occurs when KBS development teams are composed mainly of personnel from outside of the organisation? Finally, does the attitude of other parts of the organisation change on the basis of the make up of the KBS development team, or are other factors more important. For instance, is the level of commitment to KBS development higher when KBS development teams are predominantly made up of personnel drawn from conventional MIS or IT areas? Alternatively, do conventional IS personnel possess the necessary skills and experience to make an effective contribution to systems development?

Question 3.6: **Are KBS development projects more likely to be one-off activities or part of an integrated and continuing development program?**

Question 3.7: **Are KBS development projects large projects in terms of size of project development team, length of development schedule, and cost of development?**

The analysis at this point will focus on issues such as: does size alter the types of management procedures and documentation associated with the KBS development. Again problems may arise due the level of detail at which the analysis is carried out. At too fine a level of detail, comparisons may be difficult or impossible to pursue. Accordingly, the analysis will concentrate on issues such as: (a) funding and financial control procedures for the KBS project; and (b) the organisation line of control for KBS project, for such things as approval to commence project, authorisation to purchase hardware and/or software, authorisation to proceed further with systems development. In this case, it should be expected that an organisation with an on-going program of KBS development which is associated with large projects would have more formal management reporting and control procedures than if the reverse was the case.

Furthermore, it would be expected that if the organisation has an on-going KBS development program and that the KBS projects are significant then that organisation is more likely to store development metrics such as: (a) number of prototype versions developed to demonstration prototype; (b) total number of rules for each individual knowledge based system; (c) total man-hours of system development; (d) total system development costs; and (e) elapsed development time, that is from date KBS project formally commenced to date system placed into production.

Question 3.8: **What development approaches have been used in the construction of commercial KBS's? Are KBS's developed using an evolutionary approach, or is there evidence of a more structured development environment?**

Question 3.9: **Are KBS development approaches similar to those used in conventional information systems developments?**

It would be interesting to find out whether any of the KBS development tools and techniques have been applied to commercial KBS developments. This would indicate the level of interest by commercial system developers in the research and development work for this field. For example, are knowledge acquisition techniques such as KAAM (Rook and Croghan, 1989), or CDM (Klein et al., 1989) used within commercial developments, or are these too sophisticated for the type of KBS that is currently being delivered within commercial organisations. For instance, CDM was found by Klein et al. (1989) in two related case studies of fire ground command decision making to be an effective technique for eliciting or extracting tacit knowledge (as against factual knowledge that can be acquired with traditional techniques such as normal interviews) and perceptual learning (which covers things such as how do expert pilots fly aeroplanes well, and how expert fire fighters effectively extinguish fires in office buildings). However, if most commercial KBS's at present only include shallow knowledge then this technique may be inapplicable. In terms of validation and verification of the knowledge base, O'Leary (1990) proposes a somewhat unique technique in that it involves a third party controlling and directing the validation procedure, which proceeds through the following steps:

- *Face validity* - step-by-step review of the code, in which the third party validator deals directly with the expert; the system designer plays the part of a consultant.
- *Sub-system validity* - identifying and examining the revised prototype's assumptions and critical procedures, with the focus more on the prototype's details and specifics, in order to identify areas that may need further detailed development or revision.
- *Input-output comparison* - test the prototype to assess it's ability to supply responses comparable to the expert for certain combinations of events or test cases.

With respect to Question 3.9 the analysis here will focus on whether similar development phases occur within commercial KBS development such as the presence of feasibility studies, analysis and design phases, system verification and validation phases. Furthermore, the analysis will also consider the types of activities undertaken during these phases by virtue of the documentation produced.

Development Methodologies for Commercial KBS's

It is considered that this will be the hardest question of this research project to resolve. Even though the analysis will be assisted by the system development methodology assessment framework developed in the previous chapter, in many cases objective documentary evidence may not be available, and therefore a reliance on more subjective and accordingly less reliable, people's perceptions will be required.

Question 4.1: **Do commercial organisations have organisations possess KBS development methodologies?**

The real problem here will be to differentiate between a methodical development approach and a development approach the adheres to a systems development methodology. In addition to an appraisal of the development approach by the SDM assessment framework, some of the other factors that would indicate the latter case are considered to be:

- documentation exists that outlines the methodology in some detail;
- actual KBS development is structured around the methodology, and there are regular checkpoints during KBS development to ensure that the methodology is being followed;

- the methodology was established prior to the organisation embarking on KBS projects;
- the methodology is subjected to a review process at the end of each KBS development; and
- the methodology is generally recognised throughout the organisation as being the appropriate practice to adopt in the development of these systems.

Other evidence, if available, would be actual systems development documentation that is prescribed by the development methodology exists, that is minutes of steering committee meetings, planning documents, or evaluation and feasibility reports. There may also be standards promulgated by MIS for conventional system developments, for example quality assurance procedures, that have been applied to KBS developments.

On the other hand, an ad hoc process would not exhibit the characteristics outlined above, as well as displaying the following:

- the development processes and procedures, and the order in which they are executed chronologically, change across different development projects;
- there is little emphasis within the organisation on standards with respect to the development of KBS's, that is a particular methodology may be used but it depends on either the personal preferences of the KBS project manager, or has been constructed by the individual KBS project manager; and
- there has been no desire by the KBS group to standardise development within the organisation.

Question 4.2: **If the organisation does not have a KBS development methodology, are there any reasons for this?**

Some of the reasons for this may be as follows, on whether or not an organisation either creates or acquires and then applies a methodology for KBS development. It may be that KBS development is very irregular, and so the effort to develop, customise and ensure adherence to a methodological approach is not seen as cost effective. Furthermore, if the case study reveals that most KBS's are stand alone systems rather than integrated ones, this may indicate less of a need for a formal development approach. If the organisation has a perception that KBS development is less important than conventional IS development, this may mean there is less of an organisational imperative to follow a methodological development approach. Finally, if the management and control of the KBS group lies outside of the MIS area, that is the KBS group is part of the Research and Development area, this location may have less of an emphasis on a formal systems development approach.

Question 4.3: **If the organisation does have a methodology, what features do these methodologies have? For example, do these methodologies follow those that have been described in the research literature, follow the design of conventional systems development methodologies, or are unique in their structure?**

It would be interesting to determine the impact that research on the advancement of system development methodologies in general and KBS development methodologies in particular have had on commercial KBS development approaches. For instance is there a high level of awareness amongst KBS project managers of these research efforts?

Question 4.4: **If the organisation does have a KBS development methodology, what sort of documentation or other material exists describing the methodology? In addition to documentation, are there other forms of educational support for the use of the methodology, for example training courses?**

As indicated above, evidence of documentation would support the contention that the development approach is really a systems development methodology; even more so where formal training either internally or externally occurs with respect to this development approach. On the other hand in situations where documentation does not exist and the training process is very much an informal one, then this would support the opposite conclusion.

Question 4.5: **If the organisation does have a KBS development methodology, how many times has this methodology been used?**

A systems development approach would be considered more likely to be a systems development methodology if it can be demonstrated that the approach has been applied consistently across a number of system development projects. However, while this is an indicator it should not be seen as a necessary and sufficient condition which establishes whether or not a particular development approach should be considered a methodology.

Question 4.6: **What impact does a KBS development methodology have on the number of KBS developments? For instance, are organisations that have development methodologies more likely to developed more KBS's than organisations that do not?**

Other issues raised by this question are things like: (a) comparing the attitude of the organisation to KBS technology for those organisations with a KBS development methodology as against those organisations that do not have such a product; (b) the quantity and detail of the documentation produced during the KBS development; and (c) the perception of level of professionalism and effectiveness of the KBS development group by other parts of the organisation.

Research Design

Outline of Research Methodology

The research was conducted using a case study approach. Initially a list of organisations was created that were known to have adopted and applied KBS technology for more than two years. This list was then analysed further to determine what organisations would fall into the following categories:

- Organisations that had successfully completed one or more KBS developments, that is the full development life cycle had been completed AND a production system delivered which was either currently still in use, or had been used within a production environment for the intended life of the KBS. In the first case obviously the KBS was providing a solution to an on-going long-term problem, whereas in the second situation, the KBS addressed some short-term problem and therefore had only a limited life span. Where possible the organisation should also have developed a number of production KBS's that addressed different problem and knowledge domains, were constructed within different development environments, and deployed into different production environments.

- Organisations that were still investigating the technology, perhaps developing systems where the emphasis was more on raising the level of awareness within the organisation of what the technology could do, that is the objective of KBS development was not a production system but more of a technology demonstration. Accordingly, for these organisations it was considered that KBS development would not have progressed through the full development life cycle, and in some cases may not have advanced past conceptual or experimental prototype stage. In addition, for these types of projects early phases such as problem identification and selection may have had on cursory attention, or with regard to say feasibility study and assessment, not performed at all.

Out of the two categories, organisations were not selected from the latter or second category, as it was considered that at best less than meaningful answers to some of the research questions outlined above would not be forthcoming, or at worst the answers obtained would confound rather than clarify the research issue. As an illustration, if an organisation was still in the process of developing conceptual or experimental prototypes, then the answers to Question 3.6 would more likely be one-off activities which are not part of an integrated and continuing development program, whereas the opposite may be the case for those organisations that had completed and deployed production KBS's. Completion of the development process was required in order to make valid comparisons across organisations and to assess development procedures over the full development life cycle, rather than comparing developments that may have ceased at different stages.

In a similar fashion, for Question 3.7 organisations still "experimenting" with the technology would likely be involved with small development projects, small project development teams (say one or two persons at the most), leading to ad hoc development procedures and limited use of a system development methodology. Again in this situation, organisations that have delivered production KBS's would more likely to have the opposite experiences. Furthermore, some of the other research questions would not be applicable to this type of organisation. For instance, for Question 4.5, an organisation in this category would not have a KBS development methodology because it as yet had not delivered a production KBS.

Accordingly, any organisation identified as belonging to the second category was excluded from this research project. There was also a third category of organisations, namely those organisations which had been using the technology for less than two years. Again these organisations were also excluded for similar reasons outlined above. Notwithstanding this, in terms of further research, it is considered that some form of longitudinal case study for both of these organisations would be extremely relevant. For example, to track the changes in organisational attitude to KBS technology across time would provide great insight into the best strategies for other organisations to follow that are yet to introduce this technology. From the initial list the following organisations were selected:

Table 14. Organisations Selected for this Research Project

Name of Respondent	Organisation where KBS is located
BHP Information Technology	**BHP Steel (Organisation A)**
AMP Society	**AMP Society (Organisation B)**
CONTINUUM	**GIO Australia (Organisation C)**
Computer Power Group	**Commonwealth Govt. Dept. (Organisation D)**

Name of Respondent	Organisation where KBS is located
State Bank of NSW	State Bank of NSW **(Organisation E)**
Australian Taxation Office	Australian Taxation Office **(Organisation F)**
WESTPAC	WESTPAC **(Organisation G)**

Using the above list, an initial contact was made with the project manager or some other principal in charge of the KBS development - either for a current development that was proceeding through the final phases of delivering a production system, or a development which had recently completed delivery of a production KBS. The list of persons contacted is shown in the table below:

Table 15. Interview Contacts for this Research Project

Name of Respondent	Particulars
Organisation A	Laurie Lock Lee, KBS Development Manager, (049) 40-1602
Organisation B	Kathryn Kennedy, Technical project manager, (02) 685-6962
Organisation C	Paul Beinat, Project manager (CONTINUUM), (02) 228-1153
Organisation D	John Coughlan, Applied Research and Development Consultant, (06) 283-6777
Organisation E	Sue Zawa, Systems Consultant, (02) 798-1073
Organisation F	Len Carver, Director - KBS, (02) 266-0688
Organisation G	Mike Barrett, System Development Manager, (02) 319-2767

For all organisations, the data gathering component of the case study was completed in a face-to-face interview situation, most of which were conducted in a single session and lasted on average about three hours, although in three cases multiple contacts were required to complete this phase of the research. This method of data collection was considered a far better research approach than the alternative of a mail survey, as it allowed on the spot clarification and justification of answers given. More importantly, in all cases the respondent allowed the interview to be tape recorded. This meant that the tape recording could be checked at a later stage to assess the reliability and veracity of the responses recorded in question. Furthermore, as in most cases the interviewee would explain the reasons for the answer given to a particular questionnaire question, this background or interpretational material provided a greater "richness" to the information collected on the questionnaire thereby increasing the internal validity of the research.

The focus of the interview was a questionnaire which is shown in Appendix 3. The questionnaire had three major parts, namely Part I - Organisational Details, Part II - KBS System Details, and Part III - KBS Development Methodology. Parts II and III were completed with respect to a particular KBS development, that is the delivery of a KBS to the production stage. This strategy was chosen for two reasons: (a) in most cases the organisation had only completed development of one or two production KBS's; and (b) a particular development approach was identifiable so that comparisons could be made across organisations. In most cases this particular KBS development had just entered the production phase or been completed within the last 18 months. Given the dynamic nature of this field and changing level of interest by organisations in KBS technology, it was considered that systems older than 18 months would not reflect current attitudes and development processes. Accordingly, detailed information on the development of 7 commercial knowledge based systems was collected in the interview phase.

On the other hand Part I of the questionnaire was completed within the context of KBS developments in general for that organisation. Finally, in addition to the questionnaire material and the tape recordings associated with gathering answers for the questions contained in the questionnaire, each respondent was given a general but directed interview concerning the attitude of the project manager to systems development methodologies and various other issues associated with the particular KBS development.

The tape recording for each interview were transcribed and these transcripts were then used to check the answers in the completed questionnaire. Finally, a completed questionnaire and interview transcript was mailed back to the respondents for them to review. Respondents were also asked to indicate anything they felt in either the questionnaire or the interview transcripts was of a confidential nature. Replies were obtained from two respondents regarding changes to data as well as indications of areas considered sensitive or confidential. Where no replies were obtained, respondents were either happy with the material or had previously identified information that they considered was confidential.

Construct Validity: A Multiple Case Study Approach

Out of the various research techniques, the case study method was judged to be the best strategy to pursue this research, on a first principles basis. For example, this technique allowed meaningful answers to be deduced for the research questions outlined above. From an operational point of view it was important to have extended contact with the actual knowledge engineers or KBS project managers involved in real KBS development, to gain first hand, their impressions, information, and perceptions about how the rest of the organisation was coping with the introduction and use of this technology.

The same result in terms of depth of knowledge could not be gathered using the questionnaire as a survey instrument. For example, as the respondent filled in the questionnaire the interviewer could either ask clarifying questions or provide additional information to improve the answer for that question. As a result the survey would have revealed only superficial information to research issues that are deemed to have a great amount of depth. Furthermore, due to the limited number of organisations suitable for further investigation (seven in all) it was thought doubtful that any significant statistical trends would emerge from an analysis that could possibly be undertaken within a survey of a larger number of organisations. Finally, some research issues are considered quite complex, and so would have required extensive notes and additional comments to the respondent in terms of filling out the questionnaire. For example, to identify whether organisations possess formal KBS development methodologies, would have required extensive notes on first what constituted a methodology, and secondly on what was thought to be a formal development methodology.

Given the research questions outlined above, it was thought that a single case study approach would not provide great insight into how commercial organisations develop KBS's. The reasons are as follows: (a) there is no "generic" type of organisation involved in this development. As the table above shows there are developments occurring within manufacturing, banking and financial services, insurance, and public sector organisations. Furthermore, there are a number of different development tools and development approaches, none of which appear on the surface to be any better than the other. There are a variety of delivery mechanisms that could be used for the final production KBS, ranging from stand alone systems that operate independently of other systems and programs, to fully embedded systems that are only invoked and utilised by programs belonging to a particular system. Finally, the level of users for the KBS technology differs dramatically across organisations from bank managers making commercial credit decisions (within Organisation G) to clerical staff involved in answering customer enquiries (Organisation F).

Given this inherent diversity, with a single respondent there would be no facility to cross check information and evaluate whether the picture that emerges for that organisation has any relevance to the development processes undertaken by other organisations. That is, a key part of the research is to identify whether or not there are similarities in the development approach across commercial organisations that are functionally significantly different. Furthermore, although the questionnaire design has included a various points internal checks of there is still the possibility (given the lack of general knowledge in this area) that a single respondent could withhold information, embellish information so that the organisation appears to be doing something it is not, or just forget that something had happened. Given more than one organisation, the ability to cross check information and carry out follow up interviews using this analysis would greatly increase internal validity.

Accordingly, for these reasons a multiple case study approach was adopted. It was also felt that a structured approach to the respondent contacts was necessary given the multiple case study approach. As mentioned above, comparisons across organisations were made, and this could only happen with similar questions being asked at the same point in the interview. The focus of across organisation comparison will be the questionnaire as obviously actual answers for questions can be compared. Where possible additional semantic analysis of the interview transcripts was also performed. In fact, it is considered that the interview transcripts gave more insight into the reasons why KBS development proceeded in the fashion that it did within an organisation, or that the transcripts identified more clearly those important factors that influenced KBS development over the last two years, than the limited answers that are forthcoming from the questionnaires. For example, in terms of commitment by the MIS or IT area to the introduction of KBS technology the questionnaire provided a Likert scale score (which facilitated comparisons across organisations), however the interview transcript provided a great deal of background information on why the respondent felt that the particular score given was considered appropriate.

Why an Experiment Was Not Appropriate

Inherently the type of research being carried out, does not lend itself to the use of the experimental technique. For example, there research issues concern real world and not artificial laboratory type issues. Neither is the research trying to obtain any measures of performance within a controlled environment. Use of students, either undergraduate or graduate/masters students would be inappropriate because these students would have either little or no experience in the development of KBS's.

Judd et al. (1991) also provides the following reasons why the use of experiments would not be suitable for this research [p. 174]:

> *"When research goals are universalistic, since a particular setting and population of participants are not crucial aspects of the hypothesis under study, a laboratory may be appropriate."*

This research is not universalistic as it applies to a specific area of software development; the population of participants have to be carefully selected - that is, those involved in KBS development; the setting may have a significant impact on the results of the study, for example the organisation's overall commitment to the use of this technology. Furthermore, the research is not trying to examine what can happen under specific circumstances, but rather the opposite, does certain things happens under different conditions. The second reason advanced by Judd et al. (1991) concerning the suitability of a laboratory experiment is where [p. 175]:

> "... *Such research, in contrast to research aimed at determining what actually happens in a real situation, may be particularly well suited to the laboratory setting.*"

However, the principal aim of this research project is to determine what does happen in the real world, and the factors that operate within the real world which have some causal impact on why organisations have adopted a particular development approach to KBS projects.

Finally, Judd et al. (1991) state that a third reason where laboratory experiments are considered suitable is with respect to independent variables that can be manipulated and those that cannot [p. 177]:

> "... *For practical reasons, such variables as position in an organisational structure cannot be manipulated and are difficult or impossible to investigate in the laboratory.*"

The essence of this research is not to study the effects of what happens when independent variables are manipulated (notwithstanding the difficulty of doing this, for example changing the level of commitment within the organisation to using KBS technology) but to understand how the independent variables have impacted on the findings.

Why a Field Study Was Not Appropriate

A case can be made that possibly more definitive answers to the research questions shown above would be available from conducting a series of longitudinal field studies within a number of organisations. However, a number of factors render this research approach either inappropriate or difficult to accomplish. The length of time taken to perform the study (in one particular case, Organisation E, the KBS development was spread over six years). During this time advances in technology may invalidate the data obtained from the organisations and even the system developments that were selected originally within these organisations compared to developments currently under way. The fact that over time there may be changes in the measures or underlying systems development methodology (possibly due in part to the presence of the researcher), so that to some extent the original research questions are invalidated.

There may be exogenous factors that cause potential KBS developments not to commence, or existing KBS projects not to complete their development, such as the economic downturn (for example Organisation G has just dissolved their KBS area). With the small number of respondents identified as suitable, the effect within a field study situation of the organisation not completing systems development would be significant on the overall findings of the research.

Finally, this approach is considered to have no inherent advantages either in conducting the research, in the quality of the data to be collected, the range of analyses to be conducted, or in reliability of the conclusions reached as compared to using a retrospective case study approach - that is, a more inefficient method will not provide any advantages, and thus justify the extra effort.

Threats to Internal Validity of the Research

As indicated above, a flaw in the case study approach (and particularly with respect to a retrospective case study) there could be a tendency for respondents to fabricate, embellish, provide inaccurate information, or to forget important facts or contributions to questions set out in the questionnaire (this represents possibly the major threat to internal validity). In meeting this threat the overall design of the questionnaire is closed questions with a range of standard responses. In some cases the answers given to these closed questions are checked by asking a related open ended question, or the other way around. For example, Part III of the questionnaire on KBS development methodology has a closed question (the respondent can only answer "YES" or "NO" - although one respondent wanted to answer "MAYBE") on whether the respondent considers that KBS development with respect to the system in question followed a methodological approach. This is followed up by a question on the reasons to support this answer as well as other questions such as whether the methodology is documented, whether training is provided on the methodology and whether the methodology has been used on one or many developments. Accordingly, in key areas the questionnaire has a form of self-checking mechanism to ensure the accuracy and validity of the answers provided. As also indicated above, tape recordings were made of all respondent contacts. This allowed additional checks to be performed, in terms of the interview transcripts showing the responses to further questions being asked about a particular response to an answer within the questionnaire.

Another issue of concern was the length of interview contact.. In an field trial of the original questionnaire, interview time was about five hours (there were approximately 140 questions asked), which was considered too long a contact for the respondent to retain concentration on the questions being asked and also to remain enthusiastic about the research issues in general. Accordingly, as a result of this field trial, the original questionnaire was re-designed into three major components. Further assessment of the questionnaire was then performed with the objective of reducing its size. As a result of this process, two of these components were removed and may provide the focus for additional research at a later stage. The final questionnaire used within the case study contained three major parts, namely:

- PART I - Organisational Details
- PART II - KBS Project Manager Details
- PART III - KBS Development Methodology

This strategy was adopted to decrease the subjective amount of time perceived by the respondent in completing the questionnaire. As a result, in most cases there was greater co-operation between interviewee and respondent, and more consideration of answers to each question within the questionnaire, particularly with respect to the final section of the questionnaire. Furthermore, this structure allowed the case study to be completed for three contacts over a number of visits, where one part of the questionnaire was completed for each visit.

In retrospect, the data collected for these three organisations which required multiple visits may had been of higher value for the following reasons:

- the researcher had the ability to review and analyse the data already collected and therefore be in a greater state of knowledge concerning the issues for that organisation when the next part of the questionnaire was completed; and

- the time lag between contacts meant that the respondent did not remember specific answers given previously and so

> inconsistencies would emerge that were tackled at follow up interviews, once the whole of the questionnaire had been completed.

On the other hand, as the questionnaire spanned a very broad cross-section of information it was also important with the multiple contacts to ensure that respondents were aware of what had been covered in previous meetings.

Another threat to the internal validity of the research are problems with the completion of the questionnaire. For example, information was incorrectly recorded on the questionnaire during the course of the interview, either due to a transcription error in which the wrong box was ticked, or the wrong Likert score was circled. In addition there may have been some misunderstanding between what the respondent said and what the researcher thought the respondent said. Finally, in the course of the interview interruptions did occur and so there was the potential for some questions to be inadvertently missed.

In terms of meeting this threat, first the questionnaire provides the structure of the interview thereby minimising the chance of missing a question across the different respondents. If questions were missed, the tape recordings of the interviews were used to obtain an answer. In all cases, there was little need to re-contact the respondent in this area. Finally, as indicated above, completed questionnaires and full interview transcripts were returned to respondents so that these people could make a final check on the accuracy of the information provided. If changes were to be made, the respondents were to return either the interview transcripts or the amended questionnaires indicating the changes. Two respondents did return material, with minor changes indicated to both questionnaire answers and interview transcripts.

Another threat to internal validity was that the organisation had only recently adopted KBS technology and so had not moved from technology demonstration projects to production systems - may still use ad hoc procedures. Furthermore, although the organisation has a reasonably long history of developing KBS's, the project manager or knowledge engineer for the particular KBS development selected may be new to the area and so has no feeling for the appropriate KBS development methodology. With respect to the first point, the population of organisations was big enough to ensure that only organisations with more than two years experience in the application of KBS technology to systems development were included. More importantly all organisations had at least delivered one production KBS (two organisations were in this category), and in most cases had delivered more than one production KBS (with Organisation F delivering approximately 39 systems). From the other point of view, if it was not possible to eliminate organisations with less than two years experience with use of KBS technology, then a change of focus for the research would be in order. At the very least the research would now indicate that the application of KBS technology within a commercial environment is fairly immature and so the adoption of formal systems development methodologies have yet to occur.

With respect to the second point, it was found that for all organisations the KBS developments were managed by personnel with considerable experience both in conventional IS development as well as KBS development. For instance the project manager for Organisation A has been involved in KBS development for 9 nine years, and the project manager of the KBS development for Organisation C has 18 years experience in conventional IS developments with 8 years experience in KBS developments.

The final threat to the internal validity of the research was considered to be due to an ill-conceived structure to the questionnaire. That is, during the process of conducting the interviews, the questionnaire was found to have: (a) essential elements missing; (b) questions that led to imprecise or contradictory answers; or (c) allowed to much variation in an answer so that comparisons across organisations were rendered invalid. To counter this threat to internal validity the questionnaire was extensively analysed by the researcher and an independent reviewer. This process involved a substantial number of changes and revisions in the wording, and content of the original questionnaire. Furthermore, the final draft version of the questionnaire was piloted by an experienced KBS project manager to determine if there were any flaws in the design. As a result of this process, the questionnaire was significantly reduced, and minor changes made to the wording of either questions asked, or the "standard" answer options provided. During the execution of the interviews it was found that the questionnaire had a fair degree of robustness, with most organisations able to provide an intelligent answer to all questions contained within the questionnaire. In some instances further clarification of the specific questions was required by the interviewer, but this was the exception rather than the rule. Accordingly, as a result of the performance on the questionnaire during this phase of the research project, the threat to internal validity in this area is assessed to be low.

Threats to External Validity of the Research

As indicated above, the research is based on a multiple case study of seven organisations and so addresses one the major threats to external validity if only one organisation had been investigated. That is, the findings from the one organisation are not generalisable across other organisations which may have different development environments, and tackled different problem domains. However, even with a multiple case study approach, care was taken to obtain respondents from a number of different sectors and industries within the economy. With regard to this factor one organisation is involved in manufacturing operations, two organisations are general insurers, two organisations are banks, and the final two organisations are involved in public service regulation and administration of Federal Government legislation. Accordingly, it is felt that this represents a reasonable diversity in the range of activities performed by these organisations, although there is an emphasis in the financial services area (two banks and two insurance companies).

With respect to external validity, when looking at the types of KBS developments investigated, again there is a diversity in terms of the development environment, the type of production KBS delivered, and the size of the user population that interacts with the production KBS - the number of users interacting with the KBS ranged from as low as 3 to as high as many thousands. For instance, the case study research has indicated that both expert system shells (used in the development of KBS's by three organisations), as well as high level languages such as C++ (used in the development of commercial KBS's by three organisations) to construct commercial KBS's. Production KBS's operated as stand alone systems (in two organisations) while in the other five organisations these systems were either embedded systems (two organisations) or integrated with other systems (three organisations).

In terms of external validity and different problem domains, the following table indicates that in this area, diversity of problems encountered was evident:

Table 16. Types of Commercial KBS's Investigated

Name of Respondent	Description of KBS Domain
Organisation A	Blast furnace operational control (supervisory control)
Organisation B	Assist insurance agents in their dealings with prospective clients during the insurance proposal process.
Organisation C	Assessment of bodily injury insurance claims.
Organisation D	Assist in the enforcement and interpretation of Commonwealth government legislation, regulations and procedures.
Organisation E	Assess the suitability of a loan application for consumer type lending/credit.
Organisation F	Legal and administrative advisory system with learning.
Organisation G	Advise managers within wholesale banking areas on suitability of a loan prospect.

However, the problems do display some common features and so the findings of this research project may not be applicable to KBS developments that have the following characteristics:

- Where problem domains span organisational boundaries, and the user population includes people at different managerial levels. In all cases, organisations investigated in this research project delivered production KBS's that were targeted to a particular type of employee, and in most cases resolving low level problems within the organisation, such as assessing the creditworthiness of a company or an individual.

- Where KBS developments are judged to be large projects, involving development that absorbs significant amounts of personnel and other resources. In most cases, the organisations investigated in this research project had small development teams (the smallest being one person and the largest having an establishment of 15 full-time staff).
- Where KBS developments are geared to obtaining knowledge from identified experts within a particular problem domain to be used once that expert is no longer available to the organisation. In all cases for the organisations investigated the development of the KBS was not premised on the identification and co-operation of an expert.

Finally, for some areas of the research, the fact that seven organisations have been contacted, should ensure reasonable external validity in its own right. For example, if the investigation shows that all of organisations (or a majority of organisations) do not follow any prescribed development approach, then this would indicate that the research had reasonably high external validity regarding the use of a system development methodology for commercial KBS's. In this case the answer would be "NO". Greater problems for the findings with respect to this sort of analysis would obviously arise where the research finds a mixed result. Hopefully in these situations, the detail of information either available from the questionnaire or the interview transcripts will allow some rationalisation or insight to be drawn on those factors that have influenced some organisations to follow a more formal system development approach, while others have followed a more ad hoc approach.

Chapter Summary

This chapter delineated the four major research issues for this project, which were:

Question 1: What problems are solved by commercial organisations with the use of KBS technology?

Question 2: How is the KBS technology used or applied within the development of commercial KBS's?

Question 3: What are the development processes for a commercial KBS?

Question 4: What is the status of development methodologies for KBS's within commercial organisations?

Answers to these major research questions or issues will be provided indirectly by answering a series of more detailed questions all related to one of the research issues shown above. It is considered that this framework of questions will effectively address the overall thrust of the research, that is to obtain a greater understanding of how commercial KBS's are developed.

The chapter also discussed the research methodology from the point of view of construct validity and threats to both the internal and external validity of the research. With respect to construct validity, the analysis of the research questions indicated that the research goal would be accomplished and that the most appropriate research technique was being applied. In terms of both internal and external validity, the discussion assessed the level of threats that would impact on the validity of the research and the strategies adopted to limit or minimise those threats. Again it is considered from the arguments put forward in this section of the Chapter that the research had both high internal and external validity.

In conclusion, it is considered that there is a good match between the research questions outlined in the first part of this chapter, with the overall research design and methodology. In particular the case study approach was considered the best research design and the procedures involved in collection of the research data were considered to have minimised the threats to internal validity. In addition, both the questionnaire data as well as the interview transcripts are considered to provide a rich source of information that is analysed in the next chapter. The results and conclusions drawn from this analysis are discussed in the final chapter of this project report.

Chapter 4: Research Results and Analysis

Chapter Outline

Question 1: What problems are solved by commercial organisations with the use of KBS technology?

Question 2: How is the KBS technology used or applied within the development of commercial KBS's?

Question 3: What are the development processes for a commercial KBS?

Question 4: What is the status of development methodologies for KBS's within commercial organisations?

After completion of the interview phase of this research project, data had been collected on seven commercial KBS developments. These data consisted of completed questionnaires, and interview transcripts. Their was a strong relationship between the questionnaire data and the interview transcripts (which provided a richer picture than the bare answer given in the questionnaire) as each interview was structured around completion of the questionnaire. Before any analysis of these data was initiated, the questionnaire data and interview transcripts were reviewed, and then returned to the original respondent for a final check in terms of accuracy and other issues such as confidentiality.

