An Overview of the Development Principles, Stages and Building Blocks of Expert System

Ele, Sylvester I. [1], **Umoh, E. E** [2] **and Adesola, W. A.** [1]

[1]ICT Development Department, Governor's Office, Calabar, Cross River State., el_silver2@yahoo.com, shobisi@hotmail.co.uk
[2]Department of Computer Science, Cross River University of Technology, Calabar, Cross River State.,enoimah@yahoo.com

Abstract

Expert Systems are not suited for all types of problems. Many developers actively seek problems pliable to Expert System solution or tried to solve all problems encountered using Expert System. Experience has also shown that developers' attention has become more focused on the problems to be solved rather than on the solution techniques. Little or no attention is paid to development details. This paper presents the development issues; the system engineering techniques and tools; the systematic fashion in the development much like the systems methodology and model development steps; prototype documentation; cyclic development life cycle and the building blocks of Expert System. The study is expected to serve as a template and reference material for upcoming researchers in Expert System development in order to concentrate on solution techniques rather than the problem to be solved, and pay adequate attention to development details. This will ensure that the exact methodology used and quality software is developed to meet users' needs.

Keywords: Expert System, Knowledge Engineer, Domain Expert, Prototyping.

1.0. Introduction

Experts Systems are computer programs that are derived from a branch of computer science research called Artificial Intelligence (AI). AI's scientific goal is to understand intelligence by building computer programs that exhibit intelligent behavior. It is concerned with the concepts and methods of symbolic inference, or reasoning, by a computer, and how the knowledge used to make those inferences will be represented inside the machine.

Artificial Intelligence programs that achieved at expert-level competence in solving problems in task areas by bringing to bear a body of knowledge about specific tasks are called Knowledge-based or Expert Systems. Often, the term Expert Systems is reserved for programs whose knowledge base contains the knowledge used by human experts, in contrast to knowledge gathered from textbooks or non-experts. More often than not, the two terms, Expert Systems (ES) and knowledge-based systems (KBS), are used synonymously. The area of human intellectual endeavor to be captured in an Expert System is called the *task domain*. *Task* refers to some goal-oriented, problem-solving activity. *Domain* refers to the area within which the task is being performed. Typical tasks are diagnosis, planning, scheduling, configuration and design. An example of a task domain is aircraft crew scheduling. The specific task of an expert system is to be an alternative source of decision-making ability for organization to use instead of relying on the expert knowledge or skill of few people or just one person. The focus in the development of expert system is to acquire and represent the knowledge and experience of a person(s) who have been identified as possessing the special skill or mastery in a particular field [1].

According to Jones and Barrett [2], Expert systems are not suited for all types of problems. Initially, many developers actively sought problems amenable to expert system solution or try to solve all problems encountered using Expert system. Expert systems are verified specifically. Petrovic [3] also noted that expert systems are tailored-made for specific and narrowly defined problem area. For example, a diagnostic expert system for troubleshooting computers must actually perform all the necessary data manipulation as a human expert would. The developer must limit his or her scope of the system to just what is needed to solve the target problems. Duke and Regenie [4] admitted that expert system technology has been identified as a potential solution to some of these problems.

1.1 Objective

The primary objective of this paper is to present a detailed description of the system engineering techniques and tools; the systematic fashion in the development of Expert System; the methodology and model development steps; Prototype documentation approach; the development life cycle and building blocks of Expert System.

2.0 Literature Review

2.1 Early Expert System Work

Expert systems were the first major successful application technology to evolve from Artificial Intelligence research.The foundations of the field of Artificial Intelligence can be traced from many different disciplines including philosophy, mathematics, psychology, computer engineering, and linguistics [5].The first cited work in the area of Artificial Intelligence dates back to McCulloch and Pitts in 1943 [6]. They proposed a model of artificial neurons that mimic the structure of the human brain; this area later became the connectionist paradigm. In the summer of 1956, John McCarthy organized a two-month workshop at Dartmouth and 10 leading U.S. researchers interested in automata theory, neural networks, and the study of intelligence were invited. Two researchers from Carnegie Tech (now known as Carnegie Mellon University), Allen Newell and Herbert Simon were the focus of the workshop due to their reasoning program known as the Logic Theorist (LT). Soon after the workshop, LT was able to prove most of the theorems in Russell and Whitehead's *Principia Mathematica*.

