KNOWLEDGE BASED SYSTEMS :
A POWERFUL YET DECEPTIVE ENGINEERING TOOL

A.R. Greef and R. Reinecke
Department of Industrial Engineering
University of Stellenbosch
Stellenbosch, 7600
South Africa

ABSTRACT

Knowledge Based System (KBS) technology emanating from a sub-field of Artificial Intelligence can be a powerful and yet deceptive engineering tool. It's power is evident from the magnitude of published KBS applications. It's deceit lies in the illusion that the technology is universally powerful and relatively simple to implement. This paper attempts to aid the engineer with the handling and application of this new tool: by firstly presenting an overview of the historical evolution of KBS's and describing their organization, and secondly, by describing a number of successful engineering applications of KBS's. Common deceptive pitfalls and simplifications are exposed at each step.

1. INTRODUCTION

The commercial utilization of technology emanating from the field of Artificial Intelligence (AI) has produced both positive and negative results. Positive, in that useful computer based methodologies have been lifted out of pure Computer Science and made to work in the engineering environment. Negative, in that the associated generalized terminology (intelligence, knowledge) and concepts (Knowledge Systems, Expert Systems) have been manipulated to serve sales rather than implementation. The most recent sub-field of AI affected in this manner is that from which computer programs, generically known as "Knowledge Based Systems", have evolved. The impression is gained that an engineer armed with a few select terms such as "Expert System" (ES) and "Knowledge", and sporting a PC based ES Shell can set about

solving all manner of engineering problems. Attempting to solve a problem of the slightest complexity will, however, soon highlight the error of this approach.

The world wide attention which Knowledge Based Systems (KBS's) have received and the magnitude of contemporary published applications is conclusive evidence of the power of this technology as an engineering tool. However, the tool is deceptive in that it lures the unsuspecting into generalizing terms and overrating capabilities thereby giving the impression that the technology is universally powerful and relatively simple to implement.

This paper attempts to aid the engineer with the handling and application of this new tool: by firstly presenting an overview of the historical evolution of KBS's and describing their organization, and secondly, by describing a number of successful engineering applications of KBS's. Common deceptive pitfalls and simplifications are exposed at each stage of explanation.

2. THE EVOLUTION OF EXPERT SYSTEMS

In the 1960's, research in the Problem Solving sector of the AI field progressed at an explosive rate as researchers attempted to emulate the hypothesized method which humans employ to visualize and solve problems. During this period the term Artificial Intelligence came to be associated with the manner in which a computer could be programmed to mimic human intelligence. Out of this period emerged a host of problem representation and problem solving techniques [1] which could be employed to play games, solve puzzles and identify similarities in picture problems found in IQ tests.

These techniques represented games and problems as a tree (or graph) of choices. A problem's solution could then be found by traversing the tree in some manner and tracing out the path from the initial conditions to the goal conditions. The success of this methodology led researchers to believe that the only limitation to computer intelligence was the size of available computer memory and the speed at which computers could operate.

Any attempts to solve larger problems in this manner, however, were plagued by combinatorial explosion. The number of choices which the computer had to consider during the problem's solution became exceptionally large leading to unacceptably long computer operating times. In an attempt to reduce computer time, a number of heuristic search techniques (not to be confused with heuristics embedded in knowledge) were developed (detailed in Winston [2]) to force the tree search to follow more direct paths. These search aids were usually in the form of a mathematical evaluation function which exploited some aspect of a problem's solution. This approach to increasing the speed at which problems could be solved,

however, was terminated by the inability to find a generalized evaluation function for application to all problems.

This led Marvin Minsky [3] to consider, instead of representing the physical attributes of problem in a tree structure, representing knowledge which humans have already formulated during the solving of a particular problem in a similar tree structure called a semantic net. The speed of problem solving was increased by using human-derived heuristic knowledge (rules of thumb) to guide the search process. Minsky's success forced research away from finding efficient search techniques and into finding methods of representing human knowledge. Other knowledge representation formalisms derived, which merely served to chunk (or group) knowledge attributes, were "Frames" and "IF-THEN rules". Problem solving developed into a method of utilizing a generalized search technique to traverse a tree structure representing human knowledge concerning a particular problem's solution. These problem solvers were called Knowledge Based Systems as they employed human knowledge to solve problems that ordinarily required human intelligence.