Problems Solved Using KBS technology

What sort of the problems are solved by KBS technology?

For example, are these problems associated with mainstream issues for the organisation, or are they more peripheral to that organisation's core activities? The first point to make is that KBS technology has applicability across a diverse range of organisations and activities. In this research project there was one organisation involved in basic manufacturing, two general insurers, two banks, and two organisations involved in regulation and administration of Federal Government legislation. The table below outlines these organisations key activities:

Table 17. Activities of Respondent Organisations

Respondent Organisation	Organisation's Activities
Organisation A	Delivers IT services to parent company, which is involved in basic manufacturing.
Organisation B	Insurance and superannuation.
Organisation C	Insurance and superannuation.
Organisation D	Administration and enforcement of government legislation, regulations and other procedures.
Organisation E	Banking.
Organisation F	Administration and enforcement of government legislation and revenue collection.
Organisation G	Banking.

However, this diversity is dependent upon the level of granularity at which the comparison is performed. If the comparison is made at a higher level other than merely a direct comparison of activities, it is interesting to note from the above table that four out of the seven organisations are involved in the financial services area. At an even higher level, six out of the seven organisations deal with intangible products, that is are service oriented information processing organisations. Although a definitive trend cannot be asserted, due to the nature of the selection process to identify these organisations, there would appear to more opportunities for KBS development in these organisations, or a greater level of awareness of the benefits that KBS technology can offer when properly applied.

What types of problems are addressed by these organisations? The following table presents a description of the problem domain which the KBS addresses for the above organisations:

Table 18. Basic Function of the Commercial KBS's

System Name	Organisation's Activities
System A1	Blast furnace operational control and supervision.
System B1	Assist insurance agents in their dealings with prospective clients during the insurance proposal process.
System C1	Assessment of bodily injury with respect to insurance claims.
System D1	Assist in the enforcement and interpretation of Commonwealth government legislation, regulations and procedures.

System Name	Organisation's Activities
System E1	Assess the suitability of a loan application for consumer type lending and credit.
System F1	Legal and administrative support and advisory system with learning facilities.
System G1	Advise managers within wholesale banking areas on suitability of a commercial loan prospect.

Based on the data shown in this table, it would appear that in commercial organisations KBS technology has been applied to assist with core activities rather than more peripheral ones. For example, System A1 controls blast furnace operations for a steel mill, and System B1 provides assistance during the insurance proposal process between an insurance agent and a potential client of the insurance company. In both cases, the KBS appears to be addressing a core activity for that organisation without any reservation. Similar comments can be made about the other KBS's. It is also interesting to note another feature concerning the problem domain addressed by these KBS's, that is their quite narrow definition. Taking the two examples discussed above, the System A1 works for blast furnaces within the organisation, but has no broader applicability to other steel making operations undertaken, even at the same steel plant. Similarly, System B1 is not meant to be used even by other personnel who are involved in further processing of the insurance proposal after the insurance agent has completed his or her task. And for the insurance agent the system provides no assistance in other activities performed by this agent, such as client follow up, and marketing intelligence for new product offerings. Similar comments apply to the other systems delivered for the other organisations.

Are there similarities between the problems solved using KBS technology for different organisations?

At a first glance, there are some superficial similarities in the problems addressed. For example, the two banks (Organisations E and G) have used KBS technology to assist in verifying the credit worthiness of a loan applicant; for the two public service organisations (Organisations D and F) the systems were deployed to be used by lower level staff empowering them to make higher level decisions than would have previously been possible. On the other hand, even looking at these organisations, when these systems are examined in more detail there are also obvious differences. For example, System E1 is concerned with consumer credit loan applications while System G1 is concerned with commercial credit applications, which have entirely different issues to consider. Where organisations have markedly different mainstream activities there is either very little or no similarity between problems whatsoever. Accordingly, this indicates that: (a) organisations should not restrict themselves to identifying problems that other organisations have solved using KBS technology; and (b) even for organisations with similar core or mainstream activities, the problems solved by other organisations should be more a general guide rather than a specific problem that should be addressed.

Therefore, from a direct functional point of view there is an obvious diversity in the types of problems addressed by KBS technology. It is interesting to note that a functional comparison does not reveal greater commonality, say that all organisations have used KBS technology to improve their accounts receivable, accounts payable, or payroll operations. Either these areas are not a fruitful source of problems requiring KBS solution or the relative immaturity in the application of KBS technology within commercial organisations has meant a more ad hoc, less directed approach. With respect to this point, the respondent for System A1 made the following relevant comments [interview transcript, page 2]:

> *"Respondent: I think that at that time it would be fair to say that MIS manager (and he is still there) in [the organisation] ... his comments at the time we "By all means lets look at and explore this area" ... but he felt that at the time, that in five years time we would be able to buy products, and therefore it probably isn't a long term thing. But I suppose in that time his view has changed. I mean the inference was that if you wanted an expert blast furnace operator guidance system you would go and buy one. But it was found that this was not the case. Whilst there are probably twenty or thirty of them in the development stage around the world - none of them were in a package to an extent that we could just design the interfaces to it."*

Accordingly, as with the type of organisation, comparisons to identify similarities between problem domains will require abstraction to a higher level than simple comparisons of functionality. At a broader level, from the description of the organisation's principal activities and the description of the problem domain given in the two tables above, it would appear that two conclusions can be made about how commercial organisations have used KBS technology.

The first conclusion is that the problem domains at that the KBS decision support is provided at the operations level, giving assistance to well defined problems either on a continuous or at least a daily basis, rather than addressing the less well defined (semi-structured or unstructured problems) that are the concern of higher management level positions within the organisation. For instance, in addition to the systems described above, System C1 provides assistance to a relatively low level person involved assessing the amount of compensation that should be paid to someone for a bodily injury who has made a claim. This system, although inherently complex from a functional point of view in terms of its internal operation, is invoked by many assessors many times a day for the thousands of claims received. It is not involved in assisting higher level management staff for that organisation in the review or development of new insurance products, except in provision of statistical data that may be used in this higher level process. Similarly the systems used within the banking organisations are concerned with operational decisions about whether a person will have a credit or loan application approved, rather than assisting in the design or delivery of new commercial or consumer financing products.

Accordingly, although there is a high degree of difference at a functional level, when the comparison process is abstracted, there appears to be a high degree of similarity. That is, the problems addressed by the organisations selected for this case study are mainstream problems that are focussed upon operational rather than higher level management type problems. The second conclusion about the problem domains that KBS decision support is addressing is the level of direction or compulsion placed on direct users of the KBS. For this dimension, it would appear that no common approach is found at the functionality being delivered by the KBS, or from an organisational imperative dominating with respect to this factor. For System A1, users do not have an option and are therefore compelled to use the system or the results produced by the system. Alternatively, with respect to System E1 the organisation has directed that all consumer loan applications will be assessed using the

KBS. For the other systems there is a greater degree of choice about whether the system is used or not. With respect to System C1 there is no requirement either from a functional or an organisational point of view to use the system. However, the obvious advantages provided by the KBS mean that it had almost universal acceptance. Indeed the respondent for that organisation commented that new assessing staff would often state the difficulty they would have in assessing claims without the assistance of the KBS. Therefore in three cases there is a high degree of direct or implicit compulsion to use the KBS.

For the other four systems, namely Systems B1, D1, F1 and G1, their use is much more optional in nature. For the two public service organisations users had the option of using the KBS or not. In many cases users were not aware that they were using a KBS - for System D1, the KBS was just another icon on a Microsoft Window™ screen. For System F1, the KBS was specifically designed so that users would access the system less and less as their own knowledge and skill increased. Furthermore, the respondent for System F1 expected that the KBS would be de-activated about eighteen months after deployment (although this decision was up to the project sponsor and not the project manager). For System B1, insurance agents being independent of the insurance company could not be compelled to use the KBS, although there are indications of increasing usage because of the perceived advantages that the KBS offers to these agents. Finally, System G1 was an optional system, and a lack of support from senior management and marketing of the system has meant that the KBS slowly disappearing from use within the organisation (at the moment the KBS has three users located at suburban branches in Queensland).

Therefore, this aspect demonstrated a high degree of difference between organisations, and even difference (excluding the public sector organisations) between organisations within the same industry, such as for the insurance and banking organisations.

How are problems that require a KBS solution identified by commercial organisations?

Are KBS developments initiated by the MIS or IT area, user areas or another part of the organisation? A high degree of comparability occurs with respect to this question. In six out of the seven organisations the MIS or IT area did not have an active involvement in identifying potential situations suitable for KBS development - the exception being Organisation E. The areas that identified problems were either the user area (Organisations C and D) or a special technology/research and development section within the organisation, either with the assistance and co-operation of the user area (Organisations A, B, and G) or on their own initiative (Organisation F). In most cases this lack of involvement by the MIS or IT area was not only confined to the early stages of KBS development, but was evident throughout development - even to the point of KBS deployment or implementation. The reasons for this varied, and were in the main contingent upon the type of problem being considered, and the reasons or environment in which the problem had emerged.

For example, in Organisation F, the problem arose because a major division of the organisation (comprising approximately 4,000 employees) was undergoing a fundamental re-structuring of its work practices. Senior management believed that some form of computer-based support should be provided to those staff that were re-located to new positions for which they lacked skills to perform the task effectively. As a result of this view, the organisation recruited an outside person (the eventual respondent for that organisation) to head up an investigation into what could be done to alleviate the problem. A report by this person recommended the use of KBS technology and to establish an independent KBS development cell which would deliver KBS products into that division. For this organisation, according to the respondent, the MIS or IT area played nothing more than a supportive role during this process. The innovation and drive to introduce the new technology was well founded within the user area. A similar approach occurred in the other public sector organisation (Organisation D), with the difference being that development and delivery of the KBS occurred using outside consultants.

With respect to Organisation C, a senior manager within a user area was concerned with the increasing amounts being paid on insurance claims for bodily injury. This person, who the respondent stated eventually became the organisation's champion for the use of KBS technology, after looking at a number of different options (such as additional training of assessors or using standards in the assessment process) decided that possibly using KBS technology would be the best option. The respondent for Organisation C states that this eventual champion of KBS technology did not rely on help from the MIS area as it was perceived generally throughout the rest of the organisation as not being very effective. Instead the manager contacted an outside consulting organisation which then employed the respondent to build the KBS for Organisation C.

Another area that often provided the impetus to using KBS technology to solve problems in commercial organisations was what can be loosely termed the advanced technology or research and development area, often reacting to a user area request. For example, the respondent for Organisation G, who was placed within the Research and Development area, held an informal breakfast presentation on KBS technology for senior managers within user areas of the organisation. At the end of the presentation participants were requested to leave business cards if they were interested in applying the technology within their respective areas. One such manager's interest led to development of the KBS formed the system selected for investigation in this research project.

For Organisation A, the approach was a mixture of user initiative and the research and development area agreeing that the problem had a potential solution in applying KBS technology. In fact quite a mature process governs the development of KBS's within this organisation, probably due to the fact that knowledge based systems had been developed in this organisation for over nine years. The initial prototyping phases are performed within the research area, and then at some point the system is handed over to the IT area to complete development. This nexus is normally when the final version prototype is complete and the project is about to enter a re-development phase to deliver the system into the production environment.

Accordingly, two conclusions can be inferred from the above discussion. First, in most organisations the activities to identify problems suitable to a KBS solution occurred using personnel outside the MIS or IT area. Second, the procedures adopted by those personnel appear to follow a unstructured, ad hoc process in the sense that there is little in common across the organisations. In many cases, the use of KBS technology emerged as a solution to a problem rather than the problem being targeted as one that could be addressed with a KBS. It is interesting to compare this approach with that employed by the exception - Organisation E. In this case, the respondent who was in a research and development area of the IT division initiated a survey of the organisation to determine likely candidates for application of KBS technology. The results of this survey revealed 42 candidate projects throughout the organisation, from which the respondent selected one most suitable for the first KBS development. In this case a more structured approach to the identification of the problem domain occurred in stark contrast with the other six organisations.

How was KBS technology "introduced" into the organisation?

Was it an initiative of the MIS or IT area or some other area of the organisation? The reasons why each selected organisation became interested in KBS technology, and the principal area that introduced the technology to the organisation are shown in the table below:

Table 19. Reasons KBS Group Formed - Respondent Organisation

Name of Respondent	Reason for forming KBS group was ...
Organisation A	In 1984 KBS was a new research area. The research and development area became interested in expert systems and artificial intelligence with the establishment of an AI group in the IT division of the CSIRO. The organisation felt that it should do something similar.

Name of Respondent	Reason for forming KBS group was ...
Organisation B	An expert systems group was formed to research and investigate possible expert system applications within the organisation in general, as well as to pursue the development of specific expert system applications, such as assistance in the insurance proposal process to eliminate a lot of the paperwork.
Organisation C	The organisation's user area perceived that the MIS area did not have the required expertise to develop the KBS.
Organisation D	The client organisation did not have the expertise to produce system. A consulting firm was contracted to produce a turn key KBS for the client on a contract basis.
Organisation E	The organisation decided to that it needed to explore the possibility of using KBS technology, and hired a full-time KBS expert to do the initial start up KBS work.

Name of Respondent	Reason for forming KBS group was ...
Organisation F	There was an interest by senior management within the Perth office of a major user division of the organisation, as well as very senior management in the head office area, to look at decision support technology. The focus of this decision support was systems aimed at customer services. As a result of this interest advice was sort from another organisation involved in KBS development as well as from an academic and research centre based in Canberra. This effort resulted in a report which outlined the actions and referrals needed to begin KBS development within the organisation. As a result of this report a KBS development cell was created within the user division discussed above.
Organisation G	The Research and Development area of the organisation, which had a focus on R&D rather than delivery of working systems, thought the area one worthy of investigation. Initially the operations of the KBS unit was speculative funding, geared primarily for the development of small systems.

This question is considered at this point because of the link between the initial introduction of KBS technology into the organisation and the identification of problems suited to a KBS solution. In fact in the majority of organisations (five out of seven) the two were linked, as the problem that had been identified was also the first attempt by the organisation to use KBS technology (Organisations C, D, E, F and G). With regard to these organisations, areas of the organisation other than the MIS or IT area were primarily involved in the introduction of this technology for the reasons stated in the section above. The exception in this group was Organisation E with the MIS or IT area being involved, although the respondent indicated that the level of interest by this area in using the technology was initially quite high but then faded away over the next few years.

For the other two organisations, Organisation B had a little more experience in that an expert systems group existed with the goal of research and investigating the possible applications of this technology within the organisation. As a result of this effort, the group had been involved in one other KBS development. However, this group went through a number of major staff changes, as well as being re-located twice into different divisions of the organisation (the name of the group also changed), as interest in applying the technology and availability of funds both diminished. In fact the respondent for this Organisation stated a number of times that it was a miracle a group involved in a KBS development project survived at all.

Accordingly, given the staff and organisational changes little previous experience and expertise remained within Organisation B, as the group re-invented or re-created itself. Therefore, to all intents and purposes, Organisation B could be classified with the other organisations in which the introduction of the technology was the first attempt by the organisation to use KBS technology in solving a particular commercial problem. The exception in this area was Organisation A which had a completely different impetus to investigate the use the technology, that is it wanted to match the interest that the CSIRO was devoting to KBS technology. However, similar to all the other organisations (except for Organisation E) it was again a research and development area that introduced the technology rather than the MIS or IT area. Organisation A was also different to Organisation B in the sense that this area experienced none of the upheaval and change but was rather relatively stable across a long period of time (from 1984 to about 1991).

In summary therefore, as shown in the table below, in six out of the seven organisations, the introduction of KBS technology was relatively new, and these organisations could be classified as "feeling their way", even though production systems were delivered in all cases. Accordingly it would appear that the major reaction by the MIS or IT area within these organisations was very much a "wait and see what happens" attitude, with the MIS or IT areas having very little involvement in either the introduction of the technology into the organisation (six out of seven organisations), with Organisation E again being the exception organisation.

Table 20. Area that Introduced/Developed Commercial KBS

Organisation	MIS/IT	R & D	Operations	Head Office	Other
Organisation A					
Organisation B					
Organisation C					
Organisation D					
Organisation E					
Organisation F					
Organisation G					

The respondent for Organisation F summed up the approach in this fashion [interview transcript, page 3]:

> *"Ian Caddy: Right. So, was it research and development that ...*
> *Respondent: It was set up ... that is ... it was set up as an R & D cell. So in other words the [Organisation] was saying if this technology delivers nothing, then that is OK. In fact the guidance that we got from the consultants who drove the knowledge based systems project ... the study ... was that we should purposely take on high risk projects."*

Following on from this it is interesting to note, from the respondent's point of view, the level of difficulty experienced in trying to introduce KBS technology. Respondents were asked to provide a Likert score ranging from "1" (extremely difficult to introduce the technology) to "7" (extremely easy to introduce the technology). The results from this question are shown in the table below:

Table 21. Difficulty Experienced in Introducing KBS Technology

Organisation	Within MIS/IT	Outside of MIS/IT
Organisation A	4	6
Organisation B (a)	6, 2	6, 3
Organisation C (b)	N/A	7
Organisation D	3	5
Organisation E	6	1
Organisation F	2	5
Organisation G	2	4

(a) Initially "6" for both areas; other figure represents current situation.

(b) For this organisation the MIS area was almost completely ignored and so no value could be assigned.

The table below presents a comparison of the Likert scores (excluding Organisation C and using the current score for Organisation B) between those organisations where the MIS/IT area introduced the technology as against those in which the MIS/IT area did not:

Table 22. Comparison of Difficulty in Introducing KBS Technology

Area introduced KBS technology	Difficulty experienced	
	MIS/IT	Other
MIS/IT	6.0	1.0
Other	2.6	4.6

Accordingly, although the number of data points prevents any reasonable statistical analysis to be performed, it would appear that two conclusions can be inferred from this table. First, for these organisations, the area that introduces the technology has a reasonably high level of commitment (a score of "6" which indicates a very low level of difficulty, and a average score of "4.6" which indicates a moderately low level of difficulty). Furthermore, if it is the MIS/IT area then this level of difficulty is even lower ("6" as compared with "4.6"). Second, there are big differences in the level of difficulty experienced between the area that introduced the technology, and either the ultimate user of the KBS or the MIS/IT area. For one organisation there was a difference of 5 (a Likert score of "6" compared with "1") and an average difference of two for all the other organisations. This would seem to indicate the potential for conflict or disagreement to arise between these two areas in the course of the KBS development.

Are the problems addressed by KBS technology long-term or short-term?

That is, does the KBS development address problems with long lives or alternatively are these systems developed to address issues of limited life? For six organisations out of seven (Organisations A, B, C, D, E, and G), the KBS developments are related to problems that have extended lives. The problem that the KBS deals with is an on-going one and in terms of use, the KBS is invoked frequently or continuously during a typical working day. Whether the KBS solution remains the best answer in the future is not within the scope of this research. For Organisation F, the KBS was designed to have a short life expectancy, as it was expected that the problem itself had a short life expectancy (the respondent had estimated the system would be in use for about eighteen (18) months). This KBS was meant to provide assistance and knowledge to users handling customer enquires. It was expected that over time KBS users would gain the necessary skills, knowledge and experience to effectively perform their new duties without the need to resort to accessing the KBS.

Given these data there are two conclusions that can be inferred. First, even though organisations appear to be still in an experimentation phase with the use of this technology, the problems addressed are mainstream and on-going. That is commercial organisations are looking for a suitable return on investment in the systems development and not interested in merely producing some form of technology demonstration. Furthermore, the experimentation if it is to be meaningful, cannot be performed using small systems that address minor issues for the organisation. The second inference that can be drawn from the above results is that system development should proceed with the knowledge that the system will need to be maintained during its production life. With respect to this factor, it is interesting to note the different maintenance strategies that have been adopted. For Organisations A and F, production systems are delivered in such a way that users can perform their own maintenance of the knowledge base. However, for Organisation F each system development ended with an agreement between the KBS group and the project sponsor on when the system would need to be re-engineered rather than merely maintained.

For Organisations B and E maintenance occurred as part of an on-going program of upgrade releases. In addition, the respondent for Organisation E stated that although upgraded versions of the system had been released, virtually no maintenance of the KBS component had occurred. A similar evolutionary approach was also adopted by Organisation C with the release of improved versions of the KBS. For the other two organisations, namely Organisations D and G, factors such as small user population and recency in deploying the production system, meant that maintenance issues had not been formally addressed.

In summary therefore most organisations in the case study were interested in deploying KBS's with long life expectancy, rather than dealing with short term problems. Furthermore, the respondent for Organisation F claimed that other systems had been deployed with longer life expectancies than the one included in this case study. With respect to maintenance strategies, there appeared to be more diversity that commonality across organisations.

Type of Technology Used in Commercial KBS's

What designs are employed in developing the major components of a commercial KBS?

For example, are most KBS's built with knowledge represent as production rules, and forward or backward chaining for the inferencing strategy? With respect to knowledge representation the following information was obtained:

Table 23. Knowledge Representation - Commercial KBS's

Organisation	Type of knowledge representation			
	Production rules	Frames	Production rules and frames	Other
Organisation A				
Organisation B				
Organisation C				
Organisation D				
Organisation E				
Organisation F				
Organisation G				

With regard to other types of knowledge representation, for Organisation B, KBS development followed an object oriented approach, in which an object known as a "FORM" was the main knowledge representation formalism. For Organisation E knowledge was represented as a series

Organisation	Type of inferencing strategy			
	Forward chaining only	Forward and backward chaining	Inheritance	Other (a)
Organisation A				
Organisation B				
Organisation C				
Organisation D				
Organisation E				
Organisation F				
Organisation G				

(a) For Organisation E, there was no inferencing strategy as all rules within the knowledge base were fired for each consultation.

From the data shown in the above table, the predominant choice of inferencing strategy were again the more "traditional" techniques of forward and backward chaining and inheritance. These inferencing strategies were used in six out of seven organisations, the exception being Organisation E. It is also considered that the KBS developed for Organisation E does not have an inference engine as all the CASE statements contained within the system are fired. Therefore, in System E1 there are no typical inference engine components such as an interpreter and scheduler which are executing and controlling the agenda, or a consistency enforcer which manages the overall consistency of the inferencing. The reason provided by the respondent for Organisation E was that the KBS had only a small number of "rules" and so firing all of them did not impact severely on overall system performance. In summary, therefore for the other six organisations, the inferencing

With respect to the inferencing strategy used by these commercial KBS's, the following information was obtained from the respondents in this case study:

Table 24. Inferencing Strategy - Commercial KBS's

Organisation	Type of inferencing strategy			
	Forward chaining only	Forward and backward chaining	Inheritance	Other (a)
Organisation A				
Organisation B				
Organisation C				
Organisation D				
Organisation E				
Organisation F				
Organisation G				

(a) For Organisation E, there was no inferencing strategy as all rules within the knowledge base were fired for each consultation.

From the data shown in the above table, the predominant choice of inferencing strategy were again the more "traditional" techniques of forward and backward chaining and inheritance. These inferencing strategies were used in six out of seven organisations, the exception being Organisation E. It is also considered that the KBS developed for Organisation E does not have an inference engine as all the CASE statements contained within the system are fired. Therefore, in System

E1 there are no typical inference engine components such as an interpreter and scheduler which are executing and controlling the agenda, or a consistency enforcer which manages the overall consistency of the inferencing. The reason provided by the respondent for Organisation E was that the KBS had only a small number of "rules" and so firing all of them did not impact severely on overall system performance. In summary, therefore for the other six organisations, the inferencing strategy for these commercial KBS's appeared to be tied closely to the knowledge representation formalism used, and so commercial KBS's do not seem to require new or exotic inference strategies. Second, choice of the inferencing strategy may also be due to the fact that many commercial KBS developments occurred using expert system shells.

Are knowledge based system shells the major delivery environment for commercial knowledge based systems?

Alternatively, are KBS's coded systems written in either artificial intelligence languages such as LISP or PROLOG, or a more conventional language such as C (or C++)? With respect to these questions, it was found necessary to separately identify the tools used in the development of the KBS as against the tools used to deliver a production version of the KBS, because as indicated by the table below, for a number of organisations these differed:

Table 25. Development & Production Environments

Development environment	Organisation						
	A	B	C	D	E	F	G
Development system (prototypes)							
AI language							
Expert system shell							
3GL or 4GL language							
Hybrid system:							
• AI language							
• Expert system shell							
• 3GL or 4GL language							
Production system							
AI language							
Expert system shell							
3GL or 4GL language							
Hybrid system:							
• AI language							
• Expert system shell							
• 3GL or 4GL language							

For the development environment, six out of seven organisations used expert system shell software either exclusively (Organisations C and F) or in combination with a programming language of some sort (Organisations A, D, E, and G). Again, as with the type of knowledge representation and inferencing strategy, Organisation B was the exception, due mainly to the fact that this development followed an object oriented approach. The development environment was classified as "AI Language", however, the language used was an in-house developed one rather than either LISP or PROLOG. In terms of the KBS development environment there are two conclusions that can be derived from the above:

(a) expert shell system software offers significant productivity advantages for this stage of development; and

(b) expert system shell software has sufficient power and functionality to effectively carry out this side of the development process.

With respect to the production environment, it seems that expert systems shell software does not provide a suitable delivery vehicle, although with a reduced emphasis. Ignoring Organisation B as a special case, only Organisation C used expert system shell software exclusively, while the two Organisations (A and E) delivered the final production version of the KBS in a different format (both using high level programming languages and 3GL based tools). Organisation F supplemented the development system with additional facilities built in a high level programming language, while Organisations D and G used a similar hybrid approach for both the development and the production system. It should also be noted that for Organisation C the respondent stated that great difficulty in delivering the production version of the system occurred. The expert system shell software used in the development environment, was found to have severe efficiency problems in the production environment. Accordingly, the production version of the system was re-written using a different expert system shell. Therefore, it would appear that significant performance problems arise in the transition from development to production, in terms of the number of users accessing the system, and the hardware environment of the KBS's production version. Both of these issues are considered in the analysis for the next question presented below.

How are the final production versions of these KBS's used?

For example, are they stand alone systems that run on microcomputers with a simple user-oriented interactive interface? Alternatively, are they highly integrated systems operating on a range of hardware platforms? The type of production KBS finally delivered to the project sponsor for the organisations that participated in this case study research project was:

Table 26. Production Versions: Classification of Commercial KBS

Type of KBS	Organisation						
	A	B	C	D	E	F	G
Consultation type system:							
Stand alone							
Integrated system							
Embedded system							
Other							

For five out of seven systems, the production KBS was integrated to some degree with other systems within the organisation. As with other aspects, the production version of the KBS for Organisation B was unusual in the sense that users of the system were not employees of the organisation but independent insurance agents. Furthermore, the system was deployed on lap top PC's and so by design would have little likelihood of being integrated with the organisation's other data processing systems. With respect to the system used in Organisation C, the KBS replaced a pure paper based procedure and so had little requirement to integrate with other systems. However, overall one could conclude that viable commercial KBS's are more likely to require some level of integration with other conventional systems.

With respect the hardware platform that the production KBS resides on, the following information was collected:

Table 27. Hardware Environment - Commercial KBS's

Hardware environment	Organisation						
	A	B	C	D	E	F	G
Development system (prototypes)							
Centralised mainframe							
Distributed mainframe							
Mini-computer							
PC							
AI workstation							
Production system							
Centralised mainframe							
Distributed mainframe							
Mini-computer							
PC							
AI workstation							

If the above data is any guide, then it would be that future KBS developments for commercial organisations will deliver production systems that run on microcomputers (five out of seven organisations). Although the KBS development for Organisation D uses both mini-computers and a centralised mainframe, this system operates predominantly with a front end located on the PC, and the other computers mainly used in back end data retrieval and transaction record processing. Accordingly, even for this organisation the KBS component resides on a microcomputer. The system for Organisation C is the only true mainframe application, and the system for Organisation A is located on an AI workstation for efficiency reasons.

What has been the goal in using KBS technology?

For example, are commercial KBS's seen by their developers and/or users to be experts, colleagues, or merely assistants? For these questions the respondents were asked for their perception of the level of expertise found within the KBS, namely:

Table 28. Level of Expertise - Commercial KBS

Organisation	Level of expertise		
	Assistant	Colleague	Expert
Organisation A			
Organisation B			
Organisation C			
Organisation D			
Organisation E			
Organisation F			
Organisation G			

From the above table, five out of seven KBS's (Systems A1, B1, D1, F1, and G1) were considered to provide less than expert level service to their ultimate users. It is interesting to note some of the comments made by respondents on this point. For instance, the respondent for Organisation D claimed [interview transcript, page 3]:

> *"Well, that's an interesting thing. See, I see a trend, right. I mean, AI ... and Expert Systems as they were promulgated in 1985/1988 time frame - were very much geared around the acquisition of knowledge from humans. Right? And the big story was how were we going to get the knowledge out of the humans. And I perceive that is by and large gone. Because that is too hard, and it produces systems that are brittle. You can't really produce human resiliency in the system. They are very hard to maintain, and they are only useful in very, very specific circumstances. There are very specific decision support circumstances where that technology is useful. But KBS technology, as distinct from Expert Systems, the technology is being used to take knowledge out of paper documents. And the term that we use in this organisation is 'Administrative Support Systems'."*

These comments are supported to some extent by the respondent for Organisation A who claimed [interview transcript, page 9]:

> *"Let me tell you initially our first system, which was done in a traditional expert system way. I think the expert was just about to retire and had worked there for 30 to 40 years, and we did that we built a knowledge base that he was quite happy with. When we were ready to go into that sort of pre-implementation phase, it was*

about the time when TQM started to raise its head in [the organisation] and the thrust from all of that was the creative involvement of all people in generating a standard procedure. And this sort of went against the concept that you had an expert who input [their expertise] into the knowledge base and that was gospel. And I think it still probably could have worked even through using that expert as the leading figure, which is sort of what we do anyway. But the problem is that the guy retired, and when he retired he wasn't there to defend the thing he originated. And of course little things emerged, and then all of a sudden the whole knowledge base wasn't relevant any more. So you really need to have that leading person but that person really shouldn't be someone about to retire. That person should champion the whole thing, and champion the best practices of it more so than the knowledge base."

Even for the organisations with KBS's that are considered to operate at "expert level", for Organisation C the knowledge acquisition phase did not try to identify the leading expert in claims assessing for the organisation, but used a very experienced person. In summary, the development effort required to produce systems at expert level performance do not seem to be necessary, will often lead to inflexible systems that users will not accept or support.

Development Processes for a Commercial KBS

What area within the organisation undertakes the KBS development?