The work of Newell and Simon is the first documented work using the symbolic programming Paradigm of AI. Their work on LT led them to develop another program known as general Problem Solver (GPS). The success of GPS was not as widely heralded however because of the limited class of problems that it could solve. GPS was designed from the start to imitate human problem-solving Protocols regardless of the information contained in the domain. Researchers then took the opposite approach in the development of the DENDRAL program [7]. They applied the knowledge of analytical chemists to infer the molecular structure from the information provided by a mass spectrometer. DENDRAL holds a significant place in the history of Expert Systems because it was the first system to use the expertise of human problemsolvers and translate that knowledge into a large numbers of special purpose rules, known as a rule –based system. The work on DENDRAL leads to many others successful applications of this new technology known as expert systems. Feigenbaum and others at Stanford began the Heuristic Programming Project (HPP) to investigate other problem domains that could benefit from this new technology. The next major effortwas in the area of medical diagnosis. Bruce Buchanan and Dr. Edward Shortliffe developed MYCIN to diagnose blood infections [8], [9]. Using about 450 rules, MYCIN was able to perform as well as some experts, and considerably better than some junior doctors were. MYCIN is one of the most widely known of all expert system applications developed. However, MYCIN is significant to the history of expert or knowledge-based systems for two particular reasons. First, unlike DENDRAL, which used a 7model of a particular molecule as the basis for its reasoning, MYCIN was constructed from interviews with various doctors in the particular domain. Therefore, MYCIN contains a number of heuristic rules that are used by physicians in the identification of certain infections. The second major contribution of MYCIN was the later development of EMYCIN (Empty MYCIN). EMYCIN was the first Expert System shell. It took approximately 20 man-years to develop the MYCIN program.

There were other significant expert system applications that were also developed in the early days of Expert Systems. These systems include PUFF, which used EMYCIN in the domain of pulmonary disorders, DELTA/CATS, which was developed at General Electric Company to assist railroad personnel in the maintenance of GE's diesel-electric locomotives [10]. Also at this time, researchers at CMU developed the first truly successful commercial application of expert systems. The system, developed for Digital Equipment Corporation (DEC), was used for computer configuration and known as XCON (R1). XCON, originally titled R1, was developed by John McDermott at CMU for aiding in the configuration of VAX and PDP-11 computer systems at DEC.

2.2.1 Fundamental Features of ExpertSystem

Expert System developers who wish to make significant impact and develop a standard Knowledge-based System must take the following distinctive features to bear when developing Expert Systems.

a) Expert systems require special approaches to systems analysis, especially to the collection of the data (or rather knowledge) on which the system is based.The process of gathering the knowledge (knowledge Acquisition) to stock the expert system's knowledge base has proved to be the most difficult component of the knowledge engineering process. It's become known as the 'knowledge acquisition bottleneck', and Expert System projects are more likely to fail at this stage than any other.Knowledge acquisition almost always involves extracting knowledge from someone who is expert in the field concerned (a domain expert). This process, knowledge elicitation, involves a variety of interview and non-interview techniques.

b) Expert systems require particular attention to the human-computer interface. User interfaces for Expert Systems are more troublesome, and harder to develop, than those of conventional pieces of software. This is because, for various reasons, the interactions between computer and user are more complex than those involved in a conventional piece of software. For instance:

Conventional software typically performs some task, perhaps in interaction with the user. Expert systems typically assist with decision-making about how a task is to be tackled, and this means that the information that must be exchanged between the system and the user is more complex.

Different users differ in the sorts of problem-solving style they prefer.

c) Expert systems projects require special approaches to software management.The methodologies used to build expert systems have been shaped by the problems with knowledge acquisition, described earlier.For a long time, the favourite development methodology was rapid prototyping. (Cyclical development means more or less the same thing). In the mid-1980s, this approach came under criticism, because it appeared to have all the shortcomings of the unstructured approaches which had been used, with very poor results, in the early days of mainstream software. But the Structured Systems Analysis & Design methodologies did not seem to be appropriate, because of the differences between knowledge and data.