KBS's became so specialized that it was convenient to define them separately from the Problem Solving sub-field of AI. The term "Expert System" was later coined when Knowledge Based Systems first started performing equal to or better than human experts. This should be the yardstick for determining whether a KBS qualifies as an Expert System. Nowadays, with the proliferation of applications of this technology, a number of useful KBS's have evolved which do not mimic humans. To differentiate these systems from ES's, they are called Knowledge Systems [4], as although they employ human knowledge they do not perform at the level of a human expert.

Today, using the term Expert System for what is actually a Knowledge System deceives the engineer into expecting more than what is attainable from an application. It is the quality of knowledge which provides the key to expert performance and not the representation technique used, or the search method employed, both of which only provide the mechanism for its use. Using KBS technology does not automatically produce Expert Systems !

3. ORGANIZATION OF A KNOWLEDGE BASED SYSTEM

3.1 Components of a Knowledge Based System

A KBS is organized into a number of discrete system components. Figure 1(a) shows a common schematic diagram of a simple KBS organization. Figure 1(b) presents a slightly more detailed organization diagram of a small KBS used for solving problems having a relatively small search space (few rules) comprising unreliable (probabilistic) and static (time-independant) knowledge and data. Figure 1(c)

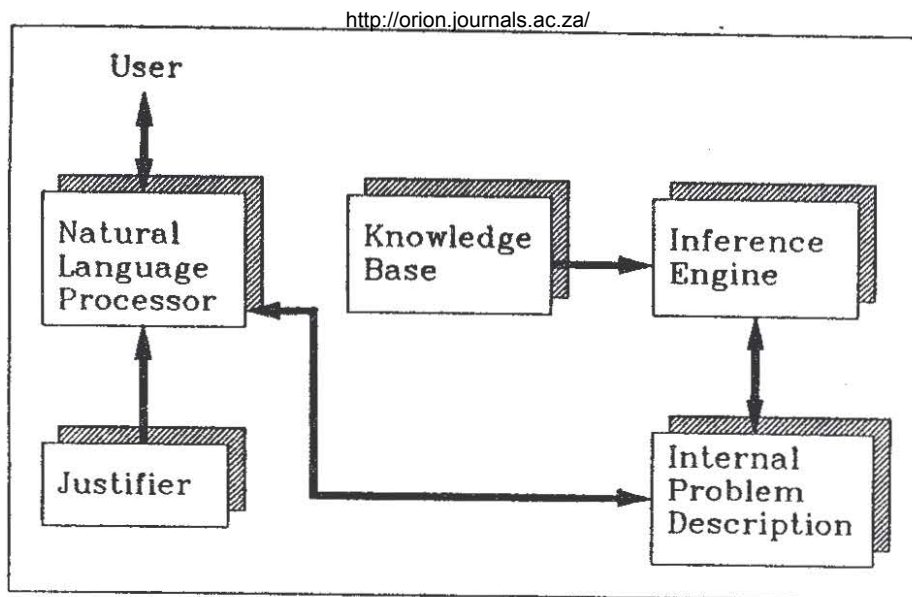indicates the organizational complexity required for large KBS's containing non-trivial knowledge and data.

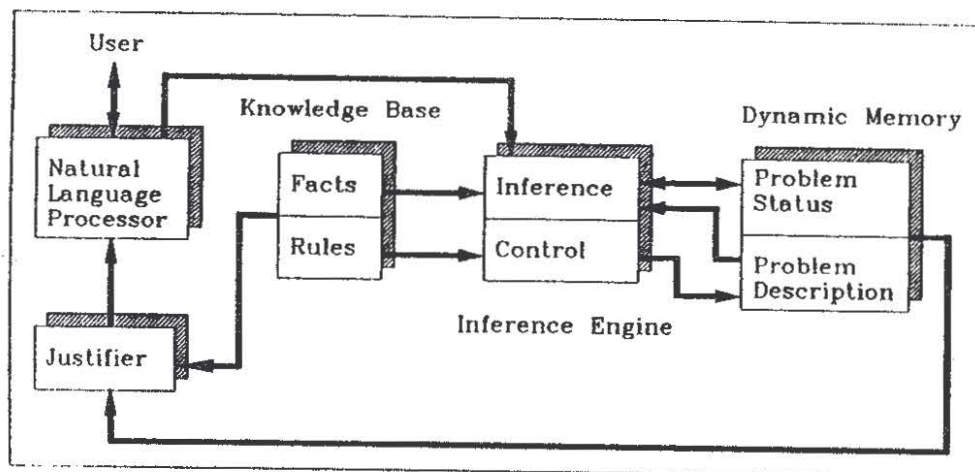FIGURE 1(a): SCHEMATIC DIAGRAM OF A SIMPLE KBS