Is KBS development undertaken by the MIS or IT area, other areas of the organisation, or by persons outside of the organisation? The answer to this question is similar to those concerned with the area that championed the introduction of KBS technology into the organisation, that is development of the KBS occurred outside the MIS or IT area. The reasons for this varied. For example, the project manager for the System C1 development wanted to avoid as much contact or involvement with the MIS department as possible. This was due to the project manager's and a more general perception within the organisation that this area was fairly ineffective [interview transcript, page 9]:

> *"Respondent: Oh, yes - oh no, nothing coming through ... no MIS. I don't want to do projects with MIS. Because they end up ... one of the biggest problems that the corporate guys have is that these MIS guys chew up a lot of money, and you never ever get running code out of the other end. You very rarely get running code out the other end. Like insurance code. Like they have had three goes at one project that I know of - used a lot of money - and they still have not got any running code. This is their third attempt now, and it looks like it is going to fail or flop. So ... Oh, this things happen. This is not just here; this happens everywhere!"*

For the KBS development within Organisation A an unusual approach was adopted. The initial development phases (from conceptual through to field prototype) were handled by the research and development area, but if the system underwent further development to realise a production system, this part of the development was undertaken by the MIS or IT area. The inference emerging from these organisations is that the technology is still to be accepted as part of the development portfolio for a commercial MIS or IT area.

What is the attitude of various areas with an organisation to KBS development?

Is the attitude supportive, non-committal, obstructive? The attitude of areas other than the one developing the KBS are indicated in the table below:

Table 29. Attitude Within Organisation to KBS Development

Area of organisation	Organisation							
	A	B	C	D	E	F	G	AVG
Perception of KBS group's effectiveness								
MIS/IT department	5	4	4	5	6	4	1	4.14
Corporate (Head Office)	6	6	6	4	5	6	2	5.00
User areas	5	3	7	5	4	6	3	4.71
Satisfaction with KBS developments								
MIS/IT department	5	2	4	5	6	4	1	3.86
Corporate (Head Office)	4	5	6	5	6	6	2	4.86
User areas	4	4	7	5	4	6	3	4.71
AVERAGE SCORE	4.83	4.00	5.67	4.83	5.17	5.33	2.00	4.55

Before any analysis of the above data is considered, it should be stated that the number of data points (seven organisations surveyed) prevents any meaningful statistical analysis of these Likert scores or any derived averages to be performed. Notwithstanding this comment, it would appear that the attitude of other areas to KBS developments is reasonably consistent across all organisations, (average Likert score of either 4 - a neutral interest or 5 - a moderately high level of interest) with the exception of Organisation G. Furthermore, when considering the averages across different departments for these organisations it is interesting to note that the MIS or IT area rates the lowest Likert score, and the corporate or head office area rates the highest (both for perceived effectiveness and satisfaction with KBS developments to date). It would appear from these data that commercial organisations are not very excited about KBS technology which seems to indicate that these organisations remain to be convinced that the technology can deliver tangible benefits. This comment applies particularly to the MIS or IT areas. On the other hand, the technology's continuing use within commercial organisations will be assisted by the interest of very senior management in the technology.

Respondents were also asked to compare conventional and KBS development, and the following Likert scores were obtained:

Table 30. Comparison of Conventional IS and KBS Developments

Particulars	Organisation							
	A	B	C	D	E	F	G	AVG
KBS developments								
Importance to organisation	6	3	6	2	3	5	1	3.71
Level of user involvement	6	1	1	3	6	7	5	4.14
Interest of senior management:								
Within MIS	4	1	2	4	7	1	2	3.00
Outside MIS	5	1	5	4	4	5	3	3.86
Effectiveness of completed systems	4	5	7	5	6	5	4	5.14
Integration with other systems	6	1	2	5	6	2	3	3.57
Conventional developments								
Importance to organisation	7	7	6	5	7	5	7	6.29
Level of user involvement	5	7	6	4	7	2	2	4.71
Interest of senior management:								
Within MIS	7	7	7	7	7	7	7	7.00
Outside MIS	5	7	6	N/A	6	3	6	4.71
Effectiveness of completed systems	7	5	3	N/A	5	2	3	3.57
Integration with other systems	7	5	2	N/A	7	3	2	3.71

The data shown in the above table supports the inference made earlier in this section, that is KBS developments are less important to the organisation as a whole than conventional developments. This can be seen in the above data when comparisons of the average Likert scores (across all organisations) are made between conventional system developments and KBS developments. In all cases, with the exception of the perceived effectiveness of the delivered production system and the level of integration, conventional systems rated higher than knowledge based systems. It would appear from these data and the data shown in the previous table, that KBS technology still has a long way to go before commercial organisations treat it no differently to more conventional data processing.

What are the typical milestones of a KBS development?

Are these milestones similar across different KBS developments by different organisations? The main reason for asking this question was to provide an insight into whether or not their are significant differences between the activities involved in a KBS development and those involved in a conventional systems development. The significant events for the production systems of those organisations involved in the case study are present in the table below:

Table 31. Key Events in the KBS Development

Organisation	Description of milestone, major event
Organisation A	1. Establishment of customer funded R&D program. 2. Delivery of successful demonstration prototype. 3. Delivery of successful field prototype. 4. Customer approval to implement into production (*key milestone*). 5. Project plan for implementation. 6. Delivery of final system. 7. Installation of maintenance procedures.
Organisation B	1. Development started in 1989, and there were initially lots of problems. 2. End of 1991 development of a field prototype used by a small number of insurance agents. 3. End of May 1992 first production version (Ver. #2) of the system released with added validation functions. 4. End of February 1993 second production version (Ver. #3) released with added production rules covering underwriting.
Organisation C	1. Production system for 60% of claims. 2. Enhanced system for greater (almost 100%) percentage of claims. 3. Conversion from development expert system shell environment to production expert system shell environment. 4. Minor changes to enhance effectiveness of the system.

Organisation	Description of milestone, major event
Organisation D	1. Initial feasibility study - technology review - match problem. 2. Concept prototype - proof of concept 3. Project definition - proposal for project. 4. Software engineering by spiral model.
Organisation E	1. Getting approval to proceed. 2. Completing system development. 3. Completing acceptance testing. 4. Release. 5. Post-implementation review.
Organisation F	1. Advisability study. 2. Project definition report. 3. Proof of concept prototype. 4. Approval to proceed. 5. Peer review - both user and technical. 6. First to "N"th production test system. 7. Production system. 8. Next cycle time frame plan.
Organisation G	1. Identification as a suitable candidate for KBS. 2. First working system - for review by outsiders. 3. Operationally useable system. 4. Validation in service - decision making performance. 5. User acceptable system.

In terms of common events or milestones across different organisations, the above table indicates that in five out of seven organisations (Organisations A, D, E, F, and G) some form of phased development occurred, beginning with a feasibility study to determine if KBS development should proceed, and an apparent emphasis on a prototyping approach to actual system development. For Organisations B and C a more evolutionary approach appears to have been taken, but again the use of prototyping in the system development phases is also evident for these organisations. Furthermore, when the types of activities and milestones are considered as a whole, there would appear to be very little difference, at this level, between conventional systems development and KBS development.

What sort of documentation is generated from a KBS development?

Are there any similarities in the documentation across different KBS developments by different organisations? A comparable comment can be made concerning the type of documentation produced during the KBS development, that is there appears to be little difference at this level between conventional IS developments and KBS developments, as can be seen from the information shown next:

Table 32. Documentation Generated During the KBS Development

Organisation	Type of documentation
Identification/selection of suitable area.	
Organisation A	Research proposal
Organisation B	Memoranda, management reports.
Organisation C	Verbal discussions between client sponsor and consultant. Very informal.
Organisation D	Feasibility study
Organisation E	Report identifying options - 42 potential projects

Organisation	Type of documentation
Organisation F	Advisability study; project definition report; high level DFD's
Organisation G	Notes
Knowledge acquisition	
Organisation A	Tables; user procedures documentation; decision tables. Also tool driven using TABLEAUX
Organisation B	Interview notes.
Organisation C	Interactive graphic based process using pictures drawn on an electronic whiteboard. Evolutionary model development in which model was developed and then processes were put in place to try and "break" the model.
Organisation D	Design notebook
Organisation E	Informal notes; documentation of rules in English.
Organisation F	Repertory grids; screen layouts; decision trees/tables; cognitive maps; E/R models
Organisation G	Notes, transcripts
Knowledge representation	
Organisation A	Mostly rules and/or frames
Organisation B	None - coded directly into the knowledge base.
Organisation C	The same documentation as described in knowledge acquisition.
Organisation D	Design notebook
Organisation E	Coded directly from previous documentation into the system.

Organisation	Type of documentation
Organisation F	Decision trees/tables; cognitive maps; object maps; frame hierarchy diagrams; E/R model
Organisation G	Rules
System design	
Organisation A	Waterfall type documentation; user requirements specification (text version of prototype; functional specification; data definitions and data dictionary; technical design; system test plan.
Organisation B	Design documents.
Organisation C	Direct coding of models and other knowledge into ADS.
Organisation D	Design issues; project issues; knowledge base representation issues; user environment issues; problem resolution; quality management.
Organisation E	No documentation for this activity.
Organisation F	Functional specification (to client, steering committee, Director of KBS); human activity models; high level DFD's; lexical model; object hierarchy; object specifications; high level system interface specification
Organisation G	Module diagrams.
Verification and validation	
Organisation A	System aspect - test; KB - monthly performance report.
Organisation B	Not until recently when system reached production versions (greater than Version #2).
Organisation C	Statistical evaluation - comparison of KBS C1 with human judgement in the settlement of the same insurance claim. Lots of claims.

Organisation	Type of documentation
Organisation D	N/A
Organisation E	Report outlining test results (rigorous statistical testing of system against human judgements); post implementation review of the decision making performance of the system.
Organisation F	Test deck - laboratory; and business area testing; peer review, evaluation reports;
Organisation G	Test cases - formal and regression.
Implementation	
Organisation A	Performance reports; Training sessions (Research Labs, and IS); TABLEAUX training.
Organisation B	Within last year - delivery document. Recently formal (classroom) training.
Organisation C	N/A
Organisation D	N/A
Organisation E	Report describing the system's functions; technical manual; post-implementation review of useability; training manual.
Organisation F	System sign offs
Organisation G	User guide; system development documentation.

Two conclusions can be reached from the data presented above. First, most of the documentation listed would be just as useful in a conventional systems development project. For example, Organisation F used classic structured development techniques or tools such as E/R models, and data flow diagrams. During the system design activity, Organisation A used classic "Waterfall type documentation". Second, there appears to be little similarity in the type of documentation used across organisations, and that choice of documentation depends to a large extent on the preferences of the project manager. For instance, the comments of the respondent for Organisation F on this point are considered appropriate [interview transcript, page 7]:

> *"Respondent: Our business is integrating products, existing products together. So we might be integrating the whole Microsoft Office™ suite ... right? ... plus hypertext, plus an expert system shell, plus an SQL back end to a database ... right? So you know, you get the whole lot, all rolled into one big ball of wax. So all the rest of it is stock standard software engineering. ... Ah, you know, we don't build and install ... we do it in stages. We have deliverables, we have milestones. We have all the proper ...*
> *Ian Caddy: So, if we say that "4" follows a traditional software engineering life cycle ...?*
> *Respondent: Yeah, it follows the spiral model. We are firm believers in the spiral model here.*
> *Ian Caddy: Assess the risk at each stage of the project's life.*
> *Respondent: Exactly. And I mean ... if you take the spiral model, that is the underlying philosophy of our approach."*

Similar comments, that the KBS development is very much a variation of standard software engineering, would also apply to the system developed for Organisation E in which the KBS was a component (amounting to about 5 per cent.) of a much larger and in style more conventional system.

What type of personnel are used to develop KBS?

For example, do most KBS development personnel have a conventional IS background? The size of the KBS development teams varied across the different organisations with respect to the different systems being implemented. As shown in the table below (which looks at the total personnel involved in the development), all projects had teams of a reasonable size, comparable to many conventional information system project teams. The teams' composition, with the exception of specialists such as knowledge engineers, is considered similar to many equivalent positions that would make up a conventional IS development team.

Accordingly for KBS developments, although requiring some specialists, many activities and tasks require the same sort of skills required for a conventional development.

Table 33. Personnel Structure of the KBS Development Team

Particulars	Organisation						
	A	B	C	D	E	F	G
Expert(s)	2	1	1	0.5	8	4	1
Users	4	6			3	7	
Knowledge engineers	1	1	4	1.5	3	2	5
Programmers	2	4		3		1	
Systems analysts		1				1	
Specialists	1					1	
Business analysts	1			1			
Other (Manager)		1					
TOTAL	12	13	5	6	14	15	6

Given that the development activities for KBS projects have a certain degree of likeness to conventional IS developments, it is not surprising that the sort of qualifications and background experience found in commercial KBS development teams, is as follows:

Table 34. Background Experience - KBS Development Teams

Organisation	KBS Project Manager		KBS Developers	
	Years in IS	Years in KBS	Average years in IS	Average years in KBS
Organisation A	1.00	3.00	6.33	2.10
Organisation B	0.00	2.00	0.00	2.00
Organisation C	18.00	8.00	6.00	4.67
Organisation D	5.00	1.50	5.00	0.00
Organisation E	4.00	1.50	3.00	0.00
Organisation F	5.00	3.00	3.50	2.25
Organisation G	10.00	5.00	3.00	0.50
AVERAGE:	6.14	3.43	3.83	1.65

As this table indicates, six out of seven organisations involved in this case study, the personnel involved in the development had considerable conventional IS experience, both for the project manager as well as project team members. This is interesting given the often stated conclusion that conventional IS personnel do not make good KBS development personnel. It is interesting also to compare the above data with two views from project managers for Organisations F and G. In both cases the respondents have verbal supported the "traditional view", that is that good IS personnel do not make good KBS development personnel. For instance, the respondent for Organisation G stated the following [interview transcript, page 3]:

> *"Respondent: No, no. They were interested, their perception was that KBS was a new way of programming. Very clear. "This is a programming language, you write programs in it don't you? No problems. We understand that!" Do you? "Well, we will learn the language and we will be able to write programs." And they went off and wrote COBOL programs using KBS tools. Absolutely, I watched it happen. I mean on a related subject, I don't know whether we come on to that - I made a serious attempt to train some systems analysts as knowledge engineers, and failed miserably. The only people who I had success with were those untainted by conventional training.*
>
> *Ian Caddy: That is something that has come through from a lot of the other contacts. In two [other] cases the contacts identified people who were english teachers with an arts degree as being their best knowledge engineers.*
>
> *Respondent: Yep. Personally I don't think it is as much to do with their personal qualifications as their experience. The guy that I failed most miserably with, and it was a total failure, had twelve years experience in the role of a systems analyst. He knew what to do, and I just could not teach the old dog new tricks. I did not succeed. And I had that experience before ... before [the organisation] with exactly the same result. So, my conclusion was that it is not even worth trying."*

Similarly, the respondent for Organisation F stated [interview transcript, page 29]:

"Respondent: ... Now, [team member], he had three years of experience there, and all three years in KBS - we picked him up straight out of uni[versity]. In fact my best staff were my graduates, because they weren't coming to you with any baggage - particularly methodological baggage. All my IT people were nuisances! Particularly for the first eighteen months - they were really all a waste of space. The best knowledge engineer I had was an ex-teacher. ...
Respondent: ... Business/IS background, IS background, IS background, Comp[uter] Science. Most of these were all postgraduate IS qualifications, and postgraduate diplomas - those sorts of things. They all had base degrees in something else, usually from business or humanities. I should explain we selected staff, we initially wanted to recruit from outside, the recruitment fees came down, and we selected staff and then trained them internally. Which is not the way that I wanted to go, but we sort of had our hands tied. But ... but ... lengthened our ability to deploy systems by about nine months. Really peeved me."

However, it should be noted that ultimately this project manager developed systems using personnel with reasonable conventional IS experience! The inference appearing here is that individual personnel may have difficulty adjusting, and that the length of experience within a conventional development environment makes the adjustment more difficult. From this point of view the respondent for Organisation F would agree. It was not impossible to retrain staff, but it was definitely easier with new graduates, that "weren't coming to you with any baggage - particularly methodological baggage" than it was for IS staff with experience in conventional developments. It was mainly a matter of adequate training for conventional IS personnel, rather than something inherently wrong with these people. This meant, for this Organisation, development was delayed but not curtailed. Another conclusion that flows from the above data is that it appears to be mandatory for the project manager to have significant KBS experience for the development to succeed.

Are KBS development projects more likely to be one-off activities or part of a consolidated and continuing development program?

It would appear that economic conditions have a strong influence in the level of activity by a commercial organisation in this area as is indicated in the data shown below obtained from the interview transcripts:

Table 35. Activity Level of KBS Developments

Organisation	KBS Development Projects		Likelihood of more developments
	One development	Many developments	
Organisation A			YES
Organisation B			NO
Organisation C			YES
Organisation D			UNLIKELY
Organisation E			NO
Organisation F			UNLIKELY
Organisation G			NO

Using the above data, only two organisations out of seven appeared to have an on-going commitment to the technology, namely Organisations A, and C. In the case of Organisation F, although around forty (40) KBS's were deployed, the KBS development cell was closed down soon after the respondent left that organisation. With respect to Organisation C, there has been limited interest in the technology, with possibly two additional projects under investigation currently to assess feasibility. Accordingly, Organisation A represents the only organisation where the technology will experience continued growth. This appears to be due to two factors: (a) the maturity of the technology within the organisation and the number of successful KBS developments; and (b) the level of commitment by a major user area, which sees potential application of KBS technology to hundreds if not thousands of decision support problems. Therefore it appears, as has been indicated with other analyses that most commercial organisations are still going through an introductory experimental phase with respect to this technology. Additionally, it may require the economy to move out of the current recession to re-awaken interest in the technology and move it closer towards a higher acceptance by MIS, user, corporate or head office areas.

Are KBS development projects large projects in terms of size of development team, length of development schedule, and cost of development?

It is interesting to compare this experimental or introductory attitude that appears to be the case for most of the commercial organisations, with the type of developments that were undertaken. Given this environment within the organisation it was expected that the type of system development would be small, and associated with minor activities of the organisation rather than mainstream. However, it was found that organisations involved themselves in reasonable development projects with significant impact on an areas day to day activities. The length of KBS developments is shown in the table below:

Table 36. Elapsed Time - Commercial KBS Developments

Organisation	Date started	Date completed	Elapsed time (months)
Organisation A	06/90	08/92	26
Organisation B	08/89	02/93	46
Organisation C	04/88	06/89	14
Organisation D	01/92	12/92	12
Organisation E	01/87	03/88	14
Organisation F	08/91	02/92	6
Organisation G	12/87	03/91	40

As the above data shows, the developments occurred over reasonably long periods of time. It should be noted that for Organisation E the development period is for the first production version of the system - it was not released for general use by bank branches until 1993, approximately size years after development of the system commenced. In the case of Organisation B, the data shown relates to the entire development process including the on-going release of later versions of the production system. The development duration to release of the first production system was thirty seven (37) months which is more an indication of the problems experienced in system development, particularly with in-house development of the inference engine.

Possibly a better indicator of size for commercial production KBS's are the number of users. As indicated in the table below, significant user populations exist for five out of seven organisations, namely Organisations B, C, D, E, and F. The system installed in Organisation A has functional problem domain limitations on its user population, and so was not designed to ever have a large number of users. In only two organisations out of seven (Organisations B and G) has the expected number of users been less than the actual, and even so, for Organisation B the number of actual users is quite significant. It is interesting to note that as most KBS's are deployed and operated on PC's that software performance was only a major issue for Organisation C, which operated its system within a multi-user mainframe environment.

Table 37. User Populations for Production Versions of the KBS

Type of user	Organisation						
	A	B	C	D	E	F	G
At start of project							
Direct users of KBS	12	> 2,000	20	> 1,000	> 1,000	120	50
Indirect users of KBS	4	0	0	0	0	< 10	0
Currently							
Direct users of KBS	12	200	100	> 1,000	> 1,000	> 1,000	3
Indirect users of KBS	4	10	0	20	0	0	0

Given the length of the developments and the sizes of the development teams, these commercial KBS developments should also be considered significant in terms of cost when looked at on a system by system basis, as can be seen from the data presented in the table below:

Table 38. Expenditures on Commercial KBS Developments

Particulars	Organisation						
	A	B	C	D	E	F	G
1988							
Est. exp. ($'000)	N/A	N/A	300	N/A	N/A	N/A	1,000
Per cent - IS exp.	N/A	N/A	< 5%	N/A	N/A	N/A	< 1%
1989							
Est. exp. ($'000)	N/A	1,500	800	N/A	N/A	500	1,000
Per cent - IS exp.	N/A	< 2%	< 10%	N/A	N/A	< 5%	< 1%
1990							
Est. exp. ($'000)	2,800	1,500	800	N/A	N/A	500	1,000
Per cent - IS exp.	5 - 10	< 2%	< 10%	N/A	N/A	< 5%	< 1%
1991							
Est. exp. ($'000)	3,000	1,500	800	N/A	N/A	750	400
Per cent - IS exp.	5 - 10	< 2%	< 10%	N/A	N/A	< 5%	< 1%
1992							
Est. exp. ($'000)	3,500	1,500	800	N/A	N/A	750	N/A
Per cent - IS exp.	5 - 10	< 2%	< 10%	N/A	N/A	< 5%	N/A
1993							
Est. exp. ($'000)	3,500	1,500	800	N/A	N/A	N/A	N/A
Per cent - IS exp.	5 - 10	< 2%	< 10%	N/A	N/A	N/A	N/A

Financial data could not be obtained for Organisation D for confidentiality reasons, and data was not separately available for Organisation E because the KBS development could not be separately identified, but an estimate of between $50,000 to $100,000 per year expenditure appeared reasonable on the levels of staffing within the project team. A final comment in terms of size, is that although the individual projects were significant, there was never a significant number of KBS projects under way in any organisation (with the possible exception of Organisation A) and accordingly the percentage of expenditure on KBS development when compared to total expenditure on IS development was small across all organisations, in which it never exceeded more than 10 per cent. of total IS expenditure on development. These data further point to the fact that KBS technology is still going through an introductory stage with commercial organisations.

What development approaches have been used in the construction of commercial KBS's?

Are KBS's developed using an evolutionary approach or is there evidence of a more structured development environment? From the comments extracted from the interview transcripts, it is considered that the predominant development approach adopted by commercial organisations was prototyping (six out of seven organisations). However, the way prototyping was applied across these organisations differed. For instance, this technique occurred within a Boehm type spiral development approach for Organisations A and D, was part of a broader, an ad hoc evolutionary development process for Organisations B and E, or through some other iterative development process - Method 1 for Organisation F, and an in-house development procedure for Organisation G.

The exception was Organisation C which adopted an evolutionary approach in which two major releases of the KBS were made. The initial release was a full production system that covered 60% off claims made on the insurance company for bodily injury (which took seven months to build) and then a second release in which all claims were handled (which took a further 11 months). Development was very much evolutionary with little identification of phases of development and very few checkpoints, milestones or review points [interview transcript, page 9]:

> *"Ian Caddy: And are there any other things, sort of milestones, that have happened since the conversion of the system to ADS ... say enhanced its ... you know it's explanation [facility], or better user interfaces, or ...*
>
> *Respondent: Oh, we have converted ... we have done lots and lots and lots of those changes. And we strive to do lots more of them. None of them you could say are really big things in themselves. And none of them we would run as a project. But we have done lots of them, to make the system more usable, to make it friendlier, make it easy to use, all of those things - user interface stuff. ... The original system and today's system you wouldn't recognise as the same system."*

It is interesting to note for this development the power of the project manager in the style of development. The organisation had no expertise in these types of developments and so contracted with a software development house (which at that stage also had little experience in KBS development). The project manager's quite forceful personality and the sponsor's willingness to leave system development team alone, meant that the development proceeded based on that project manager's philosophy. For example, this project manager's attitude to prototyping is summed up in the following remarks [interview transcript, page 27]:

"Respondent: Yeah, yeah - we learn. But it is not a formalised thing were we examine what we learn and stuff. Yeah, it is just strange ... we just get better at it through experience basically. I mean I have got a job here to do a prototype for instance. Here's a job - do a prototype, right? Now, we were basically cornered into doing it. We had someone else who introduced us in there, and told that the way to go ahead was to do a prototype, and not to do the full system. He was already in there, and when we got there that was already the requirement - to do a prototype. What a prototype normally means is ... a prototype is just another name for something that doesn't work. For most people ... a prototype means 'I want to see something on my computer ...' "

For the other organisations that adopted a prototyping approach, there were differences. For instance, Organisation A adopted the throw away version of the technique, progressing the development through at least three phases - conceptual, experimental and then field prototype - at which point development may cease as the client is happy with the field prototype and unwilling to fund further expenditure to re-engineer the prototype into a production system. For Organisations B and E, similar to Organisation C, and as Hickman et al. (1989) claim, it was difficult to differentiate the prototyping from merely an evolutionary development process. However, unlike Organisation C, although both organisations did mention the use of prototypes in the development process. For Organisations F and G the use of prototyping was more formal than for Organisations B and E, where there was the development of a conceptual prototype, but these organisations then did not have the other stages that Organisation A went through, reverting at this point to an iterative form of development.

Finally, it is interesting to note the comments from the respondent for Organisation D, where the KBS development was undertaken by an outside consulting firm. In this case the style of prototyping was not only part of the development process but also linked to the marketing procedure of the consulting organisation [interview transcript, page 6]:

> *"Respondent: It really depends on what you are doing. I mean some people ... depends how big it is for a start ... how much you have to invest in the prototype ... right? ... to get the business. I mean this is a money making organisation. It makes it's living out of delivering KBS or delivering new technology ... right? ... be it multimedia, or whatever else you are dealing with. So it is quite prepared to spend quite a bit of money on a good prototype to win the business provided there is a significant return on it. ...*
> *Well, we ... the methodology that we have is somewhat different to that. We do prototyping. We try and choose the right ... an acceptable tool at the beginning. We have a range of them here. We generally pull one in that we already have, that fits the bag, that fits the parameters reasonably well. And we [then] invest effort to build a significant prototype. In other words they are not really throw away prototypes. And we put a fair amount of effort into the user interface."*

In summary, although prototyping appears to be the predominant development approach for commercial KBS's, there are contingent factors particular to each organisation which means that this technique has not been used in a similar fashion across all of these developments.

Are KBS development approaches similar to those used in conventional information systems developments?

The answer to this question has to be approached from a number of different directions, in which the similarity to conventional systems development varies. First, for Organisations B, and C there was little similarity. For Organisation B, the development created its own inference engine and knowledge representation language, something that would not occur in conventional IS developments. In addition, the development was the first PC based development for this organisation and the first object oriented development. For Organisation C, again this development was the first successful KBS development (another attempt at using KBS technology by the MIS area failed). The core of the development process was to build a theory or model that simulates but does not emulate the judgmental reasoning process of a claims assessor. Development of this model occurred using an electronic whiteboard and pictures. Furthermore, the system itself also has changed the human intellectual reasoning process with respect to claims assessing. Accordingly, from both of these aspects there is a marked difference between conventional IS developments and these KBS developments.

For Organisation A, there were differences as well as similarities. During the initial development phases when knowledge acquisition the development of the three prototype occurred there was little in common with conventional IS development. However, where development projects progressed to build production KBS's (following to some extent the Boehm development approach), this latter phase had a lot of similarity, using a waterfall development process [interview transcript, page 14]. For Organisation E the system was eventually written in a microcomputer database language, with the knowledge represented in CASE statements in that language. Accordingly, in this case the development environment meant that many of the development activities would have been equivalent to a conventional microcomputer development.

The respondents for Organisations D and F distinguished between the product that was being developed and the development process that created the product. The product was seen as been something quite different to conventional IS development, in the sense that in both cases the system was built through the use of an expert system shell. But the development process was seen to be similar to conventional developments. For example, the respondent from Organisation F made the following remarks [interview transcript, pages 49 and 52]:

> *"Respondent: Yes. Right, philosophically speaking you are still have a problem, you are still solving a problem, you are still deploying a system, whether it be a conventional system or not. Still impacting on job design, still impacting the people in the world, still impacting the keepers of the infrastructure - the architecture, the network and load, still impacting on disk store - and you eventually deploy a computer system at some stage. Now the production and implementation issues are very similar, the problem definition issues are very similar. The method of specifying your alternatives are very similar. The deliverables in each phase, the management that you apply is very similar. Now technically, the actual systems development process is different, and it is different not because it is knowledge based systems, it is different because it is difficult to specify something that you can't see, feel, touch, or appreciate. Whereas ... if you ... maybe in*

> *twenty years time, people ... everybody will know what a knowledge base is, and specification will be easy - I don't know. But at the moment, it is still a fuzzy concept. ...*
> *Respondent: I think you can build any system in a variety of ways. I don't think that choosing Method 1 was so ... was the crux of our success ... I think was the people. Their willingness to work in a different world. I suspect that if you had thrown any methodology at that group, they would probably have found a way to make it work. I don't think any proprietary product is better than another. I think you could even bend the system development life cycle approach to build such a system. Take you longer but you could do it. And by taking you longer, you may not deliver a system that users will necessarily accept."*

For Organisation D, a lot of system development was referred to as standard software engineering, and this organisation also followed a standard development process that not only covers KBS developments but also more conventional developments. Finally, the respondent for Organisation G stated that the only real difference for KBS developments was the knowledge acquisition phase, and that excluding this, particularly in terms of the development process, there was very little difference.

In summary it would appear that the development process from a project management perspective is very similar for KBS and conventional IS developments. However, at the technical level, contingencies of particular developments may mean there is a great deal of difference in the development of the product. Further consideration of this point is dealt with in the analysis for the following questions.

Status of Commercial KBS Development Methodologies

Do commercial organisations possess KBS development methodologies?

The point to begin the determining whether or not any of these commercial organisations possessed and used a KBS development methodology is the perception of the project managers as to whether or not they believe that the system development followed a methodological approach. Their responses are shown in the table below:

Table 39. Methodologies - Perception of KBS Project Leaders

Organisation	YES	NO	MAYBE
Organisation A			
Organisation B			
Organisation C			
Organisation D			
Organisation E			
Organisation F			
Organisation G			

Five out of seven project managers thought they did, while one was undecided and one definitely thought not. Obviously, although perceptions are an important factor, perception is not a definitive answer. For example, as was considered in the literature review in Chapter 2 of this project report, it is important to distinguish between using a development methodology and being methodical in a system development. The classic example of this situation for this case study would be Organisation C. The respondent for this organisation stated that development did not follow any development methodology as the respondent did not believe in the usefulness of system development methodologies. In fact, the respondent claimed [interview transcript, page 22]:

> *"Respondent: No, but we don't ... I guess I have a philosophical aversion to them, rather than a particular aversion to any one of them. I just don't see how someone can come up [with] ... and I have been in this game a long time now, especially in conventional stuff - where I have seen methodologies by the bucket load. And I have seen organisations pay money - and get the complete set of, you know two yards worth of manuals to hand out to everybody, and they follow them rigorously, and guess what happens at the end?*
> *Ian Caddy: Nothing.*
> *Respondent: Yes. Bloody nothing! And they never get a system. And they still do it!"*

Accordingly, there was no formal management structure, no formal development process, no reviews, no specification of particular development techniques, with the exception of what the respondent called use of the scientific method. That is, the development focussed on the creation of a model of the problem, and then an assessment of how robust this model was, by attempting to "break the model". That is, development from a certain point followed the falsificationist approach or the "scientific method". In terms of the methodology assessment framework developed in Chapter 2 of this project report, there appears to be an underlying philosophy, and the development approach whatever it may be classified as, produced an end result in the sense that a successful production KBS was delivered. However, when all the other aspects of the assessment framework are considered, it is found that this approach does not possess anything in these areas. Accordingly, the conclusion that appears to emerge for this organisation followed a methodical development approach but did not use a development methodology.