As a result, special-purpose development methodologies for knowledge engineering were developed. The most well-known is KADS, which was developed at the University of Amsterdam, as part of the ESPRIT programme, in co-operation with several European partners.

2.3 **Expert Systems Cyclical Development/Rapid Prototyping**

If an Expert System solution is suitable, one should approach the development in a logical fashion much like the systems methodology steps and the model development steps. The process is largely one of improvement and development of a prototype. The knowledge base increases in both depth and breadth with organizational and representational improvements while helping guide successive stages of development. The prototype becomes the basis for further development, whether it is refined or discarded and the process restarted. It helps identify approaches that have the most merit and others that should be discarded. These decisions can be made early, minimizing the cost of development [2].

In the 1980s, the favourite approach was based on rapid prototyping (also known as cyclical development or evolutionary prototyping).The essential principles are:

- Get a prototype- a small, preliminary version of the final system, up and running at an early stage;
- Present this to the domain expert, for criticism. The domain expert is the expert in the field under attention;
- Proceed to refine this prototype with repeated debugging & knowledge accretion stages;
- Continue with this cycle until the knowledgebase is finished.

In this paper, the researcher uses schemas to explain the various principles essential in the cyclical development of Expert System as described in the bullets above. Any developer desiring to build a standard Expert or Knowledge base System, and wish to ensure good programming practice adopts these principles.

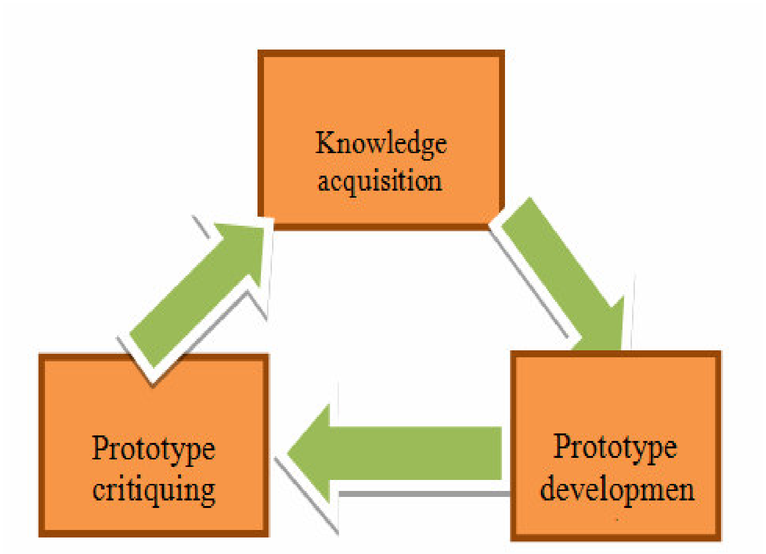

Fig. 1: Prototype at an early Stage

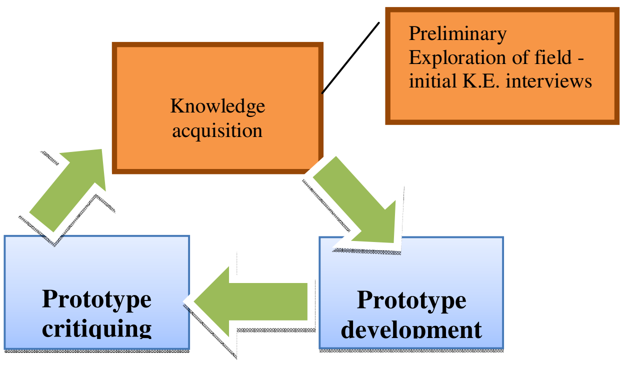

Figure 2:**Knowledge Engineer embarks on initial exploration of field (Interview with field Experts).**