FIGURE 1(b): A SIMPLE KBS'S COMPONENTS

3.1.1 The Natural Language Processor

The Natural Language Processor is the channel through which the KBS and user communicate. The user enters problem details (when prompted by the system) or asks questions regarding problem solution. The details are passed on to the Inference Engine and the questions are passed on to the Justifier (Explanation Facility). The Natural Language Processor hides the user from the internal data structures of the knowledge which are often dissimilar to one of the knowledge representation formalisms given below.
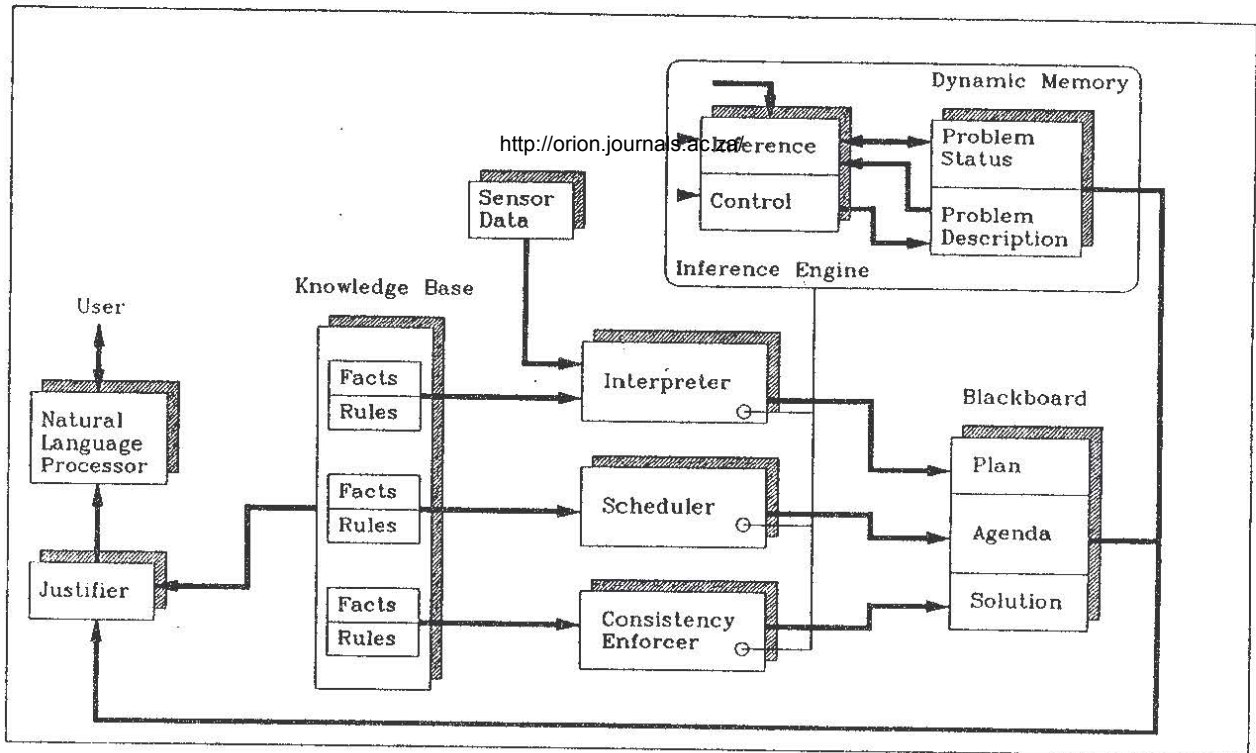
FIGURE 1(c): A COMPLEX KBS'S COMPONENTS