The reasons given as to why respondents considered a methodological approach was appropriate or not are as follows:

Table 40. Reasons for Adopting or Not Adopting a Methodology

Organisation	Reasons
Organisation A	Assure us that our development and management practices can deliver quality products/applications to our customers.
Organisation B	Add predicability in terms of resources and time, once a direction (platform, scope) has been defined. Hasten the productivity of new team members.
Organisation C	Firm belief that effective KBS systems (in fact any sort of system) can be built without the assistance or hindrance of a methodology.
Organisation D	Reduces risk. Used as a management tool.

Organisation	Reasons
Organisation E	Depends on the size of the system. As (this KBS) was a reasonably small development, with a small development team - the effort in following a methodology compared to the benefits of following a methodology did not justify it.
Organisation F	Managerial control. Standardisation of re-usable methods and techniques. Common set of approaches, techniques, tools that all staff can use and understand.
Organisation G	A process is needed for management and direction which is needed for both technical staff and managers.

In terms of other elements that are indicative of whether or not the organisation possessed a KBS development methodology are as follows. For instance, respondents were asked to provide the following information on the development approach used:

Table 41. Information on Application of the KBS "Methodology"

Particulars	Organisation						
	A	B	C	D	E	F	G
No. of times methodology used	6	0	0	> 10	0	> 10	> 1
Methodology existed prior to KBS developments	NO	NO	NO	YES	NO	YES	NO
Methodology adjusted - new developments	YES	NO	NO	YES	NO	YES	YES

Particulars	Organisation						
Methodology reviewed - end of development	**YES**	**YES**	**NO**	**YES**	**NO**	**YES**	**NO**
Methodology accepted by rest of organisation	**YES**	**NO**	**NO**	**YES**	**NO**	**YES**	**NO**
Methodology documented	**YES**	**NO**	**NO**	**YES**	**YES**	**YES**	**YES**
Training in methodology provided	**YES**	**NO**	**NO**	**NO**	**NO**	**YES**	**YES**
Contribution of methodology to successful development	**5**	**1**	**1**	**5**	**1**	**5**	**4**

In four out of seven organisations the development approach had been applied number of times. For Organisation B, the respondent claimed that the methodology had only been applied once the system had been released into the production environment and had not been used for development of the KBS prior to this point. Furthermore, the respondent for Organisation B indicated that the review of the development approach was very informal. Both Organisations D and F used a proprietary methodology which is normally targeted to conventional systems development rather than KBS development, and accordingly the methodology obviously existed before KBS developments were initiated by the organisation.

This contrasts with Organisations A and G which had developed specific KBS development methodologies rather than adapting an existing methodology. The difference between Organisations A and G is that for Organisation A this specific KBS development methodology has wider acceptance than just the KBS development area as in the case in Organisation G. Finally, it is interesting to note that for the four organisations which repeatedly used some form of methodology in their

KBS developments (Organisations A, D, F, and G) that the importance of the methodology to systems development success was either neutral (Likert score of "4") or only important (Likert score of "5") rather than higher.

With respect to the certification process the following information was obtained:

Table 42. Approval Process - KBS Development Methodology

Organisation	Approval Process
Organisation A	Approval process for our (organisation's own) methodology - close to CEO level. Approval for KBS components - workshops that make recommendations to methodology group to consider for next revision (of methodology) - quite a formal process
Organisation B	No formal approval process
Organisation C	N/A
Organisation D	Organisation's standard conventional development methodology - has a formal process to match ISO 9001 procedures. KBS development methodology is a more flexible document - informal approval.
Organisation E	N/A
Organisation F	External consultant.
Organisation G	No approval process. Informal agreement amongst members of KBS group.

Only two of the seven organisations had any formal certification procedure, one of which was in-house and at a very senior level within the organisation (Organisation A). The other organisation (Organisation F) placed lesser importance on this aspect, as the activity was assigned to an external consultant.

In answer to the question of how the methodology was developed, Organisation B felt the question was inapplicable, as KBS development occurred without the need of a methodology. For Organisation E the respondent also indicated that as the system was quite small, and therefore the development team was quite small, there was no need to follow a methodological approach. For Organisation B the fact that the system was now in production phase and there was more liaison with users and other MIS areas in the organisation, there was more of an organisational imperative to follow a methodological approach. For the other organisations a variety of sources were used with probably practical experience being the common factor across all organisations. The actual responses to this question are shown in the table below:

Table 43. Sources Used - Construction of KBS Methodology

Organisation	Methodology was developed by ...
Organisation A	Combination of research, practical experience and synthesis with existing IS methodologies.
Organisation B	Collection of best practices - usually as a result of people coming into the team. Also because changing the type of development - that is the application is nearly the end of its cycle. It is becoming more enhancement than development - which requires more predicability - that is a movement towards normal IS development practices and procedures.
Organisation C	N/A

Organisation	Methodology was developed by ...
Organisation D	Experience. Some inputs from outside sources. Standard software engineering methodology.
Organisation E	N/A
Organisation F	Literature. (Organisation's) standard methodology (Method 1 - from Andersons). Consultants - ADFA, and Andersons. US Internal Revenue Service. Research paper - Lo and Jeffrey (1990).
Organisation G	In-house; as a result of previous R&D work; modified as needed for different KBS's.

Finally, the approach used by each commercial organisation will be assessed using the framework developed in Chapter 2 of this report. The details of this assessment are presented in the table below:

Table 44. Assessment of KBS Development Approach

Particulars	Organisation						
	A	B	C	D	E	F	G
WHY: Underlying philosophy (15)	12	2	4	7	0	10	12
WHY: Raison d'Etre (10)	8	2	8	5	3	6	7
WHAT: An End Result (10)	10	5	10	10	5	10	10
WHAT: Tools and Techniques (15)	5	0	4	3	0	7	5
WHAT: Management Practices (5)	5	2	0	5	0	5	5

Particulars	Organisation						
	A	B	C	D	E	F	G
WHEN: Coverage (15)	10	3	0	6	0	6	9
WHEN: Coherence (15)	12	2	2	7	0	8	7
WHEN: Critical Processes (5)	5	0	0	3	0	4	0
HOW: Phases, etc. (10)	5	0	0	5	0	8	5
HOW: Tools and Techniques (10)	3	0	0	3	0	5	3
HOW: Additional training (5)	4	0	0	0	0	3	2
IF: Review & Quality Assurance (10)	6	0	0	8	0	8	0
TOTAL (125)	85.00	16.00	28.00	62.00	8.00	80.00	65.00

Justification for the following scores for each of the characteristics in the assessment framework is as follows:

- **WHY: Underlying philosophy (15).** For Organisation A, there is a reasonably close alignment with the Boehm development methodology, and close alignment with the view that prototyping will not produce viable production systems. There is also the consideration that the best process of developing these systems is still to initiate them in a research environment and transfer the development into the IT environment at the point of re-engineering the system. For Organisation B, an analysis of the system development notes and steering committee minutes indicated no underlying philosophy or approach to the KBS development, but more concentration on technical issues such as which C compiler to use. To some extent it is considered that Organisation C did have an underlying philosophy in the application of the scientific method to the development. Organisation D, similar to Organisation A applied a Boehm development approach although it is difficult to perceive any real underlying philosophy to their development approach, being an hoc mixture of the Organisation's standard development methodology supplemented by other informally developed in-house techniques. Organisation E did not appear to have any underlying philosophy for its development approach other than the discipline and rigour of the project manager. Organisation F used a standard development methodology supplemented other techniques. However, the underlying philosophy was that KBS development was not all that different to conventional IS development. Organisation G synthesised its development approach from research work performed by one of its team members.

- **WHY: Raison d'Etre (10).** For Organisation A, the justification for using a methodological approach was efficiency in the development process, and the ability to halt the development before significant amounts of resources had been used. For Organisation B, there was no real justification for the development approach other than survival. In fact, the steering committee notes reveal that the project was about to be cancelled a number of times. Organisation C justified the lack of a methodology by claiming that no real advantages would have accrued to the development by using one. Organisations D and F justified their development approaches by claiming that a lot of the development was standard software engineering. Organisation E justified its development approach on the size of the development. Organisation G justified its approach on the basis that management of the development was significantly improved for both the project manager and the technical development staff.

- **WHAT: An End Result (10).** In all cases these Organisations delivered a production KBS. However, in Organisation B development could have terminated for perceived lack of progress at certain points in the development. For Organisation E the KBS required a change in the hardware environment of the organisation before it could be successfully deployed.

- **WHAT: Tools and Techniques (15).** Very few organisations specified particular tools and techniques to use within the development process. In most cases these decisions were left to the project manager. For instance the principal development technique for Organisation C were pictures drawn on an electronic whiteboard. Organisation A considered that a methodology could have two levels, one management and the other technical. This organisation considered its methodology addressed the management level fairly well but had little at the technical level. It was considering applying the KADS development approach at the technical level. The other organisations used the tools and techniques thought best by the project manager. For instance, Organisation F supplemented the standard conventional development methodology with a range of tools and techniques thought appropriate for that particular phase of the development. For Organisations B, D and E there was almost no mention of appropriate development tools and techniques, that is Organisation D only mentioned the use of a design notebook. Finally there was little in common in the tools and techniques across the different KBS developments.

- **WHAT: Management Practices (5).** For Organisations A, D, F, and G there was a strong focus within the development approach on this aspect. For instance Organisation A stated that its development approach was more management oriented. The interview transcripts for Organisations D and F indicated that the development approach was selected more for management reasons than technical ones. For Organisation G the use of the in-house KBS development approach was justified more from a management point of view rather than a technical one. Organisation B had a steering committee overseeing the development, while the other Organisations did not have anything in this area. For instance, the respondent for Organisation C claimed that no formal management practices were followed, and for Organisation E a steering committee was created for the KBS only when it was deployed as a production KBS.

- **WHEN: Coverage (15).** There is a lack of this aspect in the development approaches for Organisations B, C and E, where there did not appear to be any planning, other than for the next month or so of the development. Events merely unfolded and problems were solved as they arose. For example, the KBS for Organisation C was deployed into the production environment with devastating effects on the efficiency of the hardware. The KBS had to be completely re-written in another expert system shell. The steering committee notes for Organisation B indicate a focus on the relevant issues still to be addressed without any overall framework for the development, and similar comments apply to Organisation E. Organisations A and G appeared to possess the clearest idea of a development framework for a KBS development. In Organisations D and F the framework was to all intents and purposes supplied by the standard conventional development methodology, and the interview transcripts indicate for both organisations that there was more focus on the immediate development issues rather than those further down the track.

- **WHEN: Coherence (15).** For this aspect Organisations B, C, and E would rate low due to the lack of any framework as discussed in the point above. For Organisations D and G there was only an informal certification process for the development approach, and in the case of Organisation D the development process was mainly an application of a conventional development methodology to the KBS development. The certification process for Organisation F was more formal, however, coherence in the approach would suffer as the approach followed here was adaptation of a conventional development methodology to the KBS development. Highest coherence of the development approach would apply to Organisation A as this approach had a formal certification process and was designed with KBS developments in mind.

- **WHEN: Critical Processes (5).** Consistent with the lack of a suitable development framework, Organisations B, C, and E did not indicate critical milestones in the development processes. For Organisations B and C these milestones were very much product related, both indicating that a critical event was the release of the initial production system. For Organisation E, there were only two critical events, approval to proceed and completion of the systems development. Similarly to Organisation E, the critical processes or milestones for Organisation G were product related and were identification of a suitable development followed by the delivery of the first production system. For Organisations A, D and F there was a far clearer delineation of critical milestones during the KBS development, with Organisation A possibly indicating this to a greater degree when compared to the other two organisations.

- **HOW: Phases, etc. (10).** There was no indication of phases for those KBS developments undertaken for Organisations B, C, and E. Accordingly, there is a concomitant lack of guidance in how activities within these non-existent phases are to be completed. In all cases it appeared that systems development relied almost entirely on the judgement and skills of the project manager with very little external assistance. With respect to this point for Organisation G, it is interesting to note the comments made by the respondent [interview transcript, page 27]:

"For example, take knowledge acquisition - there are more techniques than I have had hot dinners. And your job is to reach a certain output from a knowledge acquisition session, at get certain things. They are maybe twenty different ways you could reasonably take in with you, and expecting to use one, you might use fragments of three. And that I believe that happens a lot, and I believe that is the right thing to do. You know what you are trying to achieve and you have a tool bag of techniques to do it. And I used to talk about the tool bag of techniques, and that is how it should be. So I actually ... maybe what I am saying is that I think the methodology is the management process, and the technical stuff is technique."

It also appeared that for Organisations A and D, this was also the case, where the methodologies had more of a focus on management of the project and how a particular activity was performed within a particular stage of the development was left to the judgement of the project manager or members of the development team. As indicated by the documentation used within each phase of the KBS development it is considered that Organisation F had a greater focus on how the activities within the development should be completed.

- **HOW: Tools and Techniques (10).** Similar comments for all Organisations apply to this aspect of the assessment framework as were mentioned in the point above. If anything there appears to be less description of how particular tools and techniques were to be used within the KBS development than was identified with respect to the above point. For Organisations A, B, C, and F this may have been due to the level of experience of both KBS project managers as well as KBS development team members. It is interesting to note that this reason is not applicable to Organisations D, E and G in which team members had little or no experience within KBS developments.

- **HOW: Additional training (5).** Formal training in the methodology was provided by Organisation A. For Organisation F some training was provided to development team members by outside consultants, and informal training within the KBS development cell occurred for Organisation G. For the other organisations, namely Organisations B, C, E, and D no training in the development approach was provided.

- **IF: Review & Quality Assurance (10).** From the interview transcripts this aspect was identified more clearly for Organisation D, where there was a focus on quality because of ISO 9000 accreditation, and Organisation F, where specific checkpoints in terms of "GO/NO GO" and scoping decisions were made on the KBS development. For Organisation A, there was an emphasis on adopting TQM; however, the methodology indicated only one key milestone in terms of review, that is whether to proceed with the re-engineering of the development from the field prototype to a production system. For the other organisations, namely Organisations B, C, E, and G, an analysis of the interview transcripts revealed little evidence of review and quality assurance checkpoints within the development approach.

Therefore, on the basis of the above discussions, it appears that all organisations approached the system development in some methodical fashion. However, in terms of whether a KBS development methodology existed and was applied to the system development, it would appear that only two organisations could clearly fall into this category, namely Organisations A and F. On the other hand, it appears that Organisations B, C, and E clearly did not apply a development methodology. For Organisations D and G, the evidence does not appear to be conclusive one way or the other that KBS development for these organisations followed a methodological approach. The table below summarises these conclusions with regard to this question:

Table 45. Conclusion on Commercial KBS Development Approach

Organisation	YES	NO	MAYBE
Organisation A			
Organisation B			
Organisation C			
Organisation D			
Organisation E			
Organisation F			
Organisation G			

If the organisation does not have a KBS development methodology, are there any reasons for this?

The discussion for this question will focus on Organisations B, C, D, E, and G, which as indicated in the above table, did not clearly show that KBS development followed a methodological approach. The reasons for this varied considerably, however, there were a number of things that these organisations had in common. First, the KBS for all of these organisations were to all intents and purposes the first development the organisation had undertaken. Second, in four of these organisations (Organisations B, C, D, and G) KBS development occurred with a research and development type area rather than within the MIS/IT area of the organisation. (This was also true for Organisations A and F, which were both judged to have followed a methodology in their KBS development.) For the other organisation, Organisation E, although development occurred within the MIS/IT area, it was given minimum attention by senior IT staff. The conclusion to be drawn here is that KBS technology is still within an experimental phase for most commercial organisations and accordingly there is little organisational imperative to control KBS development by the application of a development methodology. Finally, for Organisations B, C, D, E, and G, the KBS development was controlled by people who had considerable experience in both conventional IS and KBS development.

Another interesting point to emerge from the analysis is that all KBS project managers did not perceive that application of a development methodology was a critical factor required to ensure development success. Indeed the evidence points in the opposite direction as three organisations (Organisations B, C, and E) that clearly did not use a methodological approach ended up deploying production systems. For Organisation C, development occurred on time, at a reasonable cost, and the system has had high acceptance by users and judged by a number of different factors to be very cost effective. A similar level of success cannot be applied to the systems for Organisations B and E, although system development did complete with a production system that appears to have growing acceptance within the user population. For these organisations there appeared to be a fundamental lack of planning (although whether a methodology would have solved this problem is a moot point), that led to long delays in development and ultimate delivery of the production system. For example, in Organisation E development of the first production version of the KBS took less than six months but the technology platform required to run the system was not available until six years later. Should this KBS development ever been initiated? In Organisation B, the steering committee minutes indicated that developed seemed to be plagued with a large number of quite different short term problems, none of which seemed to be foreseen, for example a production version of the system was released to insurance agents that had significant and difficult to resolve printing problems.

Another common factor across different organisations was the personality of the project manager. In Organisations B, C, and E where development did not follow a development methodology, the project manager for Organisation C did not believe methodologies were of any benefit. The project manager for Organisation E believed that the systems development was too small to justify creating and then applying a development methodology. Finally, the project manager for Organisation B considered that applying a systems development methodology, particularly in the early stages of development may place too much of a constraint on the development team. On the other hand, the project managers for Organisations F and G considered that use of a development methodology (in fact any methodology) greatly assisted expedition of the development. The two exceptions were considered to be Organisations A and D where use of the methodology seemed to flow more from an organisational imperative, rather than for purely personal reasons.

Finally, with respect to the KBS's developed by these organisations, there appeared to be two similarities across all organisations. First, in six out of seven organisations (the exception was Organisation D), the production KBS was considered to be a stand alone system with a limited amount of integration to other systems within the organisation, for example only a very limit amount of data sharing occurred. Furthermore, the types of problems addressed by these KBS's were relatively simplistic, with the possible exception of Organisation C. Excluding this organisation, the KBS's for Organisations A, B, D, F and G were judged to be either colleagues or assistants rather than experts. In the case of Organisation E even though the system was judged to be an expert at what it was doing, the problem was considered fairly simplistic in that most decisions made on consumer credit are checklist type procedures.

If the organisation does have a methodology, what features do these methodologies have?

For example, do these methodologies follow those that have been described in the research literature, follow the design of conventional systems development methodologies, or are unique in their structure? In terms of these development approaches gaining or applying the work published in the research literature, respondents were asked to indicate their level of familiarity with the following KBS development approaches that were discussed or published recently:

Table 46. Interest in Research on KBS Development Methodologies

Development Process	Organisation							AVG
	A	B	C	D	E	F	G	
KAAM	1	1	1	1	1	5	1	1.57
KADS	6	1	1	1	1	1	3	2.00
ES/SDEM	3	1	1	1	1	7	1	2.14
NesDEM	2	1	1	1	1	4	1	1.57
Turban methodology	1	1	1	1	1	1	1	1.00
Weitzel and Kerschberg	1	1	1	1	1	5	1	1.57
AVERAGE:	2.33	1.00	1.00	1.00	1.00	3.83	1.33	1.64

As indicated in this table below, these respondents do not consider the efforts of these researchers worthwhile in terms of providing insight into better ways of developing commercial KBS's. (In this case a score of "1" means that the respondent had no knowledge whatsoever of the research.) However, it is interesting to note that Organisations A and F, which were judged to develop KBS's using a development methodology had higher scores than for those organisations which did not. For example, the respondent for Organisation A indicated an interest KADS, to the extent that a workshop was run by a European consulting group which markets a commercial version of this development approach. Similarly, the respondent for Organisation F had reviewed the ES/SDEM approach, but did not apply to the KBS developments for this organisation.

If the organisation does have a KBS development methodology, what sort of documentation or other material exists describing the methodology?

In addition to documentation, are there other forms of educational support for the use of the methodology, for example training courses? From the interview transcripts, documentation outlining the specific KBS methodology exists for Organisations A and G. With respect to Organisations D and F the documentation is a combination of the standard conventional development methodology supplemented by additional documentation. For the other organisations, (Organisations B, C, and E) that were judged not to have used a system development methodology (either KBS or conventional) had no documentation on development approaches (even with respect to best practices that should be used in a KBS development. In fact for Organisation C, very little documentation existed at all for the KBS development.

If the organisation does have a KBS development methodology, how many times has this methodology been used?

What impact has the possession of a KBS development methodology had on the overall number of KBS developments for the organisation?

For instance, are those organisations that have development methodologies more likely to have developed more KBS's than organisations that do not have a KBS development methodology? These two questions will be considered together. With regard to those organisations that were considered to have a KBS development methodology (Organisations A and F), both had applied the methodology a number of times - approximately six times for Organisation A and in excess of 30 times for Organisation F. In fact, it is interesting to compare the impact of a development methodology on the KBS development program with the impact of a development methodology on a particular KBS development. Although KBS project managers did not consider that using a development methodology was important to the ultimate success of development, the only organisations that had continuing KBS development programs were organisations A and F (which were judged to have followed a methodological approach to the KBS development).

For organisations B, D, E, and G either one or two systems were developed and there is little expectation of further systems developments occurring. For Organisation E, the respondent made the following comments on the worth of creating and applying a development methodology [interview transcript, page 29]:

> *"Ian Caddy: I was just thinking of another organisation. So essentially would it just be a situation ... would it be correct to say that you believe that these systems can be developed effectively without the requirement of a methodology?*
>
> *Respondent: Well, yeah we have got proof of that! But it really depends on the system. I mean consumer lending is not typical of an expert system, in that it is quite simple from an expert systems point of view. I mean there are definitely complexities in it, but compared to you know scheduling or something more complex problems ... it is ... there is not really ... I don't think that there is a need for a methodology for something that simple. If you are getting into a project where you have multiple people working on it, and goes for a long time span, then it starts getting a lot more formal. But I think for one person for three months, who knows what they are doing ... for three months ...*
>
> *Ian Caddy: You spend more time complying to the methodology, rather than doing effective work.*
>
> *Respondent: Well sometimes that ... sometimes that is a good idea anyway. Because you have a bit more control over things, but in this case I don't think it was necessary."*

The respondent for Organisation E also felt that since no other KBS developments were expected, that it would be a waste of resources developing a systems development methodology that may never be used. Organisation C appears to be the exception where a limited number of other development opportunities have occurred, in which variations of the original KBS will be applied. For Organisation C the number of developments is still less than Organisations A and F, and is possibly less than would have been expected given the spectacular success of the initial KBS.

Chapter Summary

The major findings of this case study research concerning the application of KBS technology within commercial organisations is as follows. Overall, in six out of the seven organisations (Organisation A the exception), the introduction of KBS technology was relatively new, and these organisations could be classified as "feeling their way", even though production systems were delivered in all cases. This cautious somewhat conservative approach was evident in the levels of types of expertise offered by commercial KBS's. In five out of seven organisations the KBS operated at either colleague or assistant level (Organisations C and E the exceptions). For Organisation E, even though the respondent considered that the KBS operated at an expert level of performance, the problem domain itself was relatively simple.

In terms of the types of problem domains addressed by commercial organisation with KBS technology a direct functional comparison reveals obvious diversity. However, when the comparison is made at a more abstract level similarities begin to appear, namely the problems addressed by the organisations selected for this case study are mainstream problems rather than peripheral or unimportant ones, and are focussed predominantly on operational rather than higher level management type problems. However, even when the comparison is abstracted there is diversity. For example, the level of compulsion associated with each specific KBS displays a high degree of difference between these organisations, and even difference (with the exception of the public sector organisations) between organisations within the same industry, such as for the insurance and banking organisations.

With respect to the introduction of the technology into commercial organisations, in six out of seven organisations (the exception being Organisation E) this task was accomplished by either a research and development, special projects, or user area rather than the MIS/IT area. Furthermore, for six out of seven organisations (the exception again being Organisation E), the MIS/IT area was not involved in either identifying the problem as requiring a KBS solution or in the development of the KBS. This led to the situation where there was a significant difference between the levels of interest and support from the MIS/IT area and the area undertaking the KBS development, as shown in the table below:

Table 47. Difficulty in Introducing KBS Technology

Area introduced KBS technology	Difficulty experienced	
	MIS/IT	Other
MIS/IT	6.0	1.0
Other	2.6	4.6

The conclusion drawn here is that this provided the potential for disagreement or outright conflict to occur between these areas concerning the KBS development. From the interview transcripts this disagreement was evident in Organisations B, C, F, and G.

The problems addressed by the KBS development were in most cases long term ones (six out of seven organisations, with Organisation F the exception), although only in two cases had maintenance issues been considered (Organisations A and F). In other cases, either maintenance was not an issue (Organisation E), or was perceived to be part of the on-going evolutionary development process (Organisations C and B), or the system was only recently deployed (Organisation D), or the system had only low level of acceptance by users and so maintenance of the KBS was not an issue (Organisation G).

It was found that a relatively conservative approach was adopted by commercial organisations in the application of KBS technology. In five out of seven organisations (Organisations B and E the exceptions) the predominant formalism used for knowledge representation was either production rules or frames. Similarly, in five out of seven organisations (Organisations B and E the exceptions), the inferencing strategy followed either a forward or backward chaining process. In the case of Organisation E, the KBS was not considered to possess an inference engine as all rules were fired during a single consultation. System development occurred using expert system shells in six out of seven organisations (Organisation B the exception), and deployment of the production system used an expert system shell as part of the production environment in four out of seven organisations (Organisations A, B, and E the exceptions). In only one case (Organisation C) was an expert system shell used solely to deliver the production KBS.

The development environment for commercial KBS's was predominantly microcomputers - five out of seven organisations (Organisation A the exceptions in which development occurred on an AI workstation). Organisation A also deployed the production systems on an AI workstation. The production environment for the other organisations was predominantly microcomputers, the exception being Organisation C where the system was developed on microcomputers but deployed on a mainframe. For Organisations D and F mainframe and mini-computers formed part of the production environment principally as data storage and retrieval machines. The KBS that operated primarily on microcomputers.

Notwithstanding the fact that all seven organisations deployed production systems that were of some benefit to the organisation, the overall level of satisfaction with the KBS developments and the level of perception concerning the effectiveness of KBS developments was not high. The average Likert scores across the seven organisations were either "5" or lower for satisfaction with KBS developments, or "4" or lower with respect to the perception of the KBS group's effectiveness. Furthermore, these data were not collected from actual users, staff within the MIS/IT area, or senior management within the corporate or head office area, but were obtained from the KBS project manager! Accordingly, either more successful KBS developments are required to change people's perception of the technology, or more effort in terms of marketing the technology within commercial organisations is required. However, it seems that KBS developments are not significantly different with respect to these criteria when they are compared with conventional systems development as shown in the table below:

Table 48. Comparison of Conventional IS and KBS Developments

Particulars	Organisation							AVG
	A	B	C	D	E	F	G	
KBS developments								
Importance to organisation	6	3	6	2	3	5	1	3.71
Level of user involvement	6	1	1	3	6	7	5	4.14
Interest of senior management:								
Within MIS department	4	1	2	4	7	1	2	3.00
Outside MIS department	5	1	5	4	4	5	3	3.86
Effectiveness of completed systems	4	5	7	5	6	5	4	5.14

Particulars	Organisation							AVG
	A	B	C	D	E	F	G	
Integration with other systems	6	1	2	5	6	2	3	3.57
Conventional developments								
Importance to organisation	7	7	6	5	7	5	7	6.29
Level of user involvement	5	7	6	4	7	2	2	4.71
Interest of senior management:								0.00
Within MIS department	7	7	7	7	7	7	7	7.00
Outside MIS department	5	7	6	N/A	6	3	6	4.71
Effectiveness of completed systems	7	5	3	N/A	5	2	3	3.57
Integration with other systems	7	5	2	N/A	7	3	2	3.71

Are KBS developments different to conventional developments? For the organisation involved in this case study the answer appears to be "NO". For instance, the same sorts of activities and milestones that are often found in conventional systems developments occurred in these seven KBS developments. The types of documentation had a high level of similarity to what would be found in a conventional systems development. Finally, in most cases the personnel that were involved in the KBS development had previous conventional IS experience. That is there were very few KBS development staff used by commercial organisations that could be considered KBS development specialists. Even the knowledge engineers and KBS project managers all had previous IS experience, which in the case of the KBS project managers was often considerable, that is greater than 10 years.

An interesting comparison that has emerged from this analysis is that although the KBS developments were significant in terms of the elapsed development time, size of project team and project expenditure and number of users, and were successful in the sense that in all cases production systems were deployed, the likelihood of further developments at this stage are not high as shown by the table below:

Table 49. Likelihood of Future KBS Developments

Organisation	KBS Development Projects		Likelihood of more developments
	One development	Many developments	
Organisation A			YES
Organisation B			NO
Organisation C			YES
Organisation D			UNLIKELY
Organisation E			NO
Organisation F			UNLIKELY
Organisation G			NO

From the interview transcripts most respondents (five out of seven organisations, with Organisations A and C the exceptions) indicated that the organisation was either not going to use the technology in another development or was unlikely to use it. Many respondents claimed that factors such as the economic recession were the cause of these circumstances. This further indicates that commercial organisations still consider the use of the technology as an optional rather than essential.

The final major question of this chapter was concerned with whether commercial organisations constructed KBS's using a development methodology or not. Although the perception of the KBS project managers was as follows:

Table 50. Project Leaders' Perception: Use of a Methodology

Organisation	YES	NO	MAYBE
Organisation A			
Organisation B			
Organisation C			
Organisation D			
Organisation E			
Organisation F			
Organisation G			

in which most believed that the KBS developments followed some form of methodological approach. However, a more definitive analysis of the developments and answers given by respondents was made using the methodology assessment framework created as part of the literature review. This framework was discussed in Chapter 2 and validated against a number of existing methodologies. In terms of the KBS developments for the organisations involved in this case study the following information was derived using the methodology assessment framework:

Table 51. Assessment of Commercial KBS Development Approach

Particulars	Organisation						
	A	B	C	D	E	F	G
WHY: Underlying philosophy (15)	12	2	4	7	0	10	12
WHY: Raison d'Etre (10)	8	2	8	5	3	6	7
WHAT: An End Result (10)	10	5	10	10	5	10	10
WHAT: Tools and Techniques (15)	5	0	4	3	0	7	5
WHAT: Management Practices (5)	5	2	0	5	0	5	5
WHEN: Coverage (15)	10	3	0	6	0	6	9
WHEN: Coherence (15)	12	2	2	7	0	8	7
WHEN: Critical Processes (5)	5	0	0	3	0	4	0
HOW: Phases, etc. (10)	5	0	0	5	0	8	5
HOW: Tools and Techniques (10)	3	0	0	3	0	5	3
HOW: Additional training (5)	4	0	0	0	0	3	2
IF: Review & Quality Assurance (10)	6	0	0	8	0	8	0
TOTAL:	85.00	16.00	28.00	62.00	8.00	80.00	65.00

Accordingly, although five out of seven KBS project managers considered that they had applied a methodology to the KBS development, in actual fact only two organisations could be considered to have clearly followed a methodological approach (Organisations A and F), while another two organisations were more marginal (Organisations D and G), and finally three organisations clearly did not follow a methodological approach (Organisations B, C, and E).

In conclusion this analysis would indicate that a number of answers can be found for the research questions that were proposed in Chapter 3. A detailed discussion of these answers and other conclusions to emerge from the analysis performed will be outlined in the next Chapter.