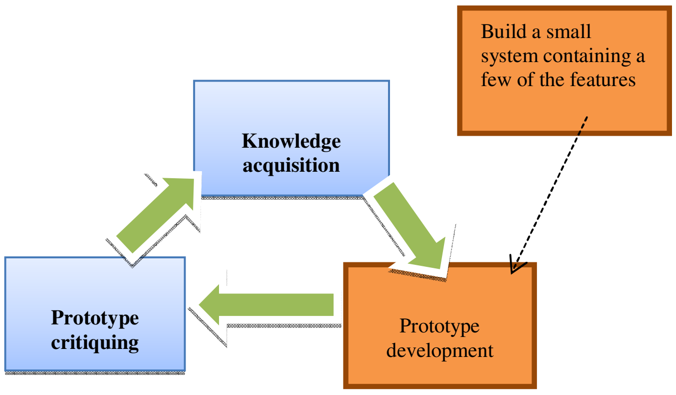

Figure 3:**The K.E. Build a small system containing few of the features from the Prototype.**

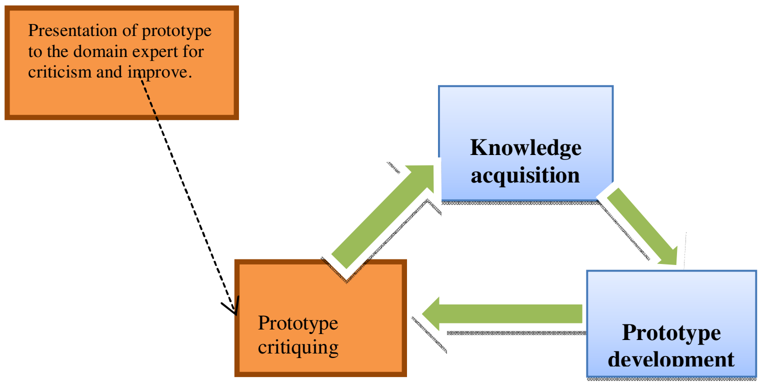
Presentation of prototype to the domain expert for criticism and improve.
Knowledge acquisition
Prototype critiquing
Prototype development

Figure 4: **Knowledge Engineer Present Prototype to the domain Expert**

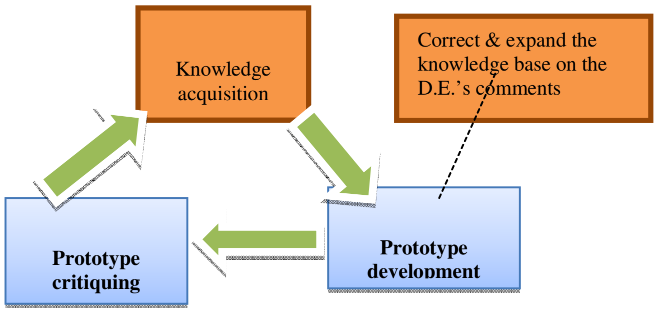

Figure 5:**The Knowledge Engineer Correct and Expend the knowledge base on the Domain Expert's comments.**

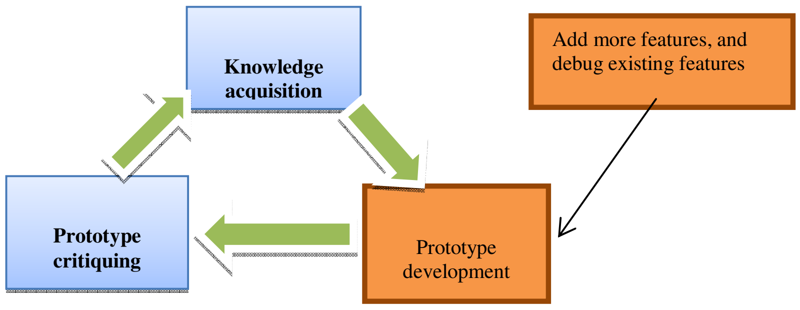

Figure 6:**Knowledge Engineer Adds more features, and debug existing features/modules to ensure error free program**