3.1.2 The Knowledge Base

The Knowledge Base is the KBS's hard-wired memory component. It can hold either declarative knowledge, procedural knowledge or both. Declarative knowledge comprises facts concerning (or relations between) objects and events, eg. "cast iron is brittle". Procedural knowledge comprises problem solving rules, eg. "if a machine part must not yield before failure then use a brittle material for its manufacture". Simple KBS's incorporate only one type of domain knowledge in the knowledge base but more complex systems may incorporate multiple types (known as sources) of knowledge. Knowledge can be stored for solving sub-problems, for changing the context in which only certain rules are applicable, for retaining solution consistency (with probabilities and time varying data) and for scheduling the application of the knowledge sources.

The knowledge base is structured using any one knowledge representation or a combination thereof; namely semantic networks, frames and rules. Any of these representations can be considered as creating a generalized tree structure which then forms the knowledge space (as shown in Figure 2). The method of encoding knowledge is determined by the structure of knowledge applicable in a problem's domain. The method is chosen to facilitate the ease of encoding knowledge for a particular domain and has no influence on the search technique used.

FIGURE 2: KNOWLEDGE REPRESENTATION STRATEGIES

The Semantic Net is a generalized structure used for representing large numbers of concepts as facts and their relationships. The network comprises a collection of objects called nodes. The nodes are connected together by links which represent actions, events or relations existing between the objects. The PROSPECTOR ES represents its knowledge for identifying mineral deposits in a Semantic Network [5].

The Rule based representation is a structure facilitating the encoding of procedural knowledge (it can also be made to accommodate declarative knowledge, however). This is the most common knowledge representation used for small knowledge systems. A certainty factor (probability) can be included in the rule's conclusions to indicate the certainty of the inference contained in the rule. By using these numbers it is possible to combine several sources of inconclusive information to form an almost certain conclusion. For example in the MYCIN ES the knowledge represented is used to give consultative advice on diagnosis and therapy for infectious diseases in a rule format [6].

The Frame based representation facilitates a mixture of procedural and declarative knowledge. A frame is a collection of facts and rules which together describe a stereotyped object, act or event. Stereotyping allows an element of predictability and expectation to be built into the knowledge representation. For example, if you enter a restaurant you can predict that a waiter will show you to a

chair and table. In the HEARSAY ES the knowledge represented is used to recognize spoken utterances requesting information from a data base [7].

3.1.3 The Dynamic Memory Component

All KBS's possess a dynamic memory area where the intermediate problem (or sub-problem in more complex systems) status and description are stored and manipulated. The memory area accommodating the problem status contains all facts which are known to be true as determined by the Inference Mechanism.

The problem description component contains the knowledge tree branches (rules), found by some search technique, which still need to be considered for validity by the Inference Mechanism. The history of the solution path which has been traversed through the knowledge representation tree is recorded here and accessed when explaining how a conclusion was drawn by the system.

3.1.4 The Blackboard

Whereas the Dynamic Memory component holds the intermediate problem representation, a Blackboard - essentially another dynamic memory area - is specifically used in more complex KBS's. The results of solving sub-problems (not sub-goals as in the Dynamic Memory component) of a global problem, by the Interpreter, are posted on to the Plan (which could be further partitioned into a hierarchy of results as in the HEARSAY ES). If a number of sub-problems are available for solution then a Scheduler arranges the sub-problems on the Agenda, according to some domain specific priority evaluation. The Control component of the Interpreter then considers the sub-problems as they appear in the queue. For KBS's whose global solutions are not monotonic or vary with time it is necessary to incorporate a Consistency Enforcer to correct the global possible solutions (called hypotheses) during problem solution.

3.1.5 The Control Component

The Controller determines the sequence in which branches (rules) of the knowledge tree in the Knowledge Base are to be considered during the problem's solution. Typical search strategies utilize a combination of depth-first or breadth-first search together with goal or data directed chaining (refer to Figure 3).