Chapter 5: Discussion and Conclusions

Chapter Outline

Discussion and Conclusions

Limitations of Current Research

Future Research Issues

In the first section of this chapter, the discussion and conclusions concerning the four major questions will be covered. As an overall impression, it is considered that the research was a success in terms of the rich qualitative and quantitative data collected as well as the analysis that could be applied, and the conclusions that were able to be inferred. However, there were some limitations to the research which will be addressed in a later section of this Chapter. One major limitation was that with only seven organisations and seven KBS's no meaningful statistical analysis could be performed with respect the quantitative data collected.

Summary of Major Research Findings

Chapter 1 outlined four major areas of interest for this research project. In terms of the analysis performed on the research data, the following conclusions can be made:

Problems solved by KBS technology. Although commercial organisations investigated in this case study have adopted an experimental approach to KBS technology, the types of problems addressed have been mainstream or problems associated with core activities. This fact meant that KBS development as a significant activity that absorbed relatively large amounts of financial and other resources for developments that spanned more than 1232 months. Even though organisations were experimenting with the technology, they took this experimentation very seriously. Second, these problems are in the main operational problems which are predominantly structured problems that are knowledge intensive. KBS technology has yet to be applied and tested as an effective decision support tool for more unstructured or semi-structured problems faced at higher managerial levels within the organisation.

Application of KBS Technology. These commercial organisations have in the main been relatively conservative in the application of the technology representing knowledge in either production rules and/or frames, and adopting a forward or backward inferencing strategy. To some extent this has been driven by the development environment which in the main has used expert system shell software on microcomputers. However, a variety of production environments were in evidence for the production versions of these KBS's, ranging from microcomputers through workstations to mainframes.

KBS Development Approach. The study did not find any common development approach, which to some extent was due to the fact that organisations were experimenting with the technology. This meant that: (a) the development occurred outside of the MIS or IT area in most cases without the direction or oversight of this area; (b) personal views of project managers often drove the development process; and (c) the organisation as a whole saw no imperative to force the KBS development group to adopt conventional IS development approaches or to formalise the KBS development process.

These factors dominated the development approach despite the fact that the KBS development team was staffed mainly by IS professionals. Finally, a common feature for a large number of these KBS developments was the use in some fashion of the prototyping development technique.

KBS Development Methodologies. The fact that organisations were experimenting with the technology and that most organisations did not have an on-going KBS development program meant that only two out of the seven organisations could be considered to have used a KBS development methodology in their KBS development.

Detailed Discussion of Research Findings

Attitude of Commercial Organisations to KBS Technology

The first major finding to emerge from this research is the attitude of commercial organisations to KBS technology. As Takac and Lehner (1990) found in their 1990 survey, commercial organisations in this case study still appear to be experimenting with KBS technology. This inference is drawn from a number of issues. For example, as Bobrow et al. (1986) state, the often quoted R1 (later to be called XCON) system developed by DEC was almost cancelled three times. This experience also occurred in Organisation B, where many of the steering committee meetings were concerned about whether the project should proceed or be terminated. The system for Organisation E was developed in about 15 months, but then took another five years for it to be deployed. For these two organisations this situation seems to be inconsistent with an attitude where KBS technology is considered a mainstream information processing technology.

For the other five organisations, KBS development occurred under very different conditions, and so possibly the different inference could be made. However, a number of other indicators provide evidence that these commercial organisations are still experimenting with the technology as well, and these are discussed below.

First, in six out of seven cases the KBS development occurred outside of the MIS or IT area, undertaken either by outside consultants (Organisations C and D) or special development/research areas (Organisations A, B, F and G). Organisation A was to some extent an exception, as this organisation had the unusual or atypical practice of commencing KBS development within the research area, but then handing over KBS development at a particular point to the MIS or IT area (normally where the project sponsor indicated that development should proceed beyond field prototype). This is in contrast to the findings of Byrd et al. (1992), who state the following [p. 121]:

> *"With the increase in complexity of hardware and software systems, developers of traditional IS have begun to feel the need for ES technology. Applications such as automated help desks and "lights out" computer operations have helped to convince systems analysts of the value of ES technology (Popolizio and Cappelli, 1989 -* [in Byrd et al., 1992, p. 121]*). A recent survey conducted by New Science Associates shows that 35 per cent. of the Fortune 500 companies had explicitly turned their ES developments over to their traditional MIS groups (Popolizio and Cappelli, 1989* [in Byrd et al., 1992, p. 121])."

It would appear therefore that Australian commercial organisations lag behind their American counterparts in the level of acceptance of this technology.

Second, in six out of seven organisations there was no established on-going program for future KBS development - the exception being Organisation A. For two organisations there were no viable KBS development cells still in existence, namely Organisations F and G. In the case of Organisations B and E, interest in KBS related only to the on-going maintenance and enhancement of the sole production KBS. For Organisation D, there was no further contact with the outside consultants to do further or new KBS development work. There was some continuing interest in KBS development for Organisation C, but that was driven mainly by isolated proposals from user areas, and related to the spectacular success of the KBS development investigated in this research project. In summary, it would appear that most commercial organisations have a reactive strategy to KBS technology, that is apply the technology where the contingencies arise mainly as a result of user impetus, rather than a pro-active strategy where the organisation endeavours to search out suitable KBS developments.

Third, the attitude towards KBS technology by other areas of the organisation was found to be relatively low, and that there were differing levels of acceptance within the same organisation as indicated by the two tables below. Lack of acceptance for KBS technology was probably due to two factors, namely:

- As the MIS or IT area (except for Organisation E) was not championing the introduction of the technology, the primary information provider did not have any commitment or interest in either demonstrating its effectiveness, or marketing the concept throughout the rest of the organisation. Accordingly, use or application of the technology, without the focus of the MIS or IT area, led to a situation where isolated sections of the organisation (either a research and development group, new or advanced technology cell, or a particular user champion) made individual KBS development proposals, without any coherent plan to utilise this technology.

Table 52. Difficulty in Introducing KBS Technology

Organisation	Within MIS/IT	Outside of MIS/IT
Organisation A	4	6
Organisation B (a)	6, 2	6, 3
Organisation C (b)	N/A	7
Organisation D	3	5
Organisation E	6	1
Organisation F	2	5
Organisation G	2	4

(a) Initially "6" for both areas; other figure represents current situation.

(b) For this organisation the MIS area was almost completely ignored and so no value could be assigned. A surrogate value of either "1" or "2" could be used.

- The second point to emerge from this case study was the disparity of interest between the developers or champions of KBS technology, and either the ultimate users of the KBS or the provider of the information service (the MIS or IT area). As the table below indicates (although Organisation E was the only one in this category and so reservations should be made about this finding), where the MIS or IT area was introducing and developing the KBS interest was very high. However, this was coupled to an extremely low level of interest by the ultimate users of the KBS. On the other hand, where a user area or a research and development/advanced technology cell introduced the technology and then developed the KBS (which occurred in six out of seven organisations), even the interest, on average, of this area in the technology was only moderately high. Further, the level of interest by the MIS or IT area in the technology was either at a moderately or very low level.

Table 53. Difficulty Experienced in Introducing KBS Technology

Area introduced KBS technology	Difficulty experienced	
	MIS/IT	Other
MIS/IT	6.0	1.0
Other	2.6	4.6

This disparity of interest had a significant impact on a number of KBS developments. For example, with respect to Organisation E, although the first production version took fifteen months to develop, the system was not deployed as appropriate hardware did not exist within the production environment, that is the bank branches. It took another four years, in which on-going enhancement and development of the KBS occurred, to finally deploy the KBS. The steering committee minutes for Organisation B, as well as the comments of the respondent indicated that at many times KBS development was on the verge of being terminated. In fact the respondent commented that it was a miracle a production system was finally deployed. For Organisation

C, the MIS or IT area was almost totally ignored (except for implementation and production operational problems and liaison), which meant that further KBS development occurred only at the initiation of a user area. For Organisation F, there was hostility if not outright conflict between the end-user computing department within MIS and the KBS development cell, as indicated in the following comments [interview transcript, page 5]:

> *"Respondent: Part of it was to do with the struggle for control. The end user computing area wanted control of this area, and made no bones about it. They went all out four years ago to do so. My boss in the [organisation] prevented that, and ... ah ... I think the act of preventing it made it very difficult to sell [the KBS idea] to the IT area as a whole. So as a consequence, my strategy was to take on allies within the IT area and use them as champions. And used very much a top down approach over the section heads. To do that again I probably would not do it that way.*
> *Ian Caddy: Why not?*
> *Respondent: I think even though people like myself coming in out of [other] organisations ... if you want the structure that you create to survive in the public sector, then you need to do those [things] in a non-personal sense. And structures are very much associated with people."*

This factor, as well as the lack of focus across the organisation for KBS development, led to interesting problems in developing and deploying production KBS's. For example, during deployment for one particular KBS, the respondent for Organisation F made the following comments [interview transcript, page 22]:

"Respondent: Yeah, the pilot ... I guess as the pilot phase ended ... ah ... they are ... people just used the system when they needed to. The interesting thing was, we piloted in Hobart, Adelaide and Sydney - out at the Penrith Office. And, we initially thought, "Oh yes, we have kicked it off; it has been running for three to six months. That is it, we don't need it any more, and should now throw it away." We got a call from Moonie Ponds Office which is down in Melbourne, and they were saying, "We are having conflict problems with your installation." We had not done an installation there! What had happened ... the system works co-operatively with the mainframe for the database part. What they had done was that they had taken a tape from the Penrith Office, someone had scored a copy of it and [then] sent it down and loaded it. ... So what this enterprising manager had done, he got hold of the thing, because he had seen it here at a conference and taken it away with him. And within ... within eight months, these managers had distributed the software themselves, we had no control over the implementation process at all."

For Organisation G, the respondent indicated that fundamental differences existed between IS, which was "locked into very 1970's solutions" and had "a total unwillingness to even consider novel technologies" and the KBS area, which was located outside of IS, was regarded by the IS area as a "group of rocket scientists with all kinds of crazy ideas, and unproven technology".

Accordingly, as indicated in the interview transcripts all these organisations experienced some level of difficulty in developing and deploying production KBS's. To some extent this raised a perception, specifically in those areas with a connection to the KBS development (such as user and IS areas), but more importantly throughout the rest of the organisation, that there was a significant amount of risk in opting for a KBS solution to a commercial data processing problem. This perception had a tangible impact on the viability of the KBS development area. For Organisations B and E, the respondents envisaged no further KBS developments beyond the current production system. For Organisation F, although this organisation deployed a large number of KBS's (approximately 40), the KBS development cell ceased to exist when the respondent left the organisation. Organisation G presented an interesting case, where the IS area absorbed the KBS group from the research and development area (which itself was subsequently dissolved), and then effectively let it wither away. Even for Organisation A, the organisation with the longest association with KBS technology, the respondent indicated that an on-going program of KBS development would possibly not be viable without the enthusiastic support of a major division within the organisation which saw thousands of potential applications of the technology.

In summary therefore, all the indications were that on average, commercial organisations had not accepted and were still "experimenting" with KBS technology. In many cases for the organisations in this case study, the experimentation had finished. For these organisations this raises the problem of applying the technology at a later point in time, when the organisation will possibly re-invent the wheel - going through the same sort of process that it had just experienced. In these situations will this type of commercial organisation ever really accept the technology? On the other hand, KBS technology presents possibly a unique experience or opportunity concerning the introduction or re-introduction of new information processing technology into commercial organisations. That is, an interesting research issue to pursue would be to track this process using a longitudinal field study approach.

KBS "Problems" Addressed by Commercial Organisations

Bobrow et al. (1986) talk about KBS development following either the low road, middle road or the high road. The low road involves direct symbolic programming, usually in an AI language such as LISP. For almost all commercial organisations involved in this case study - the possible exception being organisation B - none of them developed systems in this manner; and in the case of Organisation B the system was developed in C++ rather than LISP or PROLOG. The high road on the other hand involves building a system that contains explicit representation of fairly complete knowledge of some subject matter, and can use the knowledge for more than one purpose. For this case study all of the commercial KBS's had more focus than flexibility, with decision support provided for one particular task. For example, the two banks involved in the case study both developed KBS's to assist with loan or credit applications. However, rather than looking at this function across all types of lending, both systems were narrow in their problem domain, namely one looked at consumer credit and the other looked at commercial credit only.

Another feature of high road systems is that they involve "deep" knowledge and long chains of reasoning from first principles to practical results. This should be compared with middle road systems which Bobrow et al. (1990) typify by using MYCIN as an example. MYCIN is considered to have short reasoning chains, the system has no overall "model of disease or health, no model of how diseases cause symptoms and no model of how treatment can cure diseases" [p. 881]. In all cases, with the exception of Organisation B, it would appear that commercial KBS's investigated were "middle road" systems. For system B1, an attempt was made to include deep knowledge as development centred around creating a model of risk for bodily injury. It is interesting to note the comments made by the respondent for this system, on other KBS developments [interview transcript, page 25]:

> *"As far as an approach to solving a problem, I think I ... the way we did [this system] ... I would want to do another system. But I think it is because of the systems that we target doing. So we don't target the [another organisation] systems where they take a simple thing ... I have this view of [that organisation]. They have a smelter right ... and they have a whole bunch of data about this smelter. And they induce behaviour, they run an induction against it. It is a mechanical device. So then ... having run that, you can do a little prototype. Here's the decision tree, let me do a prototype for you. And if that works, well then I will write it in C. Oh, blow - that it is not interesting! Because what we are trying to do is actually, not ... we have no mechanical device to induce anything from. Induction at the moment is not an option. Neural nets may give us something but I don't get insight, so I can't exploit it - that is useless for now. So what we are about doing is revolutionising the way insurance is done."*

Accordingly, although a number of authors (Keravnou and Washbrook, 1989; Horn, 1991) have indicated difficulty in determining what is a "deep" expert or knowledge based system, it would appear that the commercial KBS's investigated in this case study were very much first generation, "shallow" systems. This issue will be discussed further in the future research issues section of this chapter. Even when compared to MYCIN, in many cases these commercial KBS's were developed to operate at much less than expert level. In a number of developments, knowledge was extracted from paper documents such as procedure manuals, acts and regulations, rather than from human beings, for example Organisations A, D and F. For Organisation E, inferencing did not follow long chains of reasoning, in fact there was little inferencing at all, as all rules (or CASE statements) were executed during the one consultation.

In terms of the types of problems addressed with KBS technology by commercial organisations the following conclusions have emerged. First, even though organisations adopted an experimental approach to KBS technology, the attitude was in no way a mere "look and see". These experiments with KBS technology have been large and meaningful. In all cases, the problem addressed by the technology within the organisations investigated, were mainstream problems of significance to the organisation. Furthermore, for all seven organisations, KBS projects were undertaken by development teams of reasonable size (ranging from 3 to 15 personnel), had development schedules that spanned more than twelve months, had operating budgets of up to $1 million per annum, and were deployed to be used by significant user populations (in five out of seven organisations the user population was greater than 100). In six cases out of seven, the KBS's were deployed with the intention of an extended production life. Even the exception, Organisation F, the system was expected to operate for at least 18 months before it was to be de-commissioned.

The inference to be gained was that KBS technology should be tested in a fashion that demonstrates viability to operate effectively with real world commercial problems. This was necessary for it to have any credibility and future acceptance by these organisations. Small scale system developments would still leave the issue unresolved as to whether this technology could handle problems of a larger scale.

As a corollary to addressing mainstream problems within each organisation, and as these organisations operated to some extent in different industry environments, the types of problems addressed by KBS technology were varied. Furthermore, even for those organisations within the same industry (there were two banks, two insurance companies and two public sector organisations), the problems addressed were, at a detailed level, quite different. There are two main inferences to be drawn here. First, the variation of problems addressed was indicative of the flexibility or adaptability of KBS technology within the commercial environment. Second, the amount of variation was either indicative that organisations in their experimentation have adopted an ad hoc approach to problem selection rather than a more purposive approach, thereby engendering variety. Third, as with the comments for the respondent for Organisation A, it was possibly beyond the capabilities of the current KBS technology to design truly "turn key" systems that address generic problems across commercial organisations [interview transcript, page 2]:

> *"Respondent: ... But I suppose in that time his view has changed. I mean the inference was that if you wanted an expert blast furnace operator guidance system you would go and buy one. But it was found that this was not the case. Whilst there are probably twenty or thirty of them in the development stage around the world - none of them were in a package to such an extent that we could just design the interfaces to it."*

Whether or not the new generation of expert system shell software such as Art Enterprise (Hedberg, 1993) will make a difference would be an interesting research project to pursue in the not too distant future.

For six out of seven organisations, and there must still be some debate with respect to the seventh - Organisation C), the problem addressed by KBS technology was at the operational level rather than at a higher managerial level. Using the Gorry and Scott-Morton (1989) classification, the decision support in most cases (six out of seven organisations - the exception being Organisation C) was more for structured decision making processes, rather than unstructured. It seems that the assertion made by Gorry and Scott-Morton (1989) about conventional developments, that is a concentration on easily solved structured problems, can also be applied to current KBS developments. Accordingly, for the organisations involved in this case study, experimentation with KBS technology also meant a concentration on what Hickman et al. (1989) would designate "first generation" type KBS problems rather than second generation ones.

The final issue in terms of the types of problems addressed by KBS technology was the level of compulsion associated with using the KBS. As stated above, the type of decision support being provided was very much towards the operational level of the organisation rather than the higher managerial levels. When comparing conventional IS used at this level within an organisation, that is transaction processing systems, there is a high degree of compulsion. Staff at this level are directed to use the airline reservation systems, or purchasing/accounts payable systems, and so on.

Given that the production KBS's in this case study were deployed for use by operational staff, was there a similar level of compulsion in using a production KBS? It would appear that the answer was neither "YES" or "NO", but rather "MAYBE", and depends upon contingencies of the production environment within which the KBS operated. For instance, there was a high level of compulsion associated with two production KBS's for Organisations A and E. For Organisation A the operation of the blast furnace was through the KBS rather than either with it or without it. For Organisation E, a directive was issued that all personal credit applications were to be assessed using the KBS.

On other hand, for the KBS's in Organisations B, D, F, and G, use of the system was entirely optional, the system being invoked by the user whenever that user felt additional assistance in making a decision or carrying out a procedure was necessary. With respect to Organisation B, the system was supplied to independent insurance agents, and therefore that organisation had no ability to compel the agents to use the system. For Organisation C, there was no overt compulsion to use the system. However, the apparent advantages that were provided by the system meant that most bodily injury claims for this organisation were assessed with the assistance of the KBS. It would appear therefore, that on average, use of commercial KBS's was optional rather than mandatory.

Characteristics of Commercial KBS's

Hickman et al. (1989) claim that the first wave of commercial KBS were stand-alone small scale systems, usually based on some form of diagnosis, and usually built using a shell (a software package providing an empty knowledge base and a fixed inference structure). To some extent this case study supports the above claim, but not in all areas, as indicated in the discussion below:

- **Small scale, stand alone systems.** In all seven organisations the systems, by a number of different measures, could not be considered small. The only feature that could be considered small was the proportion of KBS development compared to new conventional IS development, which for all organisations never exceeded 10 per cent. of total expenditure on new development of conventional IS. With respect to the second point, only two systems, for Organisations B and C, were stand alone - although even for these systems there was still a certain amount of data passing between the KBS and conventional systems. The results from this case study differ markedly from the above claim.

- **Based on some form of diagnosis.** In Chapter 1, the following classification was proposed by Hayes-Roth et al., (1983) [p.14] which is now used, together with the KBS descriptions obtained, to classify the systems investigated in this case study:

Table 54. Classification of Commercial KBS

Category	Problem Addressed	System
Interpretation	Inferring situation descriptions from sensor data	Organisation A (blast furnace operational control and supervision).
Prediction	Inferring likely consequences of given situations	
Diagnosis	Inferring system malfunctions from observables	Organisation E (assess the suitability of a loan application for consumer type lending and credit). Organisation G (advise managers within wholesale banking areas on suitability of a commercial loan prospect). and Organisation C (assessment of bodily injury with respect to insurance claims).
Design	Configuring objects under constraints	

Category	Problem Addressed	System
Planning	Designing future actions or strategies	
Monitoring	Comparing observations to plan vulnerabilities	
Debugging	Prescribing remedies for malfunctions	
Repair	Executing a plan to administer a prescribed remedy	
Instruction	Diagnosing, debugging and repairing student behaviour	Organisation D (assist in the enforcement and interpretation of Commonwealth government legislation, regulations and procedures). and Organisation F (legal and administrative support and advisory system with learning facilities).

Category	Problem Addressed	System
Control	Interpreting, predicting, repairing and monitoring system behaviours	Organisation B (assist insurance agents in their dealings with prospective clients during the insurance proposal process).

As can be seen from the above table, only three out of seven KBS's could be classified as a diagnostic system. It is interesting that in two cases the organisation was a bank, and that the problem being addressed was similar, that is identifying applicants who were diagnosed as bad credit risks. It would appear that the solution to a particular problem was driven more by an assessment of the best problem solving approach, and not as Hickman et al. (1989) have claimed - that often problems are selected that suit the expert system shell, or that problems are modified in order to match the shell's capabilities. Accordingly, again there is no agreement with the original statement by Hickman et al. (1989).

- **Use of expert system shell software.** The findings of this case study support this claim. In six out of seven organisations, the exception being Organisation B, expert system shell software was used in the development of the commercial KBS, as well as forming part of the delivery environment for the production KBS. In only one organisation (Organisation C) was the development and delivery of a production KBS encapsulated solely within expert system software. It would appear that the productivity advantages of using this sort of software made it a very effective development tool. The comments of the respondents for Organisations F and G support this conclusion. For instance the respondent for Organisation G thought this environment ensured greater flexibility [interview transcript, page 24]:

"... Because the big thing that you give away when you go into a conventional implementation - you give away all the flexibility that you had bought yourself by using a KBS tool. And you should not give that away lightly. You should only give it away if you are driven to it. ... There are enormous advantages in staying with your KBS development tool for the field version. In fact I would begin to wonder about the benefits on going in to KBS, if they actually froze it in code. Because once you get .. let us say that you code the thing in C. You can kiss the specification goodbye, because it is now frozen in code. You are never going to change it."

- Whereas the respondent for Organisation F thought the productivity advantages were paramount [interview transcript, page 23]:

" ... You would have to re-engineer, and re ... the development tools are so automated now that in many instances it is better to go back to scratch. Take your documentation, read it, understand the domain, and then re-build. You don't go back and try and amend your prototype. You start again straight away. And particularly when you see tools like, oh what was one of them that they were playing with ... oh ... KnowledgePro ... It is a C++ system. You can produce .. you can produce a lexical object in fifteen minutes to half an hour. That is how quickly you

can produce them. There were seventeen hundred screens in this system - I guess that mapped down to about 900 ... 900 lexical objects. Ah ... I think ... um ... all of that was built within a six to eight month period."

To summarise the findings for this section, it would appear that experimentation by commercial organisations is not for mere technological demonstrations but rather a true and rigorous assessment of KBS technology's worthiness and suitability in a commercial environment. To do this the systems developed were large, had significant user populations, development teams and operating budgets of reasonable size. Furthermore, the types of problems in most cases were self-selected by areas other than the KBS development group, and although the most common type of system used some form of diagnosis, this problem solving category did not dominate. Use of expert system shells as a development environment dominated for reasons of productivity and flexibility. Given an increasing power and functionality being delivered in these sorts of development tools it is expected that this trend will continue, possibly with a higher proportion (than was found in this case study - one out of seven) of production systems operating solely through expert system shell software.

Development Approaches for Commercial KBS's

Kierulf et al. (1990) state that [p. 152]:

" ... conventional software engineering know how works best when producing the (n+1)st version of a compiler, text processor, or other well known software systems component or applications package. A small but important class of software development projects meets none of the conditions above. For good reasons,

such first of a kind projects are typically conducted in a manner diametrically opposite to traditional software engineering practice."

The organisations investigated for this case study indicate limited support for this statement. In two organisations (Organisations D and F) use of a proprietary conventional development methodology was applied to successful KBS development. The comments of the respondent for Organisation F were that any development methodology could be used, although it may take longer to develop the final production system. In the case of Organisation A, although there was a prototyping approach used in the initial phases of KBS development, where development was to proceed past the field prototype, then a conventional development approach was applied to produce the production system. However, for the other four organisations (Organisations B, C, E, and G) the interview transcripts and other documentation revealed either a relatively rudimentary evolutionary development approach (Organisations B, C, and E), or a non-conventional KBS development approach (Organisation G).

Notwithstanding the limitations of using prototyping (which were discussed in Chapter 2 of this report), it is still the principal KBS development technique reported in the research literature (for example, Loofbourrow, 1991; Liebowitz, 1991; Sacerdoti, 1991; see other references in Chapter 2). Liebowitz (1991) commented that often using this technique meant that KBS development would not realise a production system. Often KBS development was trapped in what appeared to be a never-ending cycle of producing intermediate prototypes. Loofbourrow (1991) claimed that the principal KBS development method was "rapid unstructured prototyping", by which it is assumed that the basic development approach was some form of evolutionary development marked by release, at ad hoc intervals, of intermediate versions of the production KBS. This case study found that there was no main KBS development technique used, and contrary to Liebowitz's claim all developments completed. Also contrary to the claims of the above authors, prototyping was not the principal development approach. For the majority of organisations investigated, namely for Organisations B, C, D, and E, KBS development proceeded in a evolutionary fashion, although all respondents (except for Organisation C) mentioned the development of prototypes. However, when looked more closely for Organisations B, D, and E there were none of the supporting procedures that would mark the development approach as a prototyping one.

With regard to the initial stages of KBS development, a number of authors have stated that the selection process is a critical factor in ensuring success with the KBS development (Liebowitz, 1991; Beckman, 1991). These authors have stated that use of checklists in this area may assist in better identifying those projects that should undergo a KBS development. The KBS development methodology ES/SDEM devotes a lot of attention to this phase, and has developed a checklist based procedure to assist knowledge engineers. However, for the organisations in this case study, selection of potential areas was more an ad hoc process, often driven by the user area wanting to apply a KBS solution to a problem. For example, in Organisation C, the user area championed KBS development after investigating a number of alternatives that were not considered viable. For Organisation G the interest of the user area was paramount in getting KBS development started. The exception to this general trend was Organisation E where the respondent conducted a survey to identify potential KBS developments (42 in all were catalogued). In conclusion, it would appear that other factors are more critical to the success of the KBS development than this one.

In terms of across the KBS development, a comparison of the findings in the case study will be made with the comments outlined by Sacerdoti (1991) to determine whether the case study findings are in accordance with similar studies completed overseas. First, Sacerdoti (1991) claims that almost every expert system has big chunks of conventional code. This case study would support that assertion as in five out of seven organisations, the KBS had additional components written in high level languages or application software packages. The entire system for Organisation E was written in a conventional microcomputer database programming language. Second, Sacerdoti (1991) claims that most KBS development projects evolve through the following phases:

- **Assessment and scoping, for example evaluate project costs and risks.** In only three out of seven organisations (Organisations A, D, and F) was a formal feasibility/assessment process conducted, whereas in the other four it was either an informal procedure or hardly done at all. That is, in one case the project sponsor merely asked the KBS project manager whether the system could be developed or not.

- **System architecture, which specifies all interfaces with existing computer- or paper-based systems as well as the selection of hardware platform for the production system.** The findings of this case study do not support this claim. For example, on selection of hardware, a range of responses were obtained, with two organisations stating that no selection was necessary (Organisations C and D) to the careful selection of hardware as the system had to operate on portable microcomputer equipment at a reasonable level of efficiency (Organisation B).

- **Initial prototype, which is a small 20-rule system in your first days or weeks.** On this point the case study offers limited support. For Organisations A, F and G the respondents stated that limited effort was put into the conceptual prototype, often less than a man-month. On the other hand the development of the first prototype for Organisation D was also connected with selling the system and so substantial effort was invested into its development, particularly the user interface. With respect to other organisations (Organisations B, C, and E), the development was more evolutionary and so development of a conceptual prototype, if it occurred at all, was not seen to be as important as in the other KBS developments.

- **Incremental development phases, in which the project is broken into phases, each lasting no more than six months, with each phase terminating in a running system, and that early prototypes are thrown away.** The results of this case study do not support this claim. In only two cases (Organisations A and F) did the respondents indicate any phased development of the KBS. It is interesting to note that these were the only organisations judged to possess a KBS development methodology. In most of the other cases, the next key milestone after approval to proceed with development was the release of a production system.

- **System roll-out, in which all interested paries should participate.** In three cases (Organisations B, C, and E) system roll-out consisted of on-going releases of upgraded or new versions of the production system, with little input from users or the MIS/IT area.

- **Maintenance, enhancement and support**. This is discussed in a following paragraph.

In conclusion the findings of this case study research for Australian commercial organisations appears to be quite different to the experiences within American commercial organisations. At present the contingencies associated with each development mean that practices across organisations differ more than they have in common.

What were the findings after KBS development had completed, that is the maintenance of commercial production KBS's? First of all Agarwal et al. (1991) claims that incremental development (either using a prototyping development approach or not) will lead to problems with maintenance of the on-going production KBS. The organisations reviewed in this case study do not support this claim. (The production KBS for Organisation D was released recently, and so is not considered in the following discussion.) For Organisation E, although the system has undergone some maintenance, virtually the whole of this activity has been applied to other parts of the system, and not to the knowledge base. Organisation C, which adopted an incremental approach during KBS development (only two major releases of the development system occurred), had successfully extended and refined the system over the last few years since the systems went into production. It is interesting to compare these findings with the organisations considered to have KBS development methodologies. For Organisation A, the production system was deployed in an environment where users (or suitably qualified personnel within the user area) performed their own maintenance of the knowledge base. For Organisation F, agreements were made with user areas that specified the length in service of the production KBS. Once this point had been reached, the user area could either fund re-development of the KBS, or have the KBS de-commissioned.

Use of KBS Development Methodologies

Hilal and Soltan (1991) claim that the issue of KBS development methodologies has been a "hot topic" within the AI community in recent years, leading to the development of products such as KADS (Hickman et al., 1989), KEMRAS (Alvey Project, 1988), and POMESS (Diaper, 1988). These authors then go on to state that lack of a suitable development methodology is often the major cause of unsuccessful commercial KBS developments. However, as this case study demonstrates, lack of a suitable development methodology did not mean that developments failed. Three out of seven organisations (Organisations B, C, and E) did not apply a systems development methodology in any form but followed an incremental, evolutionary development path that eventually led to the deployment of a production KBS. In two other organisations (Organisations D and G) the KBS development was only considered to have marginally followed a methodological approach. Accordingly, at least for the organisations in this case study, presence of a methodology was not a critical success factor.

The other interesting issue followed up was the transfer of the research effort into KBS development methodologies into commercial organisations. For all seven organisations involved in this case study, the respondents, who were the KBS project managers, had little overall awareness of efforts in this area (in fact five out of the seven organisations were judged to have no awareness whatsoever). Accordingly, although it may have been a "hot topic" in the research area, these respondents showed that little transfer had occurred, indicating that these respondents either did not see any value in the research effort, or if there was some inherent value it was not applicable to the types of KBS developments completed. While organisations are still experimenting with KBS technology this low level of interest is expected to continue.