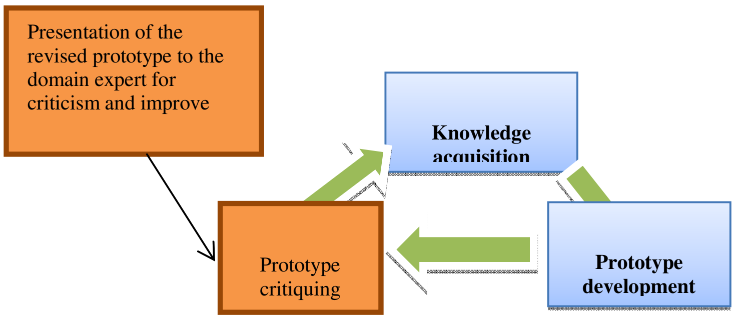
Presentation of the revised prototype to the domain expert for criticism and improve
Knowledge acquisition
Prototype critiquing
Prototype development

Figure 7:**The Knowledge Engineer presents the revised prototype to Domain Expert for criticism and improvement**

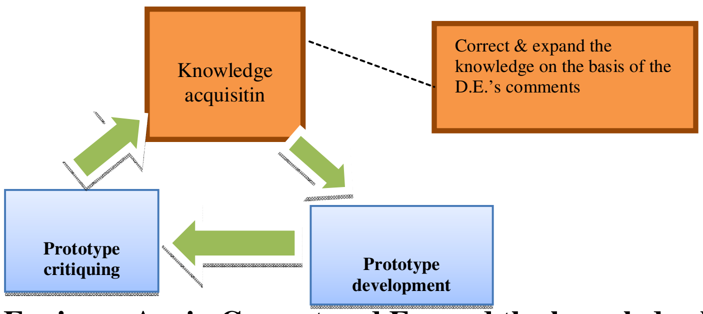
Knowledge acquisitin
Correct & expand the knowledge on the basis of the D.E.'s comments
Prototype critiquing
Prototype development

Figure 8:**The Knowledge Engineer Again Correct and Expand the knowledge base on the Domain Expert's comments.**

These principles have the advantage that the Knowledge Engineer is able to show early progress in the Knowledge elicitation task.It also generates enthusiasm in the Domain Expert.

2.4 **Expert System Development Life Cycle /Stages**

Several general approaches for developing Expert System have been proposed and adopted by many developers. But this paper discusses on the most widely accepted approaches provided by two scholars; [11], [12].

Turban's Account of the Expert System Development Lifecycle.

Turban[12] identifies the six phases and sub-phases in the development of an Expert System:

1. Project initialization
2. Systems Analysis and Des
3. Prototyping
4. System Development
5. Implementation and
6. Post-Implementation.

This is probably a good account of the way knowledge engineering projects are currently organisedin America.

Phase 1: Project Initialisation

a. Problem definition
b. Needs assessment
c. Evaluation of alternative solutions
d. Verification that an Expert System approach is appropriate
e. Consideration of management issues

It is important to discover what problem(s) the client expects the system to solve for them, and what their real needs are. The problem may very well be that more knowledge is needed in the organisation, but there may be other, better ways to provide it. 'Management issues' include availability of finance, legal constraints, and finding a 'champion' in top management.

Phase 2: System Analysis & Design

a. Produce conceptual design

b. Decide development strategy
c. Decide sources of knowledge, and ensure cooperation
d. Select computer resources
e. Perform a feasibility study
f. Perform a cost-benefit analysis.

The 'conceptual design' will describe the general capabilities of the intended system, and the required resources. There are also the problems of selecting software, and finding a domain expert (and persuading him/her to cooperate).

Phase 3: Prototyping

a. Build a small prototype
b. Test, improve and expand it
c. Demonstrate and analyse feasibility
d. Complete the design.

It is important to establish the feasibility (economic, technical and operational) of the system before too much work has been done, and it is easier to do this if a prototype has been built.Rapid prototyping provides a prevue of what the completed product will be like. It is important to communicate and follow progress in any project, not only for funding agencies and supervisors, but also for the domain expert who is interested in making best use of valuable time [2].

Phase 4: System Development

a. Build the knowledge base
b. Test, evaluate and improve the knowledge base
c. Plan for integration

The evaluation of an expert system (in terms of validation and verification) is a particularly difficult problem.