Depth-first search dives deep into the knowledge tree considering nodes in succession. This search is best used for problems which have a number of goals, but whose solution can be reached by any one goal. A breadth-first search traverses each level of the knowledge space before considering the next level. This search is best

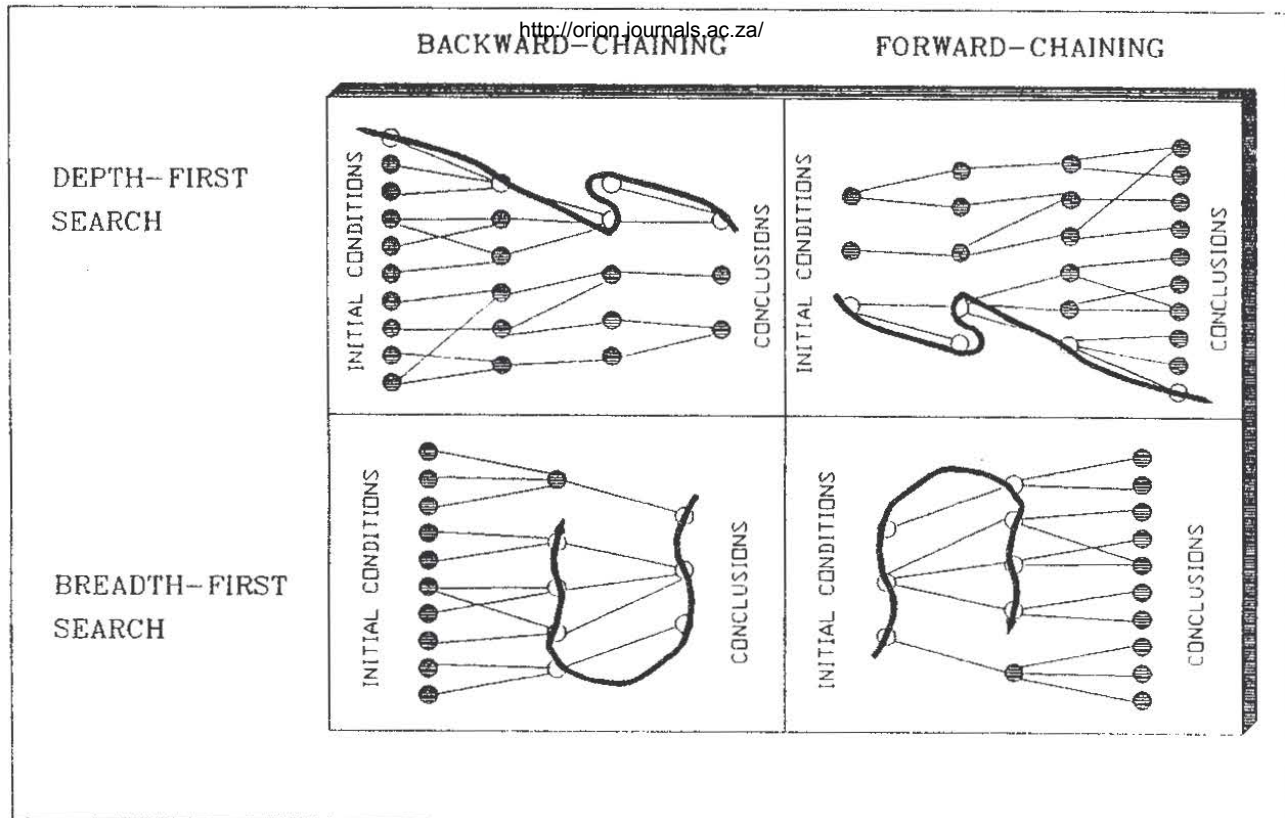used for problems which require all rules in a knowledge space to be considered for completeness.

FIGURE 3: KNOWLEDGE BASE SEARCH STRATEGIES

Goal directed search (also referred to as backward chaining) traverses a knowledge space from a problem's goals to its initial conditions. This method is best used for identification and diagnosis problems. Data directed search (also referred to as forward chaining) traverses a knowledge space from initial conditions to goal conditions. This method is best used for control problems.