It is interesting to note that the two organisations with a higher interest than the others, Organisations A and F, were also the organisations judged to have applied a KBS development methodology in the construction of the KBS. However, the type of interest shown by these respondents was focussed upon improving their own development process rather than a general interest. For instance, the respondent for Organisation A had a high understanding of the KADS methodology because the organisation had recently conducted a workshop in a KADS derivative marketed by Bolesian Inc. Similarly, the respondent for Organisation F indicated a very high understanding of ES/SDEM because of an evaluation performed to determine the applicability of this methodology. The conclusion to be drawn here is that some form of "commercialisation" of the methodology research effort is required before system developers in commercial organisations develop an interest.

In terms of assessing whether or not these organisations had KBS development methodologies, the methodology assessment framework outlined in Chapter 2 was applied. Sol (1983) suggested that there were five ways of assessing different methodologies:

- **Using an idealised methodology and then evaluating other methodologies against this idealised framework.** However, a fundamental problem with this approach is the development of an ideal KBS development methodology as few true KBS development methodologies exist, and then gaining consensus within the general AI and KBS community of this ideal. Accordingly, this approach was not considered viable.

- **Select a suitable set of features that are judged to be "good" and then determine if the candidates have these features.** The problem here is similar to the first strategy - what are considered to be the good (or ideal) features of a methodology. Furthermore, if this problem can be resolved what weighting should each feature possess when compared to all the others? Another problem with this procedure is where a particular candidate does not possess one or more of the above features. Can a valid assessment of these candidate cannot be made?

- **A third approach is to classify the features required from a method by allocating a priority to each feature.** However, this approach appears to be little different from the one above, and accordingly is considered to suffer from the same difficulties.

- **The fourth approach was to define a meta-language in which a frame of reference would be constructed to describe the methodologies.** However, Hilal and Soltan (1991) claim that this approach is hindered by the limited syntactic power of the underlying meta-language to generate a meaningful comparison or assessment mechanism.. This approach can also suffer from anomalies that arise as a result of the translation process that takes the methodology and re-expresses it in the meta-language.

- **The final approach was a contingency approach, assessing methodologies with respect to environment limitations.** This approach is not considered viable as the variety of problems, and other contingencies would make a meaningful assessment in this area difficult if not impossible.

The SDM assessment framework coupled with the system development methodology continuum developed in Chapter 2 is considered to be superior to each of the above approaches for the following reasons:

- There is no attempt to define an ideal methodology, but rather to assess each methodology on its own merits.

- There are no valued judgements made about the attributes used in the assessment framework, as the candidates are allowed to have more of a particular, attribute and less of another without an assessment being made at that stage on whether the candidate is a methodology or not. However, there is some attempt to weight certain attributes as being more important than others, which in turn will mean that candidates with high scores in these areas, are more likely to be considered a methodology.

- As indicated in the point above, scoring for attributes or features is more akin to an analog process. That is, the assessment framework allows different candidates to have more of a feature or less of a feature rather than trying to determine whether the candidate has a particular feature/attribute or not.

- There is no attempt to re-formulate the methodology into something different. Each methodology is assessed on the characteristics that it possesses. Comparisons can be made at the end when the methodology is placed on the SDM continuum.

- There is no attempt to consider any contingencies that may impact on the methodology from development environment. As stated above, the methodology is assessed on its own merits and not on it's broad applicability to different development environments. That is, the methodology is not assessed on its generic flavour, although it would be expected that a KBS development methodology is applicable to all KBS developments.

As a result of applying this assessment framework and the SDM continuum, it was considered that only two organisations (Organisations A and F) definitely applied a KBS development methodology to their development, with two other organisations in a more marginal position (Organisations D and G). Finally, three organisations were assessed not to have used a KBS development methodology (or even any sort of system development methodology) in the creation of their production KBS's.

It would appear from this case study, that the lack of a suitable development methodology occurred for the following reasons:

- In most cases development occurred outside the MIS or IT areas, and so there was less of an organisational imperative to develop these systems in this manner. Even if the assertion by the respondent for Organisation G is true, that often conventional development methodologies are flouted more than adhered to, system development within the MIS or IT area occurs within a milieu that has an abundance of methodologies to either use or ignore. The same cannot be said for KBS development.

- Because of the above, there was a greater impact of the personality and philosophical attitude of the project manager on the development process. For example, in Organisation C the project manager was averse to using any form of methodology and so system development proceeded in an evolutionary fashion. On the other hand, the project manager for Organisation F adopted the opposite view and required KBS developments under his control to follow that organisation's standard development methodology (with certain additions and modifications).

- Finally, the lack of an on-going program in KBS development meant that the respondents for a number of organisations had no desire to expend scarce resources (both intellectual or otherwise) on the creation of a systems development methodology that may never be used, or used only intermittently.

The final point to be made here, is that although the application of a development methodology to a KBS development is not necessary for the successful completion of that project, use of a methodology may lead to more efficient and effective KBS developments. This then has a flow on effect of lowering the perception by users of development risk for KBS developments and so ensuring a greater likelihood of establishing an on-going program of KBS developments. In this case study two of the three organisations (Organisations A, C, and F) which had on-going KBS developments also applied development methodologies to the system development (Organisations A and F). With respect to the other organisation, Organisation C, the number of additional developments was well below that of the other two organisations.

Lessons to be Learned From the Case Study

What can be gained from this research project and given to the new KBS director or project manager for an organisation that is either in the process of extending use of KBS technology, or about to apply KBS technology to problems within the organisation? For the successful use of KBS technology, this research project indicates the following strategies should be adopted:

- More attention should be paid to the selection of potential KBS developments that address a decision support problem. This area, for most of the organisations involved in this case study, the selection process appears to be driven by the user areas rather than the KBS development team. That is the approach of the KBS development team should be more pro-active rather than reactive. In doing so, this raises the potential of establishing an on-going program of KBS development (but as the situation with Organisation E, does not guarantee it).

- Ensure that if KBS development occurs outside of the MIS or IT area, that closer ties are established with this area. For example, select operational decision support problems that require a high level of integration with existing conventional systems. Rather than creating a climate of competition and increasing the likelihood of hostility with the MIS or IT area, a climate of co-operation should be established.

- Given that the organisation has an experimental attitude to the technology, the test of the technology should be to select a decision support problem that has the following characteristics:

  - the KBS will support a core or mainstream activity of the organisation;
  - the problem should be at an operational level rather than at a higher managerial level within the organisation; and
  - the production KBS should interact with as large a user population as possible.

- In terms of system development the KBS project manager should consider the use of expert system shell software, and the application of traditional knowledge representation and inferencing strategies. The hardware development environment will most likely be microcomputers. However, with the eventual deployment of a production KBS, the project manager should be aware that often the KBS will require integration with other information systems. Furthermore, issues such as performance, may require the re-development of the final prototype produced using the expert system shell software. On the other hand, potential KBS developers should be aware that expert system shell software technology is still evolving and becoming more powerful. Products such as ART Enterprise (Hedberg, 1993) which operates across multiple hardware platforms, may allow production KBS's to be delivered within an expert system shell environment - with all the productivity and flexibility in production that this software can offer at present in development.

- Where possible, the project manager should select either a KBS development methodology or adapt a conventional systems development methodology (say one based on an iterative development approach), to guide the development process.

Limitations of the Current Research

Questions not Addressed in the Case Study

The major limitation of this research project was the current environment and attitude of commercial organisations to KBS development. After enjoying a high degree of popularity and exposure in the late 1980's the technology is to some extent out of favour in most commercial organisations. For instance, only two organisations (Organisations A and C) are currently engaged in further KBS development. For two other organisations (Organisations B and E) the current KBS development is the sole KBS development. In organisations F and G the KBS development group has been dissolved, and for Organisation D, the client has not instituted any further KBS development projects. A number of organisations related this decline to economic circumstances, as well as the tentative rather than enthusiastic acceptance by commercial organisations for the technology.

Accordingly, given this environment some of the research questions proposed earlier were not proceeded with during the interview phase of the project. In particular, there were a number of questions concerning multiple KBS developments within a single organisation, namely:

- Are there similarities between the problems addressed within a single organisation?

- Within a single organisation are the development approaches similar for different KBS developments?

- Within a single organisation is the application of the KBS technology similar across different KBS developments?

In this case study only two organisations (Organisations A and F) had effectively delivered more than one production KBS. For Organisation F, the respondent had subsequently left the organisation, and the KBS development cell had been closed down. Therefore additional information concerning the other KBS developments was unavailable. Accordingly, as only one organisation had access to information on multiple KBS developments, it was considered that a meaningful answer to these questions should be gained once KBS development activity is restored to those levels experienced in the late 1980's.

The other major limitation in terms of research questions concerned the development and use of systems development methodologies in the construction of commercial KBS's. As only two organisations were found to definitely possess KBS development methodologies, one an adaptation of a proprietary conventional SDM and the other an in-house developed product based on the spiral model (Boehm, 1988), no further work was done in this area.

Limitations of the Research Data

The major limitation with the research data was the inability to obtain, for all organisations suitable metrics and other quantitative data for the KBS development. In some cases this sort of information did not exist (Organisations C and E), either due to the size of the development, or the management practices of the project manager. In two other cases (Organisations F and G), the respondents responsible for the KBS development had subsequently left the organisation and so access to this sort of documentation was difficult or impossible. Accordingly, quantitative analysis could only be performed on the case study data collected through the interview questionnaire. In addition the limited number of organisations investigated meant that no meaningful statistical analysis of the quantitative data could be performed. However, for most research questions there was a clear indication of commonality across these organisations (often six out of seven organisations), and therefore providing high external validity to the research findings.

Limitations of the Research Method

The major limitation of the research method was that the case study was retrospective in nature. That is, the KBS's investigated were already developed and either in production or had been in production and were now de-commissioned. Accordingly, this raised problems for some questions within the questionnaire. The answers obtained were often a subjective opinion which relied on the respondent accurately recalling what happened - which in some cases was a number of years ago. Furthermore, as these developments were now completed the information gathered for them may not reflect current practice. Furthermore the opinions of these project managers would necessarily have also changed as a result of this KBS development as well as other, and due to other advances to the technology that have occurred during this time.

Another limitation relates to external validity, as these organisations may not provide a representative cross section of all organisations engaged in KBS development. It is hoped that further case study research will be conducted, particularly for organisations located in other states such as Victoria.

Future Research Issues

Commercial KBS's - First or Second Generation Systems?

With regard to whether commercial organisations are still developing first generation or second generation knowledge based systems, Barbuceanu (1991) states [p. 234]:

> *"Two major ideas have reshaped our understanding of knowledge acquisition and expert system construction. The first*

> *is that knowledge acquisition is essentially a modelling process (Clancey, 1989). This is in contradiction with the traditional "transfer" view according to which knowledge is extracted from human experts and translated into some executable representation language. In the model based view knowledge engineers actively build models of expertise in the same way scientists build theories to explain phenomena in their domains.*
>
> *The second major idea is that this modelling should be carried out at a conceptual level that abstracts from implementation level detail. This is Newell's knowledge level. Knowledge acquisition carried out at the knowledge level as a modelling activity has been termed second-generation knowledge acquisition (Akkermans, et al., 1990)."*

Keravnou and Washbrook (1989) also claim that first generation KBS's, while reaching high levels of performance and deriving the right recommendations in terms of problems addressed suffer from a number of limitations. Hickman et al. (1989) support this claim stating that one of the problems with these so-called first generation expert systems has been the "co-operativity problem". In the consultation that proceeds between a human expert and somebody else there is a great deal of flexibility in which hypotheses may be suggested and examined, facts or ideas clarified. Hickman et al. (1989) claims that expert systems have tended only to model the expert inferences and not the co-operative aspect of problem solving. The result has often been that such systems are unusable since they simply do not fit into the environment traditionally occupied by the human expert.

Keravnou and Washbrook (1989) are more specific than Hickman et al. (1989) and state that limitations with first generation KBS's can be classified into three main areas, namely human-computer interaction, flexibility, and extensibility. Second generation KBS's should therefore provide more functionality in these areas, that is possess better human-computer interaction, have more flexibility in their operation, and demonstrate that they can adapt (rather than be re-engineered as was the case with Organisation F) over time. Furthermore, Keravnou and Washbrook (1989) claim that there is no general agreement on what identifies a deep or second generation system. Accordingly, future research could be conducted using the taxonomy shown below (Keravnou and Washbrook, 1989, p. 206) to determine whether or not organisations now possess second generation KBS's and the reasons for their development.

Table 55. First Generation Limitations - Commercial KBS's

Problem solving flexibility
• Monolithic, rigidly applied reasoning: - Inability to dynamically plan its reasoning strategy for a specific case, based on the characteristics of that case. - Orthogonal strategies not supported. • Performance degrades dramatically when dealing with difficult (rare) cases. • Inability to recognise that a problem case is at the periphery or outside or its area of expertise.
Human computer interaction
• Inadequate user interface: - Information required to be entered in very specific terminologies and formats.

- Historic information on a case not maintained. • Inadequate dialogue structure: - System raises incoherent or redundant questions.
- User not allowed to volunteer information of focussing guidance. - User not allowed to revoke an answer, or to pursue the effects of an alternative answer (to see "what if ..."). • Inadequate explanation structure: - "Explanations" are just rule playbacks, and not meaningful. - Explanations are not user-tailored and do not cover all the explanation needs of the user.
Extensibility (maintainability)
• Difficult to modify the system knowledge, both manually and automatically; consistency checks not facilitated. • Inability of the system to evolve on the basis of its experiences in problem solving.

There are indications that the system developed for Organisation B is a second generation system. For example, the following comments were made by the respondent on the problem solving ability of the KBS [interview transcript, page 20]:

"Respondent: Well, the model is correct when it predicts a diversity of behaviour, the diversity of behaviour that we have to predict. So, we start out with a model, say we had something about leg injuries. We

> *start with a model that deals with leg injuries. And that is a fairly complex model in itself. So that has to deal with the scale of leg injuries. From a bump - it has to do that properly, and has to give you the right judgement for that, to the most ... an amputation after ten operations - and give you a sensible answer for that. So that at both ends of the spectrum the model works; so it has to pass that test. And then what you have to ... try push it in other directions as well. So that when you stretch the model, and propose what are boundary type conditions that the model behaves properly. If it behaves at the boundary and behaves well in the middle, then we are starting to get pretty happy about the model."*

However, the other organisations are still considered to have first generation KBS's. To verify this contention a future research project could apply the above taxonomy to each of these KBS's to determine whether or not these systems can be considered either first or second generation ones.

Development Methodologies for Commercial KBS's

The results of this case study research showed that few organisations have reached a threshold in their use of KBS technology to begin addressing issues such as the use of a KBS development methodology. Overall, the thrust within Australia seems to be that there were not enough KBS's either built or proposed to spend the time, effort and frustration in building a KBS development methodology, or to go to the cost of buying proprietary KBS development methodologies such as Bolesian's SKE (Loofbourrow, 1991). However, given a renewed emphasis in applying KBS technology to decision support problems within commercial organisations and the consequent desire to establish an on-going program of KBS developments, then at some time this threshold will be reached where there is an impetus to develop, or acquire and then change/refine a KBS development methodology. As such additional case study research could then be undertaken in the future to attempt an answer for each of the following research questions:

- **What are the principal factors that move an organisation into addressing methodological issues with respect to its KBS developments?**

  Do these factors include: (a) number of KBS's developed; (b) adoption and use of the technology by the MIS or IT area; (c) recognition by individual project managers that use of a development methodology increases the likelihood of development success as well as improving the productivity of the development process, and the effectiveness of the delivered product; (e) the size of KBS development projects expected, or under way (either individually or collectively); (f) the impact of the KBS's on the organisation's operations.

  With respect to point (c), Wilson et al. (1989) state that a methodology has obvious benefits in that system developers have a better ability to plan, and to estimate resources for the development project, to estimate the size of the development, to have a framework within which to monitor and adapt development, and have the documentation used in monitoring and controlling the development. Will these benefits be ones that are perceived by KBS project managers as worthwhile?

- **If KBS development methodologies are built, what structure will these products take on?**

It would appear that three alternative paths could be pursued, namely:

- Specific KBS development methodologies are created, such as KADS, or a proprietary KBS development methodology such as Structured Knowledge Engineering by Bolesian, which is based on KADS.
- Particular development processes are created as add-ins to more conventional system development methodologies. For example, Wilson et al. (1989) report on the Gemini project in which KBS development procedures are to be developed that will integrate with the SSADM conventional system development methodology.
- Conventional systems development methodologies are adapted to KBS development. In this case study, the two organisations (that is Organisations A and F) which were judged to have KBS development methodologies both followed this approach. For Organisation A the KBS development methodology followed to a large extent Boehm's (1988) spiral methodology, while that for Organisation F followed a standard proprietary methodology marketed by one of the large accounting firms.

It would be interesting to determine which of the above paths was followed by commercial organisations and the reasons for their particular choice. This area, given that few organisations have KBS development methodologies also presents the opportunity to undertake a field study to determine how development methodologies are constructed (specifically KBS development methodologies) to resolve questions such as:

- How does a methodology differ from the best practices derived from a number of system development projects that an experienced project leader applies consistently to system developments? What factors determine whether a particular process, tool or technique is absorbed into a KBS development methodology, or is left just as a best practice?

- Does the level of integration of KBS's with other systems within an organisation drive the emphasis to use a system development methodology?

- Are the earlier phases more important than later, and so more attention, discussion and documentation are generated for these phases rather than the later ones?

- Should the methodology provide the framework or focus for the system development in order to ensure with changes of staff, particular at the project leader level, that continuity is provided? Or is this problem trivial, and so not a factor that forces organisations to adopt a development methodology (conventional or KBS)?

- Is the main aim of producing a KBS development methodology to provide training and assistance to novice project leaders and system developers, but merely paid only lip service by experienced project leaders system developers?

- Is the construction of a KBS development methodology the shared accumulated common sense or insightful conclusions/revelations made by one person or more of a group effort, with the active involvement of a broad cross-section of the organisation?

- In the development of the KBS methodology, will there be more focus on the management aspects of the development rather than the technical (for the conventional SDM's ISAC and SSADM the reverse is the case)? That is, the KBS development methodology will be more concerned with the effect of poorly estimated phases for knowledge acquisition, than with whether an effective knowledge acquisition tool or technique has been applied in this phase of the KBS development.

Automating KBS Development

Will the future in KBS development lie in this direction? The Spang Robinson report for December 1991-January 1992 (Vol. 8, No. 1), stated that [pp. 22-23]:

> *"A research effort is underway at DEC that is taking a holistic approach to automating tasks in the workplace. Led by John McDermott, famed builder of the XCON system, the Technical Director of DEC's AI Technology Centre, the goal of this two year old effort is to design and implement an integrated programming framework, "Easy Programming", of reusable modules of code for a broad range of tasks."*

As the Spang Robinson Report states [p. 23]:

> *"What sets this work apart from most CASE efforts is its close attention to the entire work context and the sensitive issues of technology insertion. "We [AI programmers] weren't paying enough attention to the contextual issues in the workplace. We were trying point solutions", observes McDermott. "The most critical issue is a better understanding of the problem for automating, and finding what computational assistance is needed." Now, taking an interdisciplinary approach, the team is drawing on the social science and anthropology to model their workplace and identify situations where th automation of co-operating agents (workers) will facilitate and complement the work. "A task is no longer an island you automate, McDermott believes. "This is very misleading and simplified. In reality, the workplace is an ocean with islands and relations. We are looking for pieces of automation that fit in."*

Will there be the development of CASE tools for KBS, or will CASE tools which exist at present incorporate intelligent components within them and be generalised to produce KBS's? An appropriate answer to this research question may have to wait until there is far greater acceptance of the technology that at present by commercial organisations.

Methodology Assessment Framework

More research effort is needed to refine the methodology assessment framework developed in Chapter 2 of this report. For example, the different weights assigned to each attribute within the framework should be examined, particularly by commercial organisations to determine whether the current scheme is appropriate. In addition, further work should be directed to establishing suitable scoring procedures which will ensure consistent values being derived for each of the attributes within the framework. This will ensure greater consistency in the application of the framework by different people as well as across time by the same person.

Finally there should also be more validation of the framework with assessments made for techniques such as prototyping (in both forms, that is "throw-away" as well as "keep-and-enhance), and specific knowledge acquisition techniques such as CDM and KAAM.

Commercial Development of KBS's

As alluded to above, further research should also be conducted to perform case studies for organisations that have an on-going development program of KBS development to complete the two-way contingency analysis presented in Chapter 2 with respect to single organisations.

This technology as it is new to organisations would also provide a unique opportunity to perform a longitudinal case study to assess whether the Nolan (Gibson and Nolan, 1974) stages of acceptance of technology are appropriate. In most cases the organisations in this case study had developed either one KBS or a small number of KBS's. It would be interesting to track the development of further KBS's, from the point of view of the processes and procedures used, for these organisations.

Chapter Summary

As the final chapter of this report, this chapter presents the major findings and conclusions of the research. It is considered that a far better understanding of how a broad cross section of commercial organisations have applied KBS technology, as result of carrying out this research.

However, there are limitations to the findings which are due to:

- the area is one where change and innovation are always apparent and so the findings of the case study, particularly as it was a retrospective case study may not accurately reflect the current "state of play";

- that a number of respondents had left their organisation and so could not report on current events within that organisation; and

- the fact that at present KBS development is quite depressed, primarily due to economic circumstances, in organisations may have had some impact on the relevance of the findings for future years when interest in using the technology increases.

Notwithstanding these limitations, this research project is considered to be a success in terms of its major findings. That is in the application of KBS technology, commercial organisations:

- mainly have not used development methodologies to create production KBS's;

- are still considered to be experimenting with the technology and have not accepted the technology as a viable alternative to conventional solutions;

- have applied the technology to provide decision support for problems that occur at the operational level within the organisation rather than at higher, more management oriented problems;

- have used expert system shell software primarily as a development vehicle, and in combination with other software as a production environment;

- have selected significant mainstream problems within the organisation to prove that the technology can provide substantial benefits when applied properly.

The other indication of the success of this research project is the foundation or launching pad it provides for further research in this area, namely:

- What factors will lead to the transition from first to second generation KBS's within commercial organisations?

- What factors are necessary to establish an on-going program of KBS development within commercial organisations?

- What factors would lead to the MIS or IT areas of commercial organisations adopting and marketing KBS technology to the rest of the organisation?

- Will commercial organisations move towards a more methodological approach to the development of KBS's? If so, what factors will drive this trend?

- If more commercial organisations adopt KBS development methodologies, what characteristics will these products possess?

- Will KBS development methodologies be developed in-house, be purchased as proprietary systems, or be additional modules provided within conventional system development methodologies?

- What impact will advances in technology such as intelligent CASE have on the development of commercial KBS's?

- What impact will advances in the capability and power of expert system shell software have on commercial KBS developments?

In addition further work should also be devoted to the extension and refinement of the SDM assessment framework and the SDM continuum approach to the evaluation of system development methodologies, as well as development tools, techniques and procedures.

This chapter marks the end of a long journey in which the level of knowledge about the application of KBS technology within commercial organisations has been substantially raised. Like all successful research not only were the original questions answered, but the research also uncovered a number of other issues and questions worthy of further investigation. The fact that a better understanding of how commercial organisations at present have applied KBS technology will provide substantial assistance for those organisations who are at present contemplating the future use of this technology.

Another indicator of the success of this research project has been the unexpected. Along the way a number of interesting things have emerged, the most important being development of the assessment framework for systems development methodologies. This is considered to have made a major contribution in both determining what systems development methodologies are, as well as providing a process to assess and compare different methodological approaches.

Bibliography

Adelman L., "Measurement Issues in Knowledge Engineering", IEEE Transactions on Systems, Management and Cybernetics, Vol. 19, No. 3, May/June 1989.

Agha G., "Concurrent Object-Oriented Programming", Communications of the ACM, Vol. 33, No. 9, September 1990.

Akkermans J.M., Freiling M.J., Shuman S.J., Rehfuss S., Messik S.L., "Ontological Analysis: An Ongoing Experiment", Proceedings AAAI Workshop on Knowledge Acquisition for Knowledge Based Systems, Banff, Canada, 1986.

Alavi M. and Joachimsthaler E.A., "Revisiting DSS Implementation Research: A Meta-Analysis of the Literature and Suggestions for Researchers", MIS Quarterly, Vol. 16, No. 1, March 1992, pp. 95-116.

Alty J., "Expert System Building Tools", in "Topics in Expert System Design: Methodologies and Tools", by G. Guida and C. Tasso (Editors), Elsevier Science Publishers, North Holland, Amsterdam, 1989, ISBN 0-444-87312-X.

Alvey Project No. IKBS/098, KEMRAS, British Hydromechanics Research Association, 1988.

Anjo Anjewierden J.W. and Toussaint C., "Shelley - computer-aided knowledge engineering", Knowledge Acquisition, No. 4, 1992, pp. 109-125.

Applegate L.M., Konsynski B.R. and Nunamaker J.F., "Model Management Systems: Design for Decision Support", Decision Support Systems, Vol. 2, 1986, pp. 81-91.

Agarwal R., Linck R. and Tanniru M., "Knowledge Base Maintenance", Expert Systems, Summer 1991, pp. 6-14.

Athas W. and Boden N., "Cantor: An Actor Programming System for Scientific Computing", Proceedings of the NSF Workshop on Object-Based Concurrent Programming, G. Agha, P. Wegner, and A. Yonezawa (Editors), ACM, New York, April 1989, pp. 66 - 68, Special Issue of SIGPLAN Notices.

Australian Broadcasting Commission, Four Corners program, "National Mutual", reporter M. Westfield, 15 February 1993.

Avison D.E. and Fitzgerald G., "Information Systems Development: Methodologies, Techniques, and Tools", Blackwell Scientific Publications, Oxford, 1988.

Bachant J. and McDermott J., " R1 revisited: Four years in the trenches", AI Magazine, Volume 5, No. 3, March 1984, pp. 21 - 32.

Bachant J. & Soloway E., "The Engineering of XCON", Communications of the ACM, March 1989, Vol. 32 No. 3, pp. 298-317.

Bailey J.J., "Economic Evaluation of Knowledge-Based Systems", Master of Science Paper, Massachusetts Institute of Technology, Boston, Maryland, 1987.

Bailin S.C., "An object-oriented requirements specification", Communications of the ACM, Vol. 32, No. 5, May 1989, pp. 608 - 623.

Baldwin D. and Kasper G.M., "Towards Presenting Management Domain Knowledge", Decision Support Systems, Vol. 2, 1986, pp. 159-192.

Barbuceanu M., "Models: Towards Knowledge Level Modelling Environments", Australian Workshop on KA for KBS, August 21-23, 1991, pp. 234-259.

Barker E. and O'Connor D.E., "Expert Systems for Configuration at Digital: XCON and Beyond", Communications of the ACM, March 1989, 32(3), pp. 298-317.

Barthelemy S., Edin G., Toutain E. and Becker S., "Requirements analysis in KBS development", ESPRIT Project P1098 Deliverable D3 (Task A2), Cap Sogeti Innovation, 1987.

Basili V.R. and Ramsey C.L., "Arrowsmith-P - A Prototype Expert System for Software Engineering Management", IEEE Proceedings of the Expert Systems in Government Symposium, McLean, Virginia, October 24-35, 1985.

Baskin A.B. and Michalski R.S., "Experience with ADVISE and related programs", in "Topics in Expert System Design : Methodologies and Tools", by G. Guida and C. Tasso (Editors), Elsevier Science Publishers, North Holland, Amsterdam, 1989, ISBN 0-444-87312-X.

Beck R. and Cunningham H., "A laboratory for teaching object-oriented thinking", in Proceedings of OOPSLA '89, New Orleans Louisiana, October 1989, SIGPLAN Notices (ACM), Vol. 24, No. 10.

Beckman T.J., "Selecting Expert System Applications", AI Expert, February 1991, pp. 42-48.

Beinat P. and Tsui E., "Extending COLOSSUS to Cover Compensation for Future Economic Loss", Intelligent Systems in Accounting, Finance and Management, Vol. 1, 1992, pp. 275-287.

Bielawski L. and Lewand R., "Expert Systems Development - Building PC-Based Applications", QED Information Sciences Inc., Wellesley, Massachusetts, 1988, ISBN 0-89435-239-3.

Biggs S.F. and Selfridge M., "GC-X: A Prototype Expert System for the Auditor's Going Concern Judgement", Working Paper, University of Connecticut, January, 1986.

Blanning R.W., "Knowledge Acquisition and System Validation in Expert Systems for Management", Human Systems Management, 1984, pp. 280-285.

Bobrow D.G., Mittal S., and Stefik M.J., "Expert Systems: Perils and Promise", Communications of the ACM, Vol. 29, No. 9, September 1986.

Boehm B.W., "A Spiral Model of Software Development and Enhancement", IEEE Computer, May 1988, pp. 61-72.

Booch, G., "Object Oriented Design with Applications", The Benjamin-Cummings Publishing Company, California, 1991, ISBN 0-8053-0091-0.

Borgida A., Greenspan S., and Mylopoulos J., "Knowledge representation as a basis for requirements specification", IEEE Computing, Vol. 18, No. 4, 1985, pp. 82 - 101.

Bowerman R.G. and Glover D.E., "Putting Expert Systems into Practice", Van Nostrand Reinhold Company, New York, 1988, ISBN 0-442-20842-1.

Boynton A.C. and Zmud R.W., "An Assessment of Critical Success Factors", Sloan Management Review, Summer 1984.

Brachman R., "On the epistemological status of semantic networks", in Findler N. (ed.), "Associative Networks", Academic Press, New York, NY, 1979.

Braden B., Kanter J. and Kopcso D., "Developing an Expert Systems Strategy", MIS Quarterly, Vol. 13, No. 4, December 1989, pp. 459-467.

Brightman H.J., Ramakrishna H. & Elrod R., "Matching Problem Diagnostic-Tools to Managers Decision Styles - A Contingency Approach", Omega-International Journal of Management Science, 1988, Vol. 16, No. 1, pp. 1-9.

Brookes C.H.P., "A Framework for DSS Development", Information Systems Forum Report 84/1, University of New South Wales, 1984.

Brown C.E. and Murphy D.S., "The Use of Auditing Expert Systems in Public Accounting", Journal of Information Systems, Fall 1990, pp. 63-72.

Brown C.E. and Phillips M.E., "Expert Systems for Management Accountants", Management Accounting, January 1990, pp. 18-23.

Bubenko, J.A. (1986): "Information Systems Methodologies - A Research View", in "Information Systems Design Methodologies: Improving the Practice", by Olle T.W., Sol H.G., and Verrijn-Stuart A.A. (editors), Elsevier Science Publishers, B.V. North-Holland, 1986, pp. 289-318.

Buchanan B.G. and Feigenbaum E.A., "DENDRAL and META-DENDRAL: their applications dimension", Artificial Intelligence, Vol. 11, pp. 5-24.

Buchanan B.G., Sutherland G.L. and Feigenbaum E.A., "Heuristic DENDRAL: A program for generating explanatory hypotheses in organic chemistry", in Meltzer B. and Mitchie D. (eds.), "Machine Intelligence 4", Edinburgh University Press, Edinburgh, 1969, pp. 209-254.

Burns R.N. and Dennis A.R., "Selecting the Appropriate Application Development Methodology", Database, Fall., 1985.

Byrd T.A., Cossick K.L. and Zmud R.W. (1992): "A Synthesis of Research Requirements Analysis and Knowledge Acquisition Techniques", MIS Quarterly, March 1992, pp. 117-138.