3.1.6 The Inference Mechanism

The Inference Mechanism determines if a knowledge tree branch (rule) can be pursued or rejected. This is done by testing its validity against the facts contained in the Knowledge Base, the problem's status, and the details entered by the user. A simple inference mechanism would simply match symbols to test for equality. The PROLOG programming system uses a more complex mechanism called Resolution [8].

Validated knowledge facts and rules which are found to be valid are used to update the Dynamic Memory area to represent the intermediate problem solution.

3.1.7 The Interpreter

The Interpreter solves sub-problems obtained from the Agenda and updates the Plan partition of the Blackboard. It may also process low level data (sensor data) and use the results as facts to update the Plan or to solve the sub-problem.

3.1.8 The Scheduler

These search techniques instituted in the Control component may not always be efficient enough for KBS's which have large knowledge bases comprising diverse sources of knowledge. It then becomes advantageous to incorporate a Scheduler into the KBS which uses some domain-specific evaluation technique to determine the order in which sub-problems are to be considered by the Control and Inference components. Decisions made by the Scheduler are guided by applicable rules contained in the partitioned Knowledge Base. The ordered sub-problems are posted on to the Agenda partition of the Blackboard.

3.1.9 The Consistency Enforcer

The Consistency Enforcer updates that partition of the Blackboard which holds a problem's possible hypotheses and ensures that they remain consistent with the problem's solution. This Blackboard partition is commonly used in KBS's incorporating probabilities or certainty factors in their knowledge base. In its simplest form it will update the numerical probabilities associated with the possible goals. KBS's whose problems vary with time need to eliminate previously valid conclusions whose validity is nullified by the introduction of new data into the system.

3.1.10 The Justifier

A KBS must be able to disclose its line of reasoning during problem solving which will convince a person who, although not an Expert, is familiar with the problem domain. The Justifier thus draws on all the resources in the KBS in an attempt to answer user questions regarding intermediate problem statuses and the reason for the application of a particular piece of knowledge.

3.2 Architecture of a Knowledge Based System

In accordance with the KBS dictum that "knowledge is power" it is the problem domain knowledge which determines the architecture of the system. An ES will only provide problem solutions equal to or better than its human counterpart if it contains expert knowledge. It will also find solutions at a speed equal to or better than its human counterpart if the knowledge is elucidated to aid the problem search by using an appropriate knowledge representation formalism, and is structured for rapid

traverse by using an appropriate search technique. For large systems, which require the interaction of a number of knowledge sources, additional system components are required. This is for the sole purpose of fragmenting the global problem into appropriately selectable sub-problems whose application will lead to rapid problem solution.

To construct a KBS of ES proportions it is necessary to first extract, elucidate and structure a human expert's knowledge. This knowledge should then be mapped on to an appropriate knowledge representation formalism. A KBS may then be designed by specifying the necessary components and their internal operation. This step should be performed by considering other successful systems operating in similar problem domains. Development of the ES, from a prototype to a fully functional system, is then an iterative procedure where both the system components and knowledge are tuned for peak performance.

Often a shell is considered for the task of developing a KBS. KBS shells are obtained from previously successful applications. Only the knowledge is removed from the Knowledge Base leaving the rest of the system organization untouched. The quintessential characteristic of ES's, however, is that they are particularly suited to small areas of expertise (problem domains). It follows then, that its shell is only suitable for problem areas which display almost identical characteristics. If the engineer wishes to apply the KBS tool to a diversity of applications then it is highly unlikely that any one shell will be available which suits all particular problem areas.

To derive a small Knowledge System it is often possible to use a rule-based, backward chaining, depth-first search oriented, PC shell for most applications. It is essential, however, that the engineer recognizes the difference between the simple KS and the more complex ES and is not deceived into considering KS development as being an approach for the development of Expert Systems.

4. KNOWLEDGE BASED SYSTEM APPLICATIONS IN ENGINEERING

KBS applications in engineering can be classified into a few generic types: Interpretation, Identification, Diagnosis, Design, Planning and Control. Previously successful ES's, developed for these applications, display characteristic organizational strategies formulated specifically to suit the type, and use of, knowledge elucidated for problem solving. It is imperative for an engineer attempting to implement a KBS for a specific application, to identify its generic category and then to become familiar with those KBS's already applied in that specific category. This will enable the engineer to identify the present capabilities of the KBS and aid in using this technological tool for a wide range of applications.

A number of ES examples, which vary in complexity, are presented below in their generic categories. It is not intended to give an overview of Engineering applications but merely an idea of the types of applications to which KBS technology has already been applied.

- Interpretation Systems: These programs convert observations and data into symbolic descriptions (words and sentences as opposed to numeric data). Typical application areas for these systems include signal analyses, vibration analyses and speech analyses.

The HEARSAY ES converts an electronically varying signal, into the spoken sentences from which the signal was derived. The sentences are questions asking for information from a data base and are constructed from combinations of approximately 1000 words.

- Identification Systems: These systems identify or select objects based on problem attributes or requirements. Typical application areas include welding electrode selection, metal alloy identification and machine tool selection.

The Weldselector ES [9] recommends an electrode or prioritized number of electrodes which would be suitable for performing a particular welding task. The system is rule-based and interfaces to a data base containing welding electrode types and attributes.

Swift [10] describes an ES (probably a KS) which makes decisions as to what materials and coatings should be selected to fulfill specified surface finish and strength requirements.

- Diagnosis Systems: These systems identify disorders, malfunctions and inconsistencies in observed data and can determine reasons for incongruous behaviour. Typical application areas for these systems include fault finding in vehicle engines, electronic equipment and computer software.

The General Electric Company developed a rule-based, backward chaining ES to assist mechanics in diagnosing and repairing Diesel Locomotive Engine problems [11]. After the user selects a problem area, the system prompts the user to answer questions concerning the engine's symptoms. Once the fault has been identified the ES advises on how to repair the fault in the form of training video sequences.

- Design Systems: These systems assemble components of a product which adhere to the constraints given for the design. Typical application areas include mechanical design and computer configuration.

An ES has been developed at the University of Houston [12] which designs pipe flanges. The system incorporates the design experience of a number of experts which has greatly improved the speed and efficiency of such designs.

The XCON ES [13] developed and implemented by Digital Equipment Corporation configures VAX computers according to a client's specifications. Program performance is reported to be better than any human expert in the field.

-    Planning Systems: These systems arrange actions or functions for a sequential attainment of a final goal or solution. Typical application areas include process planning and manufacturing scheduling.

At the University of Maryland, Nau [14] investigated the use of an ES for process planning. The system plans a sequence of machining operations to perform a job at least cost and specifies the type and size of machine tools to be used.

Taylor [15] has developed an ES which plans welding process and edge preparation techniques which are appropriate when welding materials of various dimensions and properties.

-    Control Systems: These systems often encompass a number of the above categories. They must repeatedly interpret current situations, predict the future, diagnose the cause of anticipated problems, formulate a remedial plan, and monitor its execution to ensure success.

Agogino, Srinivas and Schneider [16] have developed a multiple sensor ES for diagnosing, monitoring and controlling an unattended milling machine. It monitors the machine and detects inconsistencies in expected and observed machine behaviour. Once it has diagnosed operational behavioural problems (such as chatter) it performs controlled actions to restore desired matching operations.

Greef and Reinecke [17] at the University of Stellenbosch have investigated the utilization of KBS's for the supervision and control of manufacturing devices and Flexible Manufacturing Cells (FMC). In this system, a KBS is specifically devoted to each device in a FMC. The KBS's then communicate with each other, via a network, thereby directing the control of their attached devices and interactively directing the control of the FMC.

5. APPLICATION OF KBS'S IN SOUTH AFRICAN MANUFACTURING

In the face of international competition South African industry is beset by two major problems which demand solution: firstly, a scarcity of expertise and skills in all spheres of manufacturing, and secondly a large unskilled labour force which cannot be replaced by automation.