Caddy I. and Stephens G. (1992): "Raider: Case Study of a Knowledge Based Internal Audit Tool", Expert Systems in Information Management, Vol. 5, No. 1.

Caddy I.N., Jamieson R., and Stephens G.M., "A Framework for Developing Expert Systems in Auditing and an Example: RAIDER", Proceedings of AI'90, Joint Australasian Conference on Artificial Intelligence, Perth, November, 1990.

Canning R., EDP Analyser, Canning Publications Inc., Volume 23, No. 12, December, 1985.

Canning R., EDP Analyser, "The One Minute Knowledge Engineer", Volume No. 25, No. 3, March, 1987, pp. 13-14, Canning Publications Inc.

Carter W.B., Beach L.R. & Inui T.S., "The Flu Shot Study: Using Multi-attribute Utility Theory to Design a Vaccination Intervention", Organisational Behaviour and Human Decision Processes, Vol. 38, 1986, pp. 378-391.

Catchpole C.P., "Information Systems Design for the Community Health Services", PhD Thesis, Aston University, Binningham, 1987.

Checkland P.B., "Systems Thinking, Systems Practice", John Wiley, Chichester, UK, 1981.

Checkland P.B., "Systems Thinking and Computer Systems Analysis: Time to Unite", DEC Seminar Series in Information Systems, presentation given at the London School of Economics, 5th November 1987.

Ching M., "The Importance of Evaluation in Expert System Development", Special Topics in Information Systems, December 1987.

Clancey W.J., "The epistemology of a rule based system - a framework for explanation", Artificial Intelligence, Vol. 20, pp. 215-251, 1983.

Clancey W.J., "Knowledge-Based Tutoring: The Guidon Program", MIT Press, Cambridge Massachusetts, 1987.

Clancey W.J., "The Knowledge Level Reinterpreted: Modelling How Systems Interact", Machine Learning, Vol. 4, No. 3/4, December, 1989, pp. 285-293.

Clements D., "Practical Expert Systems : Technology Overview and DIAS Reference Manual", Glenlake Publishing, Perth 1989.

Coad P. and Yourdon E., "Object Oriented Analysis", Prentice Hall, Englewood Cliffs, New Jersey, 1990, p. 240.

Colmerauer A., Kanoui H., Pasero R. and Roussel P., "Un Systeme de Communication Homme-machine en Francais", Research Report, Groupe Intelligence Artificielle, Universite Aix-Marseille II, France, 1973.

Compton P., "Knowledge Base Maintenance without a Knowledge Engineer", Research Seminar Presentation, Centre for Information Technology Research, School of Information Systems, University of NSW, 1991.

Compton P., Horn R., Quinlan R. and Lazarus L., "Maintaining an expert system", in Quinlan J.R. (Editor), "Applications of Expert Systems", London, Addison-Wesley, 1989, pp. 366 - 385.

Cook T., "Expert Systems: Panacea or Just a Niche Tool?" in OR/MS Today, June 1991, pp. 6-8.

Cox, B.J., "There is a Silver Bullet", Byte, October 1990, pp. 209 - 218.

Cranswick K., "Just yesterday's buzzword?", Computing, June 1990, p. 21

Cupello J.M. and Mishelevich D.J., "Managing Prototype Knowledge/Expert System Projects", Communications of the ACM, Vol. 31, No. 5, May 1988.

Cutts G., "Structured Systems Analysis and Design Methodology", Paradigm Publishing Ltd., London, 1987, ISBN 0-948825-86-3.

Davis M.W., "Applied Decision Support", Prentice Hall, Englewood Cliffs, N.J., 1988, ISBN 0-13-040072-6, Chapter 1, pp. 1-10.

Davis R., "Interactive Transfer of Expertise: Acquisition of New Inference Rules", Artificial Intelligence, Vol. 12, No. 2, 1979, pp. 121-158.

Davis R. and King J., "An overview of production systems", in Elcock E.W. and Mitchie D. (eds.), "Machine Intelligence 8", John Wiley, New York, 1977.

Davis G.B., "Comments on the Critical Success Factors Method for Obtaining Management Information Requirements in Article by John Rockhart", MIS Quarterly, September 1979.

Davis G.B. and Olson M.H., "Management Information Systems : Conceptual Foundations, Structure and Development", pp. 564-598, 2nd Edition, McGraw-Hill Book Company, 1985. ISBN 0-07-Y662412-X.

de Greef H.P. and Breuker J.A., "Analysing system-user co-operation in KADS", Knowledge Acquisition, No. 4, 1992, pp. 89-108.

Diaper D., "The Promise of POMESS: a people orientated methodology for expert system specification", Proceedings of Human and Organisational Issues of Expert Systems, Stratford-upon-Avon, England, 4-6 May, 1988.

Dolk D.R., "Data as Models: An Approach to Implementing Model Management", Decision Support Systems, Vol. 2, 1986, pp. 73-80.

Downs E., Clare P. and Coe I., "Structured Systems Analysis and Design Method: Application and Content", Prentice Hall International (UK), Hermel Hempstead, Hertfordshire, 1988, ISBN 0-13-854324-0.

Duda R., Gaschnig J. and Hart P., "Model Design in the PROSPECTOR Consultant System for Mineral Exploration", in Mitchie D. (ed.), "Expert Systems in the Microelectronic Age", Edinburgh University Press, Edinburgh, 1979, pp. 153-167.

Duff C. and Howard B., "Migration Patterns", Byte, October 1990, pp. 223 - 232.

Eshelman L., "MOLE: A knowledge acquisition tool that buries certainty factors", Proceedings of the Second Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada, 1987.

Flanagan J.C., "The Critical Incident Technique", Psychological Bulletin, Vol. 51, No. 4, 1954, pp. 327 - 358.

Fiegenbaum E.A., "Knowledge Engineering in the 1980's", Department of Computer Science, Standford University, Standford, CA, 1982.

Feigenbaum E.A. and McCorduck P., "The fifth generation", Addison-Wesley, New York, 1983.

Fischhoff B., "Eliciting Knowledge for Analytical Representation", IEEE Transactions on Systems, Management and Cybernetics, Vol. 19, No. 3, June 1989.

Floyd, C. (1986): "A Comparative Evaluation of System Development Methods", in "Information Systems Design Methodologies: Improving the Practice", by Olle T.W., Sol H.G., and Verrijn-Stuart A.A. (editors), Elsevier Science Publishers, B.V. North-Holland, 1986, pp. 19-54.

Forgy C.L. (1982): "Rete: a fast algorithm for the many patterns/many objects match problem", Artificial Intelligence, Vol. 19, pp. 17-37.

Forgy C.L. and McDermott J. (1977): "OPS, a domain-independent production system language", International Journal of Computing and Artificial Intelligence, Vol. 5, pp. 933-939.

Forman E.H., "AHP Is Intended for More Than Expected Value Calculations", Decision Sciences, Vol. 21, 1990, pp.670-672.

Forsythe D.E. and Buchanan B.G., "Knowledge Acquisition for Expert Systems: Some Pitfalls and Suggestions", IEEE Transactions on Systems, Management and Cybernetics, Vol. 19, No. 3 May/June 1989.

Fox M.S., "AI and Expert System Myths, Legends, and Facts", IEEE Expert, February 1990.

Fox G., Johnson M., Lyzenga G., Otto S., Salmon J., and Walker D., "Solving Problems on Concurrent Processors, Vol. I, General Techniques and Regular Problems", Prentice Hall, Englewood Cliffs, New Jersey, 1988.

Garg-Janardan C. and Salvendy G., "A conceptual framework for knowledge elicitation", in Knowledge Acquisition Tools for Expert Systems, edited by Boose J. and Gaines B., Academic Press, Harcourt Brace and Jovanovich, London, 1988, ISBN 0-12-115920-5.

Gashnig J. et al., "Evaluation of expert systems : issues and case studies" in Hayes-Roth et al., 1983.

Giarratano J.C. and Riley G., "Expert Systems: Principles and Programming", PWS-Kent Publishing Co., Boston Massachusetts, 1989, ISBN 0-87835-335-6.

Gibson C.F. and Nolan R.L., "Managing the Four Stages of EDP Growth", Harvard Business Review, January/February 1974, p. 76 ff.

Goranzon B. and Josefson I. (Editors), "Knowledge, Skill and Artificial Intelligence", p. 4, Springer-Verlag, London, 1988, ISBN 3-540-19519-X.

Gorry G.A. and Scott-Morton M.S., "A Framework for Management Information Systems", Sloan Management Review 1989, Vol. 30, No. 3, pp. 49-61.

Goul M. and Tonge F. (1987): "Project IRMA: Applying Decision Support System Design Principles to Building Expert-Based Systems", Decision Sciences, Vol. 18, No. 3, Summer 1987, pp. 448-467.

Gruber T.R. and Cohen P.R., "Design for acquisition : principles of knowledge-system design to facilitate knowledge acquisition", in Knowledge Acquisition Tools for Expert Systems, edited by Boose J. and Gaines B., Academic Press, Harcourt Brace and Jovanovich, London, 1988, ISBN 0-12-115920-5.

Guida G. and Tasso C. (Editors), "Topics in Expert System Design : Methodologies and Tools", Elsevier Science Publishers, North Holland, Amsterdam, 1989, ISBN 0-444-87312-X.

Hall A.D., "Three-dimensional morphology of systems engineering", IEEE Transactions on Systems, Management, and Cybernetics, Vol. SMC-5, No. 2, April 1969, pp. 156 - 160.

Hackathorn R.D. and Karimi J., "A Framework for Comparing Information Engineering Methods", MIS Quarterly, Vol. 12, No. 3, September 1988, pp. 203-220.

Harmon P., "A Brief Overview of Software Methodologies", Intelligent Software Strategies, VII, No. 1, pp. 1- 19, Newsletter. Circulation office: 37 Broadway , Arlington, MA 02174, USA, 1991.

Harmon P. and King D., "Artificial Intelligence in Business", 1985.

Hayes S., "The Use of Expert Systems Technology", Workshop, 1990 Australian Joint Artificial Intelligence Conference, Perth, December 1990.

Hayes S., "Introduction to Expert Technology and its Application", Workshop, Parts I and II, 1990 Australian Joint Artificial Intelligence Conference, Perth, December 1990.

Hayes-Roth F., Waterman D.A. and Lenat D.B., "Building Expert Systems", Addison-Wesley Publishing Company, Reading Massachusetts, 1983, ISBN 0-201-10636-8.

Hayward S.A., Wielinga B.J., and Breuker J.A., "Structured analysis of knowledge", in Knowledge Acquisition Tools for Expert Systems, edited by Boose J. and Gaines B., Academic Press, Harcourt Brace and Jovanovich, London, 1988, ISBN 0-12-115920-5.

Hedberg S., "New Knowledge Tools", Byte Magazine, July 1993, pp. 106-111.

Henderson-Sellers B. and Edwards J.M., "The Object-Oriented Systems Life Cycle", Communications of the ACM, Vol. 33, No. 9, September 1990.

Henderson J.C. & Schilling D.A., "Design and Implementation of Decision Support Systems in the Public Sector", MIS Quarterly, Vol. 9, No. 2, June 1985, pp. 157-169.

Hilal D.K. and Soltan H., "A suggested descriptive framework for the comparison of knowledge based systems methodologies", Expert Systems, Vol. 8, No. 2, May 1991, pp. 107-114.

Hickman F.R., Killin J.L., Land L., Mulhall T., Porter D. and Taylor R.M., "Analysis for Knowledge-Based Systems: A practical guide to the KADS Methodology", Ellis Horwood, Chichester, UK, 1989.

Hockman Z. and Pearson C.J., "Evaluation of knowledge based systems in agriculture: a case study", in Proceedings of Australian Workshop on KA for KBS, Pokolbin, Australia, Aug. 21-23, 1991, pp. 157-166.

Hopple G.W., "Decision Aiding Dangers: The Law of the Hammer and Other Maxims", IEEE Transactions on Systems, Management and Cybernetics, Vol. 16, No. 6, June 1986, pp. 834 - 843.

Horn W., The Challenge of Deep Models, Inference Structures and Abstract Tasks", Applied Artificial Intelligence, No. 5, 1991, pp. 87-96.

Horsfall A., "X-JOB Overview", Working Paper, Fujitsu Australia Ltd., 1990.

Houdeshel G. & Watson H.J., "The Management Information and Decision Support (MIDS) System at Lockheed-Georgia", MIS Quarterly, Vol. 11, No. 1, March 1987, pp. 127-140.

Hutchins E.L., Hollan, J.D., and Norman D.A., "Direct Manipulation Interfaces", Technical report 8503, Institute for Cognitive Science, University of California, San Diego, May 1985.

Huang D., "A Framework for the Credit-Apportionment Process in Rule-based Systems", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 3, May/June, 1989, pp. 489-498.

Irgon A., Zolnowski J, Murray K.J., and Gersho M., "Expert System Development - A Retrospective View of Five Systems", IEEE Expert, June 1990.

Jackson I.F., "Corporate Information Management", Prentice Hall International Inc., 1986, ISBN 0-13-174236-1.

Jackson P., "Introduction to Expert Systems", 2nd Edition, Addison Wesley Publishing Company, 1990, ISBN 0-201-17578-9.

Jalote P., "Functional refinement and nested objects for object-oriented design", IEEE Transactions Software Engineering, Vol. 15, No. 3, 1989, pp. 264 - 270.

Jamieson R. and Szeto R., "Preliminary investigation of the Impact of Knowledge Base Information Systems on Organisations", Working Paper, June 1987.

Jamieson R., "Knowledge Based Expert Systems Development within the Auditing Domain", Conference Proceedings of the Accounting Association of Australia and New Zealand, Adelaide, August, 1986.

Johnson P., Woudhead R. and O'Brien J. (1989): "The Ford Motor Company Direct Labor Management System", AI Expert, August, pp. 42-52.

Johnson P.E., "What Kind of Expert Should a System Be ?", The Journal of Medicine and Philosophy, Vol. 8, pp. 77 - 97, 1983.

Johnson P.E., "The Expert Mind : A New Challenge For the Information Scientist", IFIP, 1984.

Judd C.M, Smith E.R., and Kidder L.H., "Research Methods in Social Relations", 6th Ed., Holt, Reinhart and Winston, Inc., Fort Worth, 1991, ISBN 0-03031149-7.

Kahn G., Nowlan S., and McDermott J., "MORE: An intelligent knowledge acquisition tool", Proceedings of the Ninth International Conference on Artificial Intelligence", Los Angeles, California, 1985.

Kahn G.S. and Bauer M., "Prototyping : Tools and Motivations", in "Topics in Expert System Design : Methodologies and Tools", by G. Guida and C. Tasso (Editors), Elsevier Science Publishers, North Holland, Amsterdam, 1989, ISBN 0-444-87312-X.

Kelly G.A., "The Psychology of Personal Constructs", Norton, New York, 1955.

Khoo S.P., "Knowledge Base Systems Development Methodology: Knowledge Acquisition, Knowledge Representation in Organisational Domains", RT2 Presentation, 1986.

Keravnou E.T. and Washbrook J., "What is a deep expert system? An analysis of the architectural requirements of second-generation expert systems", The Knowledge Engineering Review, No. 4:3, 1989, pp. 205-233.

Klein G.A., Calderwood R.A., and McGregor D., "Critical Decision Method for Eliciting Knowledge", IEEE Transactions on Systems, Management and Cybernetics, Vol. 19, No. 3, May/June 1989.

Kidd, A.L. (Editor), "Knowledge Acquisition for Expert Systems : A Practical Handbook", Plenum Press, New York and London, 1987, ISBN 0-306-42454-1.

Kierulf A., Chen K. and Nievergelt J., "Smart Game Board and Go Explorer: A study in software and knowledge engineering", Communications of the ACM, Vol. 33, No. 2, February 1990, pp. 152-166.

Kirs P.J., Sanders G.L., Cerveny R.P., and Robey D., "An Experimental Validation of the Gorry and Scott Morton Framework", MIS Quarterly, June, 1989, Vol. 13, No. 2, pp. 183-197.

Klinker G., Bentolila J., Genetet S., and Grimes M., "KNACK - Report driven knowledge acquisition", International Journal of Man-Machine Studies, Volume 26, No. 1, January 1987, pp. 65 - 79.

Konsynski B.R., "Special Issue: Decision Support and Knowledge-Based Systems", Journal of Management Information Systems, Vol. 5, No. 1, Summer 1988, pp. 3-5.

Kunz, J.C., Stelzner M.J., and Williams M.D., "From classic expert systems to models", in "Topics in Expert System Design : Methodologies and Tools", by G. Guida and C. Tasso (Editors), Elsevier Science Publishers, North Holland, Amsterdam, 1989, ISBN 0-444-87312-X.

Lamberti D.M. and Wallace W.A., "Intelligent Interface Design: An Empirical Assessment of Knowledge Presentation in Expert Systems", MIS Quarterly, Vol. 14, No. 3, September 1990, pp. 279-311.

Lane E.F. & Verdini W.A., "A Consistency Test for AHP Decision Makers", Decision Sciences, Vol. 20, 1989, pp.575-590.

Laurent J., Ayel J., Thome F., and Ziebelin D., "Comparison evaluation of three expert system development tools: KEE, Knowledge Craft, and ART", The Knowledge Engineer Review, Vol. 1, No. 4, 1986, pp. 18 - 29.

Le Blanc L.A. & Kozar K.A., "An Empirical Investigation of the Relationship Between DSS Usage and System Performance: A Case Study of a Navigation Support System", MIS Quarterly, Vol. 14, No. 3, September 1990, pp. 263-277.

Lenat D.B., "AM: An AI approach to discovery in mathematics", in Davis R. and Lenat D.B. (eds.), "Knowledge-Based Systems in Artificial Intelligence", McGraw-Hill, New York, 1980.

Lehner P.E., "Towards an Empirical Approach to Evaluating the Knowledge Base of an Expert System", IEEE Transactions on Systems, Management and Cybernetics, Vol. 19, No. 3, May/June 1989.

Lehner P.E. and Adelman L., "Perspectives in Knowledge Engineering", IEEE Transactions Systems, Man and Cybernetics Society, 1989.

Leonard-Barton D., "The case of integrative innovation: An expert system at Digital", Sloan Management Review, Vol. 29, No. 1, Fall 1987, pp. 7-19.

Levi K., "Expert Systems Should be More Accurate than Human Experts : Evaluation Procedures from Human Judgment and Decision Making", IEEE Transactions on Systems, Management and Cybernetics, Vol. 19, No. 3, May/June 1989.

Liebowitz J., "Useful approach for evaluating expert systems", EXPERT SYSTEMS, April, Vol. 3, No. 2, 1986.

Liebowitz J., "When is a Prototype an Expert System?", Expert Systems, Spring 1991, pp. 17-22.

Liebowitz J. and De Salvo D.A., "Structuring Expert Systems : Domain Design and Development", Yourdon Press, Prentice Hall, Englewood Cliffs, New Jersey, 1989, ISBN 0-13-853441-1.

Linster M. and Musen M.A., "Use of KADS to create a conceptual model of the ONCOCIN task", Knowledge Acquisition, No. 4, 1992, pp. 55-87.

Liskov B. and Guttag J., "Abstraction and Specification in Program Development", MIT Press, 1986.

Lo L. and Jeffrey R., "An Initial View on Size Estimation for Expert Systems Applications", Proceedings of the 4th Australian Joint Conference on Artificial Intelligence, Perth, 21-23 November 1990, pp. 355-368.

Lock Lee L.G., "A Preliminary Methodology - Systems Analysis for Knowledge Based Systems in Industry", B.H.P. Central Research Laboratories, Shortland NSW, 1986.

Lock Lee L.G., McNamara A.R., Teh K.C., Lie H.M., Orenstein B.J., and Brown D.J.H., "Rapid Prototyping Tools for Real-Time Expert Systems in the Steel Industry", ISIJ International, Vol. 30, No. 2, 1990, pp. 90-97.

Longworth G., "Designing Systems for Change", NCC, Manchester, 1985.

Loofbourrow T.H., "Structured Knowledge Engineering", Expert Systems, Winter 1991, pp. 47-53.

Lundeberg M., Goldkuhl G. and Nilsson A., "Information Systems Development: A Systematic Approach", Prentice Hall Inc., Englewood Cliffs, N.J., 1981, ISBN 0-13-464677-0.

Lyytinen K., "A Taxonomic Perspective of Information Systems Development : Theoretical Constructs and Recommendations", 1987.

Maddison R.N. (Ed.), "Information Systems Methodologies", Wiley Heyden, 1983.

Marcus S., "Taking backtracking with a grain of SALT", International Journal of Man-Machine Studies, Vol. 26, No. 4, April, 1987, pp. 383 - 398.

Marcus S., "Automating Knowledge Acquisition for Expert Systems", Kluwer Academic Publishers, Norwell, Massachusetts, 1988, ISBN 0-89838-286-6.

Markov A.A., "A Theory of Algorithms", National Academy of Sciences, USSR (in English), 1954.

Mathiassen L., "Systemudvikling og Systemudviklings-Metode", DUE-Rapport Nr, 5, 2, Oplay, Mathematisk Insitut, Aarhus Universitet, April, 1982.

Matsumoto S., "ES/SDEM: Software Development Engineering Methodology for Expert Systems", in "Future Generation Computer Systems", Elsevier Science Publishers, North Holland, 1989, pp. 33-39.

McCallum J.C., "Maintenance of expert systems : Life-cycle validity", SIGART Newsletter, Number 94, October 1985.

McCracken D.D. and Jackson M.A., "life Cycle Concept Considered harmful", ACM SIGSOFT, Software Engineering Notes, Vol. 17, No. 2, April 1982.

McCulloch W.S. and Pitts W., "A Logical Calculus of the Ideas Immanent in Nervous Activity", Bulletin of Mathematical Biophysics, Vol. 5, 1943, pp. 115-137.

McDermott J., "R1: an expert in the computer system domain", Proceedings of the National Conference on Artificial Intelligence, 1980, pp. 269-271.

McDonald I.G., "Automating Information Engineering", unpublished paper, JMA, London, 1987.

McGraw K.L. and Seale M.R., "Multiple Expert Knowledge Acquisition Methodology, MEKAM", Texas Instruments Incorporated, 1986.

McGraw K.L. and Harbison K., "Knowledge Acquisition : Principles and Guidelines", pp. 12 - 13, Prentice Hall, Englewood Cliffs, New Jersey, 1989, ISBN 0-13-516436-2.

McGregor J.D. and Korson T., "Object-Oriented Design", Communications of the ACM, Vol. 33, No. 9, September 1990, pp. 39 - 60.

Macarthy J., "Recursive functions of symbolic expressions and their computation by machine", Communications of the ACM, Vol. 4, No. 4, April, 1960, pp. 184-195.

MacDonald I.G., "Information Engineering: An Improved, Automatable Methodology for the Design of Data Sharing Systems", in Olle et al. (Editors), "Information Systems Design Methodologies", Elsevier Science Publishers B.V., North-Holland, 1986.

McNeile A.T., "Jackson System Development", in Olle T.W., Sol H.G. and Verrijn-Stuart A.A. (eds), "Information System Design Methodologies: Improving the Practice, Proceedings of the IFIP WG8.1 Working Conference, Noordwijkerhout, The Netherlands", North Holland, Amsterdam, 1991.

Messier W.F. and Hansen J.V., "Expert Systems in Accounting and Auditing: A Framework and Review", Proceedings of the University of Oklahoma Behavioural Research, Conference, August 1983.

Meyer B., "Bidding farewell to globals", Journal of Object-Oriented Programming, Vol. 1, No. 3, 1988, pp. 73 - 76.

Meyer B., "From structured programming to object-oriented design: the road to Eiffel", Structured Programming, Vol.1, 1989, pp. 19 - 39.

Meyer B., "Programming as contracting", Technical Note, TR-EI-12/CO, Version 2, Interactive Software Engineering, California, 1989, p. 46.

Meyer B., "The new culture of software development: Reflections on the practice of object-oriented design", Proceedings of the First International Conference on the Technology of Object-Oriented Languages and Systems, 1989, pp. 13 - 23.

Meyer B., "Tools For The New Culture: Lessons From The Design of The EIFFEL Libraries", Communications of the ACM, Vol. 33, No. 9, September 1990, pp. 68 - 88.

Michie D., "Expert Systems: The End of the Beginning", AI & Society, Springer-Verlag London Ltd., 1991, pp. 142-147.

Minsky M., "A framework for representing knowledge", in Winston P.H. (ed.), "The Psychology of Computer Vision", McGraw-Hill, New York, 1975.

Montazemi A.R. and Conrath D.W. (1986): "The Use of Cognitive Mapping for Information Requirements Analysis:, MIS Quarterly, Vol 10, No. 1, March 1986, pp. 45-56.

Moses J, "A MACSYMA Primer", Mathlab Memo No. 2, Computer Science Laboratory, Massachusetts Institute of Technology, 1975.

Motta E., Rajan T. and Eisenstadt M., "A Methodology and Tool for Knowledge Acquisition in Keats-2", in Guida P. and Tasso G. (eds.), "Topics in Design of Expert Systems", North Holland, Amsterdam, pp. 265-296.

Mullin T.M., "Experts' Estimation of Uncertain Quantities and its Implications for Knowledge Acquisition", IEEE Transactions on Systems, Management and Cybernetics, Vol. 19, No. 3, May/June 1989, pp. 616-625.

Mumford E., "Managerial Expert Systems and Organisational Change : some Critical Issues", in Critical Issues in IS Research, edited by Boland R.J. and Hirschheim R.A., 1986.

Munro M.C., "An Opinion ... Comment of Critical Success Factors Work", MIS Quarterly, September 1983.

Mykytyn K, Mykytyn P.P. and Slinkman C.W., "Expert Systems: A Question of Liability?", MIS Quarterly, Vol. 14, No. 1, March 1990, pp. 27-42.

Naumann J.D., Davis G.B. and McKeen J.D., "Determining Information Requirements: A Contingency Method for Selection of a Requirements Assurance Strategy", Journal of Systems and Software, Vol. 1, 1980, p. 277.

Neale I., Review of "Analysis for knowledge-based systems: a practical guide to the KADS methodology", by Hickman F.R., Killin J.L., Land L., Mulhall T., Porter D. and Taylor R.M., Ellis Horwood, Chicester, UK, 1989.

Newell A. and Simon H.A., "Human Problem Solving", Prentice Hall, New Jersey, 1972.

Newell A., Shaw J.C. and Simon H.A., "A variety of intelligent learning in a general problem solver", in Yovits M.C. and Camerson S. (eds.), "Self-organising systems, Pergamon Press, New York, 1960.

Noble D.F., "Schema-Based Knowledge Elicitation for Planning and Situation Assessment Aids", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 3, May/June 1989, pp. 473-482.

Nunamaker et al., 1991; J.F., Dennis A.R., Valacich J.S., Vogel D.R., George J.F., "Electronic Meetings Systems to Support Group Work", Communications of the ACM, July 1991, Vol. 34, No. 7, pp. 40-61.

Offutt Daniel, contributing author, University of Michigan, in "Automating Knowledge Acquisition for Expert Systems", by Marcus S., Kluwer Academic Publishers, Norwell, Massachusetts, 1988, ISBN 0-89838-286-6.

O'Keefe R.M., Balci O., and Smith P.E., "Validating Expert System Performance", IEEE Expert, Vol. 2, No.4, 1987, pp. 81 - 89.

O'Leary T.J., Goul M., Moffitt K.E., and Radwan A.E, "Validating Expert Systems", IEEE Expert, June 1990.

O'Leary D.E. "Validation of Business Expert Systems", Audit Judgement Symposium, 1986.

O'Leary D.E., "Validation of Expert Systems - With Applications to Auditing and Accounting Expert Systems", Decision Sciences, Vol. 18, No.3, 1987, pp. 468 - 486.

Olle T.W., Sol H.G. and Verrijn-Stuart A.A. (eds), "Information System Design Methodologies: Improving the Practice, Proceedings of the IFIP WG8.1 Working Conference, Noordwijkerhout, The Netherlands", North Holland, Amsterdam, 1991.

Parsaye K. and Chignell M., "Expert Systems for Experts", John Wiley & Sons, 1988, ISBN 0 471-60721-5.

Passero L. and Kanui R., "Un Intepreter du Language de Programation PROLOG", Université Aix Marseilles, 1972.

Perry W.E., "A Structured Approach to Systems Testing", The QED Series, Prentice-Hall, 1983.

Perry W.E., "Effective Methods of EDP Quality Assurance", The QED Series, Prentice-Hall, 1983.

Pigford D.V. and Baur G., "Expert Systems for Business: Concepts and Applications", Boyd & Fraser Publishing Group, Boston Massachusetts, 1990, ISBN 0-87835-439-5.

Popolizio J.J. and Cappelli W.S. (1989): "New Shells for Old Iron", Datamation, April 15, 1989, pp. 41-48.

Pracht W.E. & Courtney J.F., "The Effects of an Interactive Graphics-Based DSS to Support Problem Structuring", Decision Sciences, 1988, Vol. 19, No. 3, pp. 598-621.

Prerau D.S., Gunderson A.S., Reinke R.E., Adler M.R., "Maintainability Techniques in Developing Large Expert Systems", IEEE Expert, June 1990, pp. 71-80.

Quillian M.R. "Semantic memory", in Minsky M. (ed.), "Semantic Information Processing", MIT Press, Cambridge, Massachusetts, 1968, pp. 227-270.

Remus W., "Semi-Structured Recurring Decisions: An Experimental Study of Decision Making Models and Some Suggestions for DSS", MIS Quarterly, June 1987, Vol. 11, No. 2, pp. 233-243.

Revski G. (1983): "A Recursive Approach to the Comparison of Systems and Software Design Methodologies", Research Report of the Kingston Polytechnic School of Electronic Engineering and Computer Science, April.

Robinson J.A., "A machine oriented logic based on the resolution principle", Journal of the Association for Computing Machinery, Vol. 12, 1965, pp. 23-41.

Rockhart J., "Chief Executive Define Their Own Data Needs", Harvard Business Review, March-April 1979.

Rook F.W. and Croghan J.W., "The Knowledge Acquisition Activity Matrix : A Systems Engineering Conceptual Framework", IEEE Transactions on Systems, Man. and Cybernetics, Vol. 19, No. 3, May/June 1989.

Roussel P., "Prolog Manuel de référence et d'utilisation", Groupe d'Intelligence Artificielle, Marseille-Luming, 1975.

Rumbaugh J., Blaha M., Premerlani W., Eddy F., and Lorensen W., "Object-Oriented Modelling and Design", Prentice Hall, Englewood Cliffs, New Jersey, 1991, ISBN 0-13-629841-9.

Saaty T.L., "Decision Making for Leaders: The Analytical Hierarchy Process for Decisions in a Complex World", Lifetime Learning Publications, California, 1982, ISBN 0-543-97959-9, Chapters 1-5, pp. 1-92.

Sabherwahl R. & Grover V., "Computer Support for Strategic Decision-Making Processes: Review and Analysis", Decision Sciences, 1989, Vol. 20, pp. 54-76.

Sacerdoti E.D., "Managing System", AI Expert, May 1991, pp. 26-33.

Sage A.P., "A case for a standard for systems engineering methodology", IEEE transactions on Systems, Management and Cybernetics, Vol. SMC-7, No. 7, pp. 156 - 160, July 1977.