KBS technology can help to alleviate these problems by virtue of its main characteristic: KBS's can hold and manipulate encoded human expertise and skills. By exploiting this characteristic, engineers can apply this technology to problems previously unsolvable by algorithmic computer techniques. The scarcity of expertise and skill can be alleviated by distributing KBS's for supporting personnel in all

manufacturing tasks. In this way the second problem can also be addressed by using KBS's to support unskilled labour during the supervision, control and maintenance of automated equipment. http://orion.journals.ac.za/

While it is accepted that this technology is not a panacea for South Africa's problems, the intensity with which the industrialised nations are researching the field is indicative of its potential in manufacturing and cannot be ignored. What remains then, is for the South African engineer to explore the power of the KBS, to exploit its characteristics and to investigate new applications.

6. CONCLUSION

This paper has attempted to point out the deceptive nature of Knowledge Based System terminology by considering the evolution of these systems and has shown, by way of describing both simple and complex system organization, that it is confusing to transfer simple terminology, concepts and methods of architectural design from Knowledge Systems to more complex Expert Systems. Furthermore, the selected KBS applications presented are indicative of the power of Knowledge Based Systems which is manifest in their wide range of applications and demonstrated in their ability to solve problems equally well or better than their human counterparts.

This technology has great potential for alleviating some of South Africa's manufacturing problems. By understanding the field of KBS and its associated terminology and concepts, a powerful new tool will be added to the engineer's tool box, allowing appropriate utilization of the technology for a variety of applications.

REFERENCES

[1]   A.R. GREEF, R.R. REINECKE, Problem Solving using Artificial Intelligence Techniques, *ORiON*, Vol 4, No 1, 47-64 (1988).

[2]   P.H. WINSTON, *Artificial Intelligence*, (2nd ed), Addison-Wesley Publishing Company, Inc., USA (1984).

[3]   M. MINSKY, A Framework for Representing Knowledge in Winston, P.H., (ed), *The Psychology of Computer Vision*, McGraw-Hill, 210-277 (1975).

[4]   E. TURBAN, Review of Expert Systems Technology, *IEEE Transactions on Engineering Management*, Vol 35, No 2, 71-81 (1988).

[5]   F. HAYES-ROTH, D. WATERMAN, D. LENAT, *Building Expert Systems*, Addison-Wesley Publishing Company, Inc., 54-55 (1983).

[6]   T. O'SHEA, M. EISENSTADT, *Artificial Intelligence. Tools, Techniques, and Applications*, Harper & Row, Publishers, New York (1984).

[7] L.D. ERMAN, V.R. LESSER, The HEARSAY-II Speech Understanding System: A tutorial, *Trends in Speech Recognition*, Wayne, A.L., (ed), Prentice-Hall, Inc., N.J., 361-381 (1980).

[8] R. KOWALSKI, *Logic for Problem Solving*, Elsevier Science Publishing Co., Inc., New York (1979).

[9] G.H. SCHAFFER, Artificial Intelligence: A tool for smart manufacturing, *American Machinist and Automated Manufacturing*, 83-94 (1986).

[10] K. SWIFT, A. MATTHEWS, Expert Systems in Engineering Design, *Engineering*, 673-678 (1983).

[11] R.K. MILLER, *Artificial Intelligence, Application for Manufacturing*, SEAI Technical Publications, USA (1984).

[12] K.J. KORANE, Applying Expert Systems to Mechanical Design, *Machine Design*, 121-124 (1986).

[13] D.G. BOBROW, S. MITTAL, M. STEFIK, Expert Systems: Perils and Promise, *Communications of the ACM*, Sept 1986, Vol 29, No 9, 880-894 (1986).

[14] D.S. NAU, *Expert Computer Systems and their Applications to Automated Manufacturing*, National Bureau of Standards, USA (1982).

[15] W.A. TAYLOR, Expert Systems to Generate Arc Welding Procedures, *Metal Construction*, 426-431 (1986).

[16] A. AGOGINO, S. SRINIVAS, K. SCHNEIDER, Multiple Sensor Expert System for Diagnostic Reasoning, Monitoring and Control of Mechanical Systems, *Mechanical Systems and Signal Processing*, Vol 2, No 2, 165-185 (1988).

[17] A.R. GREEF, R.R. REINECKE, A Micro-computer Based Knowledge System for Robot Supervision, *Conference Proceedings of the 18th International Symposium on Industrial Robots*, Lausanne, Switzerland, edited by Prof C.W. Burckhardt, I.F.S. Publications, UK, (1988).