Shortliffe E.H., "Computer-Based Medical Consultations: MYCIN", Elsevier, New York, 1976.

Shlaer S. and Mellor S.J., "Object-Oriented Systems Analysis: Modelling the World in Data", Yourdon Press Computing Series, 1988, p. 144.

Shpilberg D., Graham L.E. and Schatz H. (1986): "ExperTAX^{sm}: An Expert System for Corporate Tax Planning", Expert Systems, July, Vol. 3, No. 3, pp. 136-151.

Silver M.S., "Decision Support Systems: Directed and Non-directed Change", Information Systems Research, Vol. 1, No. 1, March 1990, pp. 47-70.

Slagle J.R., Gardiner D.A., and Han K., "Knowledge Specification of an Expert System", IEEE Expert, August 1990.

Smith G.F., "Defining Managerial Problems - A Framework for Prescriptive Theorising", Management Science, 1989, Vol. 35, No. 8, pp. 963-981.

Smith D.L., Nauss R.M. & Bird D.A., "Decision Support for Bus Operations in a Mass Transit System", Decision Sciences, Vol. 21, 1990, pp. 183-203.

Sol H.G., "A feature analysis of information systems design methodologies: methodological considerations", in Olle T.W., Sol H.G. and Tully C.J. (eds.), "Information System Design Methodologies: A Feature Analysis", North Holland, Amsterdam, 1983.

Sowa J.F., "Conceptual Structures: Information Processing in Mind and Machine", Addison-Wesley, Reading, Massachusetts, 1984.

Sowa J.F., "Conceptual Graph Notation", Proceedings of Fourth Annual Workshop on Conceptual Graphs, Detroit, Michigan, 1989.

Sviokla J.J., "An Examination of the Impact of Expert Systems on the Firm: The Case of XCON", MIS Quarterly, Vol. 14, No. 2, June 1990, pp. 127-140.

Takac P. and Lerner J., "The Annual Review of Expert Systems Activities in Australia", RMIT Report, as reported in Cranswick K., "Just yesterday's buzzword?", Computing, June 1990, p. 21.

Tasker S., Guest lecture "Management Issues with Knowledge Based Systems", Knowledge Based Information Systems Course, Master of Commerce Program, University of New South Wales, September 1992.

Taylor R.M., Porter D., Hickamn F., Steng K.-H., Tansley S., and Dorbes G., "System evolution - principles and methods (the life cycle model)", ESPRIT Project P1098, Deliverable Task G9, Touche Ross, 1989.

The Spang Robinson Report, "The Spang Robinson Report on Artificial Intelligence", Vol. 3, No. 14, December 1987.

The Spang Robinson Report on Artificial Intelligence, July 1991, Vol. 7, No. 7, John Wiley and Sons, Inc., New York.

The Spang Robinson Report on Artificial Intelligence, December 1991-January 1991, Vol. 8, No. 1, John Wiley and Sons, Inc., New York.

The Oxford English Dictionary, 2nd Edition, Clarendon Press, Oxford, 1989.

Tolcott M.A., Marvin F.F., and Lehner P.E., "Expert Decision Making in Evolving Situations", IEEE Transactions on Systems, Management and Cybernetics, Vol. 19, No. 3, May/June 1989.

Turban E., "Expert Systems and Applied Artificial Intelligence", Macmillan Publishing Company, New York, 1992, ISBN 0-02-421665-8.

Turban E., "Decision Support and Expert Systems: Management Support Systems", 3rd edition, Macmillan Publishing Company, New York, 1993, ISBN 0-02-421691-7.

Turner J. A., "Understanding the elements of system design", in Critical Issues in Information Systems Research, Boland Jnr. R.J. and Hirschheim R.A. (Editors), John Wiley, Chichester, 1987, pp. 97 - 111.

Van Melle W., "A Domain-independent Production Rule System for Consultation Programs", International Journal for Computing and Artificial Intelligence, 1979, pp. 923-925.

Veryard R., "What are methodologies good for?", Data Processing, Vol. 27, No. 6, July/August, 1985.

Waterman D.A., "A Guide to Expert Systems", Addison-Wesley Publishing Company, Reading Massachusetts, 1986, ISBN 0-201-08313-2.

Walters J.R. and Nielsen N.R., "Crafting Knowledge-Based Systems: Expert Systems Made Realistic", John Wiley & Sons, Inc., New York, 1988, ISBN 0-471-62479-9.

Watson H.J., Rainer R.K. Jr., and Koh C.E., "Executive Information Systems: A Framework for Development and a Survey of Current Practices", MIS Quarterly, Vol. 15, No. 1, March 1991, pp. 13-30.

Weitzel J.R. and Kerschberg L., "Developing Knowledge-Based Systems: Reorganising the System Development Life Cycle", Communications of the ACM, Vol. 32, No.4, April 1989.

White P., "Towards a repeatable methodology", Perspective, Vol. 1, No. 2, April 1982.

Wielinga B.J., Schreiber A. Th. and Breuker J.A., "KADS: a modelling approach to knowledge engineering", Knowledge Acquisition, No. 4, 1992, pp. 5 - 53.

Wilson E., Edge W., Fogelgarn L., Simnett R. and Gay G., "Going Concern: A Prototype Expert System", Proceedings International Conference on Using Expert Systems by Accountants/Auditors, Amsterdam, The Netherlands, 26-28 September 1988.

Wilson M., Duce D. and Simpson D., "Life cycles in software and knowledge engineering: a comparative review", The Knowledge Engineering Review, No. 4:3, 1989, pp. 189-204.

Wirfs-Brock R. and Johnson R., "Surveying current research in object-oriented design", Communications of the ACM, Vol. 33, No. 9, September 1990.

Wirfs-Brock R.J. and Wilkerson B., "Object-oriented design: a responsibility-driven approach", In Proceedings of OOPSLA '89, 1989, pp. 71 - 75.

Yourdon E. and Constantine L.L., "Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design", Yourdon Press, A Prentice Hall Company, Englewood Cliffs, NJ, 1979, ISBN 0-13-854471-9.

Zadeh L.A., "Fuzzy sets", Information and Control, Vol. 8, 1965, pp.338-353.

Zawa S., "Expert Systems Study", State Bank Report, 1986.

Zawa S., "The Personal Loan Assistant - A prototype expert system", State Bank Report, 1987.

Zawa S., "The Personal Loan Assistant - User Manual", Version 0.2, 1987.

Appendix 1: Examples of Expert System Building Tools

Information obtained from Bowerman and Glover (1988) [pp. 141-163]:

Product	Manufacturer/Distributor	Primary Knowledge Representation/Inference	Other Knowledge Representations	Other Inference Mechanisms
ACORN	Gold Hill Computers	Lattice of frames (inheritance)	IF/THEN, WHEN/DO,IF/DO rules, object oriented programming	Forward, backward, goal-directed forward chaining
Advisor Design Software / Advisor Expert Controller	Ultimate Media Inc.	IF/THEN rules - forward and backward chaining - up to 7 premises per rule	-	-
Aion Development System/PC (ADS/PC)	Aion Corporation	IF/THEN/ELSE/ELSE/IF rule objects (forward and backward chaining)	State object for decomposition during development	-
Aion Development System/MVS (or VM) (ADS/MVS or ADS/VM)	Aion Corporation	IF/THEN/ELSE/ELSE/IF rule objects (forward and backward chaining)	State object for decomposition during development	-
Automated Reasoning Tool (ART)	Inference Corporation	IF/THEN rules (forward backward, and mixed chaining)	Frames, viewpoints, logic and object oriented programming	Inheritance, logical dependencies, truth maintenance, temporal and hypothetical
ENVISAGE	System Designers Software, Inc.	IF/THEN rules (forward and backward chaining)	-	Fuzzy logic, Bayesian inference, demons
ESP ADVISOR	Expert Systems International	IF/THEN-like rules (PROLOG backward chaining)	Knowledge base sections - partitions	Pseudo-forward chaining through sections
ESP Frame-Engine	Expert Systems International	Frames (inheritance)	IF/THEN rules, demons	Forward and backward chaining on rules
Expert-Ease	Jeffrey Perrone & Associates, Inc.	Rules induced from examples	-	-
Expert Edge	Jeffrey Perrone & Associates, Inc.	IF/THEN rules (backward and limited forward chaining)	-	Auto-truth maintenance of knowledge base, Bayesian statistics
Expert System Developing / Consulting Environment (ESDE/ESCE) - VM or MVS	IBM	IF/THEN rules (forward and backward chaining)	Problem decomposition modules	Scaled uncertainty propagation

Product	Manufacturer/Distributor	Primary Knowledge Representation/Inference	Other Knowledge Representations	Other Inference Mechanisms
1st-Class	Programs in Motion, Inc.	Rules entered or induced from examples (forward and backward chaining)	-	Nine statistical calculations, ID3 induction "what if" capabilities
GURU	Micro Data Base Systems, Inc.	IF/THEN rules (forward backward, and mixed chaining)	-	fuzzy logic
INSIGHT2+	Level Five Research	IF/THEN/ELSE production rules (forward and backward chaining)	-	"What if" reporting
Integrated Knowledge Environment (IKE)	LISP Machine, Inc.	IF/THEN rules (forward and backward chaining)	Frame objects, icons	Inheritance
Knowledge Craft	Carnegie Group, Inc.	Frames (inheritance semantics)	IF/THEN rules, logic and object oriented programming agendas, methods	Forward and backward chaining, demons, procedural attachment, "what if", time simulation
Knowledge Engineering Environment (KEE)	Intellicorp	Frames (inheritance and hierarchies)	IF/THEN rules, logic and object oriented programming, agendas, methods	Forward and backward chaining, demons, procedural attachment, "what if", time simulation
Knowledge Engineering System (KES)	Software Architecture and Engineering	Production rules (IF/THEN) (backward chaining with object classes)	"Hypothesise and test" (frame-like) statistical models, demons	Frame network with abduction reasoning, Bayesian statistics, inheritance
Knowledge WorkBench (KWB)	Silogic, Inc.	IF/THEN/ELSE rules in the form of special Prolog classes (backward Prolog chaining)	Entity relationship model	3 types of reasoning plus user-defined, top-down search
LOOPS	Xerox AI Systems	Objects (like Smalltalk) - (message passing)	Access, rule programming, LISP procedures	-
M.1	Teknowledge	IF/THEN rules (forward and backward chaining)	-	-
Personal Consultant Easy	Texas Instruments Incorporated	IF/THEN rules (forward and backward chaining)	-	-
Personal Consultant Plus	Texas Instruments Incorporated	IF/THEN rules (forward and backward chaining)	Frames, LISP functions, access-oriented functions	Inheritance, problem decomposition procedures
Process Intelligent Control (PICON)	LISP Machine, Inc.	IF/THEN rules (forward and backward chaining)	Frame objects, icon hierarchy, other rule types	Inheritance, simulation chaining, simulation
RuleMaster	Radian Corporation	IF/THEN or induced rules (forward and backward chaining)	Hierarchical design of modules	Network transitions

Product	Manufacturer/Distributor	Primary Knowledge Representation/Inference	Other Knowledge Representations	Other Inference Mechanisms
SAGE	Systems Designers Software, Inc.	IF/THEN rules (backward chaining)	-	Fuzzy logic, Bayesian inference
S.1	Teknowledge	Relational frames (class hierarchy and inheritance)	IF/THEN rules, rule categories, procedural attachment	Forward and backward chaining, demons
TIMM	General Research Corporation	IF/THEN rule format, auto-generated from user data (emulated forward and backward chaining)	Frame-based system	Analogical partial match inferencing, demons, procedural attachment
TIMM-PC	General Research Corporation	IF/THEN rule format, auto-generated from user data (emulated forward and backward chaining)	Frame-based system	Analogical partial match inferencing, demons, procedural attachment
XSYS/EXSYS	California Intelligence	IF/THEN/ELSE rules (forward and backward chaining)	-	-

Appendix 2: Knowledge Based Systems in Auditing

SYSTEM NAME	FUNCTION	DESCRIPTION OF DOMAIN	TOOLS USED	STATUS	REFERENCE
TICOM	Audit	Internal control evaluation	Pascal	Prototype	Bailey et al (1984)
ACLS	Audit	Analytical review	AL/X	Prototype	Braun & Chandler (1984)
EDP-XPERT	IS Audit	Auditing advanced EDP systems	AL/X	Prototype	Hansen & Messier (1984, 1986)
AUDITOR	Audit	Audit allowance for bad debts	AL/X	Prototype	Dugan (1985)
IC ANALYZER	Audit	Evaluates data on internal controls extracted form a client's DB in order to decide on the quality of the system of internal control	?	Prototype	Gal (1985)
ICAEW	IS Audit	Data processing controls risk assessment evaluated	Few shells	Development	Edwards (1986)
ICES	Audit	Internal control evaluation	?	Prototype	Grudnitski (1986)
ARISC	Audit	Evaluates a client's system of internal controls through a model of the auditor's decision making process	GALEN	Prototype	Meservy et al., (1986)
AUDITPLANNER	Audit	Considers materiality in audit planning, ensuring audit evidence will be timely, relevant and sufficient	EMYCIN	Research	Steinbart (1987)
CAPS	Audit	Investigates the structure of auditor judgement in the planning process (risk based)	?	Prototype	Boritz (1988)
INTERNAL CONTROL	Audit	Internal control evaluation of accounts receivable	AION ADS/PC	Prototype	Edge & Wilson (1988)
?	Bank Audit	Monitors certain financial transactions to detect fraud	XiPlus	Working	Lecot (1988)
EXPERTEST	Audit	Tailors standard audit programs of substantive tests for individual assignments	Qshell based on Gold Hill Common Lisp	Production	Bickerstaff (1988)
SAM	Audit	Commercial tailoring of standard audit programs	C + own DB	Production	Bickerstaff (1988)
EXPERT AUDITOR OPINION	Audit	Expresses audit opinion via expert interpretation of AUP 3 on the financial statements	VP Expert	Prototype	Holmes (1989)

Appendix 3: KBS Development Questionnaire

Notes:

- Please provide, as accurately as possible, answers for all questions shown in this questionnaire.

- Where an answer requires you to provide a rating, circle the most appropriate number (from the range "1" through to "7"). The ratings should be assessed on the following scale:

1	-	Extremely unimportant Extremely low level of involvement Extremely low level of interest
2	-	Very unimportant Very low level of involvement Very low level of interest
3	-	Unimportant Moderately low level of involvement Moderately low level of interest
4	-	Neither important or unimportant Neither a high or low level of involvement Neither a high or low level of interest
5	-	Important Moderately high level of involvement Moderately high level of interest
6	-	Very important Very high level of involvement Very high level of interest
7	-	Extremely important Extremely high level of involvement Extremely high level of interest

Part I - Organisational Details

1. Organisation name: ______________________________

Name of Respondent	Name of Organisation in which KBS is located

2. Organisation's principal activities are: ______________________________

____

____

3. Contact: Position/title: ______________________________

Name: ______________________________

Telephone no.: (____) ________ - ________

4. How was the KBS group formed? ______________________________

____

____

5. What area within your organisation initiated/championed the use of KBS technology:

- ☐ MIS department
- ☐ Research and development area
- ☐ Operational, production area
- ☐ Head Office initiative
- ☐ Other (specify)

____

6. How difficult was it to introduce KBS technology:
(**Scale:** 1 - Extremely difficult; 7 - Extremely easy)

- Within MIS 1___2___3___4___5___6___7
- Outside of MIS 1___2___3___4___5___6___7

7. Does the KBS group form an identifiable budget unit:

☐ YES ☐ NO (Go to Question 10)

8. Estimate of funding for KBS developments:

Year	Amount of Funding ($ '000)	Compared to IS Percentage of total funding (%)
1992		
1991		
1990		

9. Percentage budget allocations by function:

Type of function	KBS Developments	Compared to other IS Developments
	Percentage of total allocation (%)	
New systems development		
Systems maintenance		
Production/Operations		
Training		
Hardware purchase/maintenance		
Software purchase/maintenance		
Other (specify):____________		
Total	100	100

10. No. of staff involved in KBS development: ______ (persons)

11. Proportion of staff involved in KBS development:

Compared with (which ever is applicable):	1990	1991	1992
Total IS staff			
Research and development staff			
Other (specify):____________________			

12. If the KBS group does not form an identifiable budget unit, then the percentage of MIS, etc. budget allocated to KBS technology (e - estimate; a - actual): (__e__a__) ______(%)

13. If the KBS group does not form an identifiable budget unit, then how are its operations funded?

Beginning of project	Current situation	
☐	☐	MIS budget allocation
☐	☐	R & D budget allocation
☐	☐	One of "special projects" allocation
☐	☐	Separate allocations for each KBS development
☐	☐	Other (specify):

____

____

14. Your rating of the KBS group's effectiveness: 1___2___3___4___5___6___7
(**Scale:** 1 - Extremely ineffective; 7 - Extremely effective)

15. Technical achievement of KBS function: 1___2___3___4___5___6___7
(**Scale:** 1 - Extremely low; 7 - Extremely high)

16. The perception of the KBS group's effectiveness from:
(**Scale:** 1 - Extremely ineffective; 7 - Extremely effective)

- MIS department perspective 1___2___3___4___5___6___7
- Corporate (Head Office) perspective 1___2___3___4___5___6___7
- User area(s) perspective(s) 1___2___3___4___5___6___7

17. Satisfaction with KBS developments to date from:
(**Scale:** 1 - Extremely unsatisfied; 7 - Extremely satisfied)

- MIS department perspective 1___2___3___4___5___6___7
- Corporate (Head Office) perspective 1___2___3___4___5___6___7
- User area(s) perspective(s) 1___2___3___4___5___6___7

18. How important is it during KBS development to:
(**Scale:** 1 - Extremely unimportant; 7 - Extremely important)

- Meet deadlines 1___2___3___4___5___6___7
- Operate within budget 1___2___3___4___5___6___7
- Work to user requirements 1___2___3___4___5___6___7
- Demonstrate cleverness, innovation 1___2___3___4___5___6___7

19. In your organisation, what are the significant milestones or checkpoints that are associated with KBS development?

____
____
____
____
____
____
____
____
____
____
____
____
____
____

20. Consider ***KBS DEVELOPMENT.*** What is the level of:
(**Scale:** 1 - Extremely low; 7 - Extremely high)

- Importance to the organisation 1___2___3___4___5___6___7
- Level of user involvement 1___2___3___4___5___6___7
- Interest of senior management:
  - Within MIS department 1___2___3___4___5___6___7
  - Outside MIS department 1___2___3___4___5___6___7
- Effectiveness of completed systems 1___2___3___4___5___6___7
- Integration with other systems 1___2___3___4___5___6___7

21. Consider ***CONVENTIONAL SYSTEMS DEVELOPMENT.*** What is the level of:
(**Scale:** 1 - Extremely low; 7 - Extremely high)

- Importance to the organisation 1___2___3___4___5___6___7
- Level of user involvement 1___2___3___4___5___6___7
- Interest of senior management:
  - Within MIS department 1___2___3___4___5___6___7
  - Outside MIS department 1___2___3___4___5___6___7
- Effectiveness of completed systems 1___2___3___4___5___6___7
- Integration with other systems 1___2___3___4___5___6___7

Part II - KBS System Details

1. System name: ______________________

2. Project manager: ______________________

3. Contact telephone no.: (____) ____ - ____

4. Description of KBS domain:

____

____

____

____

____

____

____

____

5. Major objective of KBS development:

5.1. Production system:
- ☐ Expert level
- ☐ Colleague level
- ☐ Assistant level

5.2. Demonstration prototype:
- ☐ Expert level
- ☐ Colleague level
- ☐ Assistant level

5.3. Technology demonstration:
- ☐ Expert level
- ☐ Colleague level
- ☐ Assistant level

6. Date development started: ____/____/____

7. Date development completed: ____/____/____

8. For this system, where was the KBS group located in the organisation:

Within MIS:

- ☐ Normal line reporting
- ☐ Created for each individual KBS development project
- ☐ Part of "special projects"
- ☐ Other (specify) ____________________

Not within MIS:

- ☐ Specify division/department/section name ____________________

____

9. Type of KBS:

- ☐ Stand alone, consultation type
- ☐ Integrated system, consultation type
- ☐ Embedded system
- ☐ Other (specify)____________________

____

10. No. of users:

Type of user	Expected users at start of project	Currently
Direct users of the system		
Indirect users of the system		

11. Hardware development platform:

		Demonstration prototype(s)	Production system
Mainframe	- centralised	☐	☐
	- distributed	☐	☐
Minicomputer		☐	☐
PC		☐	☐
AI workstation		☐	☐

12. KBS development tools:

Demonstration prototype(s):

☐ AI language (specify): ______________________

☐ Expert system shell (specify): ______________________

☐ 3GL or 4GL (specify): ______________________

☐ Hybrid system:

- AI language (specify): ______________________
- ES shell (specify): ______________________
- 3GL or 4GL (specify): ______________________

Production system:

☐ AI language (specify): ______________________

☐ Expert system shell (specify): ______________________

☐ 3GL or 4GL (specify): ______________________

☐ Hybrid system:

- AI language (specify): ______________________
- ES shell (specify): ______________________
- 3GL or 4GL (specify): ______________________

13. Development of KBS:

☐ In house

☐ Using external consultants

☐ An out-sourced activity

☐ Other (specify) ______________________

____

14. Knowledge representation used:

- ☐ Production rules
- ☐ Frames
- ☐ Combination of frames and production rules
- ☐ Semantic networks
- ☐ Other (specify)____

____

15. Inference engine uses:

- ☐ Hypothesise and test
- ☐ Forward chaining only
- ☐ Backward chaining only
- ☐ Both forward and backward chaining
- ☐ Inheritance
- ☐ Other (specify)____

____

16. User interface operates as (you can tick more than one box):

- ☐ No user interface, e.g. embedded system
- ☐ Command or menu driven
- ☐ Rule trace only
- ☐ Has context sensitive HELP
- ☐ Has context sensitive EXPLANATION FACILITY
- ☐ EXPLANATION FACILITY focuses on how conclusions, recommendations, etc. were reached
- ☐ A graphical/icon interface

17. Who owns the KBS? ____

18. Funding for KBS development came from:

- ☐ User area
- ☐ One-off allocation from MIS budget
- ☐ From KBS budget
- ☐ Other (specify)________________________________

19. Composition of KBS development team for this system:

(Provide best estimate for system)

Category	No. of staff	Resource contribution (person-months)
Expert(s)		
Users		
Knowledge engineers		
Programmers		
Systems analysts		
Specialists, e.g. knowledge acquisition		
Business analysts		
Other		
Total		

20. Qualifications of KBS development team:

Qualifications	Experience		Predominant area of KBS expertise			No. of staff
	Years in IS	Years in KBS	AI language	ES shell	Other, e.g. knowledge acquisition	

21. ***DOCUMENTATION*** generated and used at the following milestones during the development of KBS:

- Identification/selection of suitable area.

- Knowledge acquisition

- Knowledge representation

- System design

- Verification and validation

- Implementation

22. Can you describe, in diagrammatic form, the "organisational line of control" applied to the development of this system?

23. Criteria used in selecting KBS development software were (you can tick more than one box - rank of "1" would be most important, with higher ranks of lesser importance):

Rank

- ☐ ___ No selection, software used in previous KBS developments
- ☐ ___ KBS software evaluations
- ☐ ___ Good reviews in AI magazines and journals
- ☐ ___ Recommended by KBS staff
- ☐ ___ Cost of software
- ☐ ___ Suitable to existing hardware
- ☐ ___ Recommended by another organisation
- ☐ ___ Other (specify)______________________________________

____

24. Criteria used in the hardware selection were (you can tick more than one box - rank of "1" would be most important, with higher ranks of lesser importance):

Rank

- ☐ ___ No selection, hardware used in previous KBS developments
- ☐ ___ No selection, KBS staff required to use currently available hardware
- ☐ ___ Response time critical - hardware selected mainly on this criterion
- ☐ ___ Embedded KBS, and so used hardware of the application system in which the KBS is embedded
- ☐ ___ Cost of hardware
- ☐ ___ Formal assessment of suitable hardware by KBS staff
- ☐ ___ Other (specify)______________________________________

____

Part III - KBS Development Methodology

1. Do you think that KBS development should follow a methodological approach?

☐ YES ☐ NO

2. What are your principal reasons for thinking that KBS development should follow a methodological approach?

____

____

____

____

3. What are your principal reasons for thinking that KBS development should not follow a methodological approach?

____

____

____

____

4. Indicate your level of familiarity with the following system development methodologies:
(**Scale:** 1 - no knowledge; 7 - extemely familiar)

Knowledge Acquisition Activity Matrix [Rook & Croghan]	1__2__3__4__5__6__7
KADS [Touche Ross]	1__2__3__4__5__6__7
ES/SDEM [Fujitsu]	1__2__3__4__5__6__7
NesDEM [Ching & Jamieson, 1987]	1__2__3__4__5__6__7
Turban methodology [1990]	1__2__3__4__5__6__7
Weitzel and Kerschberg	1__2__3__4__5__6__7

5. Were the normal IS system development standards and procedures applicable to the development of this system?

☐ YES ☐ NO

6. No. of times systems development methodology used: ____

7. How was the methodology developed?

8. What approval process was applied to certify the methodology?

9. Was the methodology "adjusted" or changed as new systems were been developed?

☐ YES ☐ NO

10. Did the methodology exist before development of KBS's was initiated?

☐ YES ☐ NO

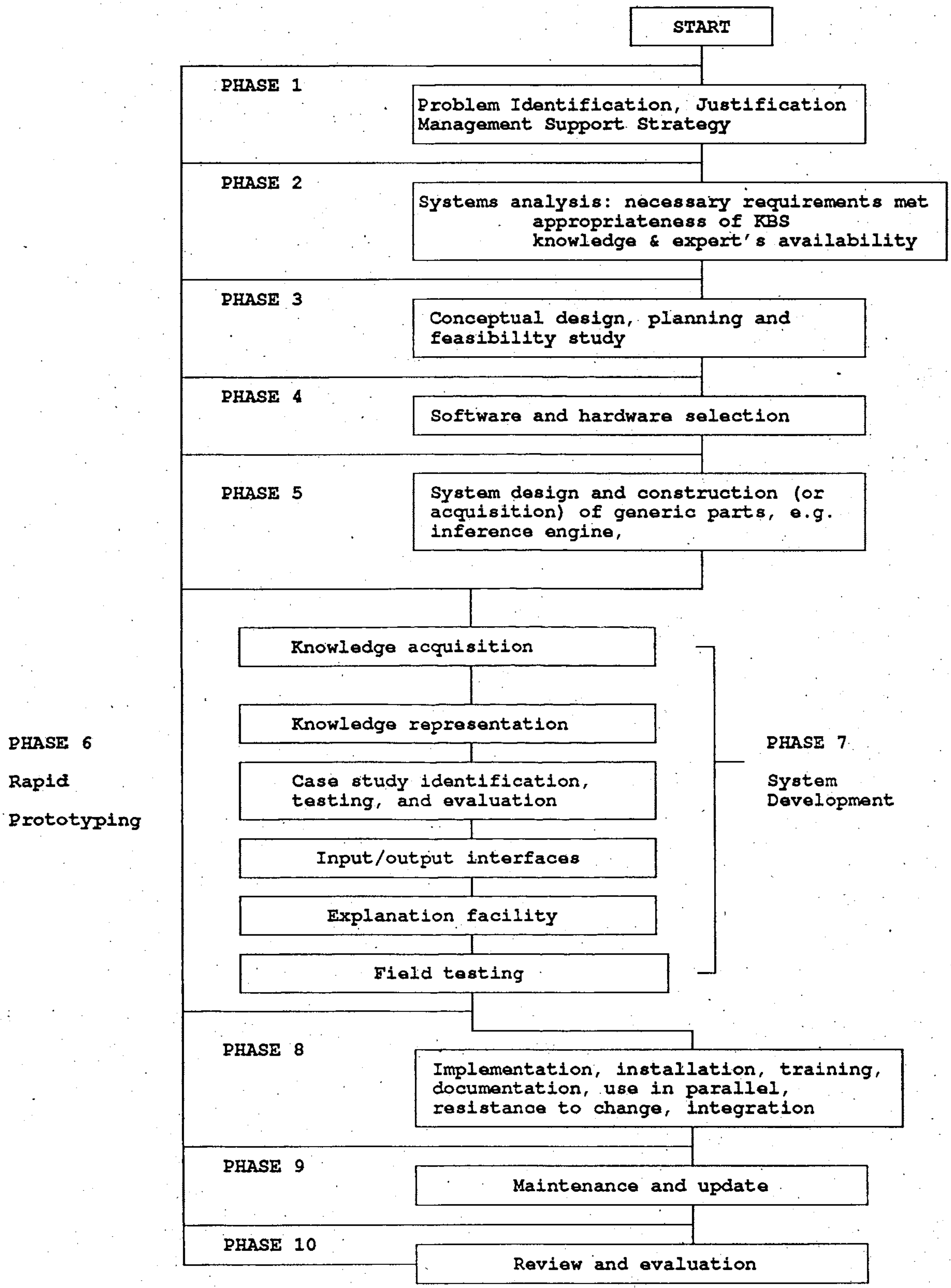
11. Was the methodology reviewed at the end of developing this KBS?

☐ YES ☐ NO

12. Does the methodology have acceptance throughout the organisation?

☐ YES ☐ NO

13. The diagram below presents a popular "textbook" system development methodology applicable to KBS.



14. Degree to which model reflects KBS development in practice:

1___2___3___4___5___6___7

(**Scale:** 1 - Extremely low; 7 - Extremely high)

15. What good features does the above KBS development methodology possess?

____

____

____

____

____

____

____

16. What are the major defects with respect to the above model (you can tick more than one box):

Rank

- ☐ ___ No defects, the model is a good representation
- ☐ ___ Does not allow for contingencies such as staff relocation, etc.
- ☐ ___ Too great an emphasis on the initial phases, whereas rapid prototyping is most important
- ☐ ___ Model too sequential and most phases overlap, or there is often return to earlier phases during development
- ☐ ___ No indication within the methodology of user involvement
- ☐ ___ More suited to systems developed in an AI language than those developed using an ES shell
- ☐ ___ Other (specify)________________________________

___ ________________________________

___ ________________________________

17. Do you consider that your organisation has a formal KBS development methodology?

☐ YES ☐ NO

18. Does documentation exist that outlines the development methodology?

☐ YES ☐ NO

19. Major reasons for not adopting a formal KBS development methodology are:

- ☐ Access to important personnel, such as the expert means that a more ad hoc approach is often the only viable way of proceeding
- ☐ Different problem domains mean that a single methodology is inappropriate
- ☐ Other (specify)____

____

20. Reporting procedures used during KBS development are: Frequency

- ☐ Regular reports to MIS ______
- ☐ Verbal reports to interested parties ______
- ☐ Formal reports to system development steering committee ______
- ☐ Budget and staffing reviews ______
- ☐ Other (specify)________________________________ ______

____ ______

21. Contribution of KBSDM to system success was: 1___2___3___4___5___6___7
(**Scale:** 1 - Very low; 7 - Very high)

22. Was training in the KBSDM provided?:

- ☐ YES. To who: ____

____

- ☐ NO

As this completes the questionnaire,
I would like to thank you for your enthusiastic participation