Phase 5: Implementation

a. Ensure acceptance by users
b. Install, demonstrate and deploy the system
c. Arrange orientation and training for the users
d. Ensure security
e. Provide documentation, Arrange for integration and field testing

If the system is not accepted by the users, the project has largely been a waste of time. Field testing (leading to refinement of the system) is essential, but may be relatively lengthy.

Phase 6: Post-implementation

a. Operation
b. Maintenance
c. Upgrading
d. Periodic evaluation

A person or group of people must be put in charge of maintenance (and, perhaps, expansion). They are responsible for correcting bugs, and updating the knowledgebase. They must therefore have some knowledge engineering skills. The system should be evaluated, once or twice a year, in terms of its costs and benefits, its accuracy, its accessibility, and its acceptance.

What is quite interesting in this approach is that Turban leaves open the options that the prototype which features in phase 3 might be an **evolutionary** prototype or a **throwaway** (off-the-cuff)prototype. In the 1st case (evolutionary), phase 4 would consist mainly of expanding this prototype, by adding more and more knowledge, until it became the knowledge base for the finished system. In the 2nd case (throwaway), it would consist of drawing lessons from building the prototype and using these to assist in building the knowledge base from scratch, using a more appropriate tool.

In other words, the development lifecycle as described is either a disguised form of the rapid prototyping (cyclical development) procedure mentioned previously, or it is a substantially different approach which is liable to produce a substantially different knowledge base.

The tendency among European knowledge engineers is to reject evolutionary prototyping (and rapid prototyping / cyclical development) in favour of throwaway prototyping.

Waterman's Account of the Expert System Development Lifecycle.

Waterman [11] provided five phases/stages approach in the development of Expert System: Identification , Conceptualization, Formalization, Implementation, and Testing.These agreed with the stages/phases of Knowledge Engineering creation postulated by Sasikumar*et al* [13].

Identification

Identification is the requirements analysis step carried out in traditional software development. It involves a formal task analysis to determine the external requirements, form of the input and output, setting where the program will be used and determines the user [11]. Sasikumar*et al* [13], further explained that domain identification requires that the Knowledge Engineer should acquire adequate familiarity with the field; decide that the task is appropriate for using an Expert System; exercise his judgment to define the scope of the Expert System to be built; decide what hardware and software will be required to implement the Expert System; estimate the effort involved in creating the Expert System; estimate the number of professionals to be involved and the time to be taken in creating the Expert System; and create a rapid prototype of the Expert System, to get a better estimate of the effort involved.

Conceptualization

The second stage of ES development, conceptualization, involves designing the proposed program to ensure that specific interactions and relationships in the problem domain are understood and defined. The key concepts, relationships between objects and processes and control mechanisms are determined. This is the initial stage of knowledge acquisition.

Formalization

Formalization involves organizing the key concepts, sub-problems and information flow into formal representations. In effect, the program logic is designed at this stage (Waterman). The structure used to represent knowledge includes hierarchy, taxonomy and IF-THEN form [13].

Implementation

During the next stage, implementation, the formalized knowledge is mapped or coded into the framework of the development tool to build a working prototype. The contents of knowledge structures, inference rules and control strategies established in the previous stages are organized into suitable format.

Testing

The last stage, testing, involves considerably more than finding and fixing syntax errors. It covers the verification of individual relationships, validation of program performance and evaluation of the utility of the software package. Testing guides reformulation of concepts, redesign of representations and other refinements.

Some knowledge engineers would object to Turban's sequence of steps on the grounds that the computer resources should not be finally selected until the precise nature of the knowledge had been established. That is not until after phase 3 (Prototyping phase).

3.0 Expert System Building Blocks

A typical rule based (knowledge base) Expert System consists of three major components (called the kernel of the Expert System). They are; the Working memory, the knowledge base (Rule and Facts), and the Inference engine [14].The kernel of an Expert System contains those components that are the basic and the required components for all expert systems, hence, they are refers to as the building blocks of expert system [1].The figure below provides an overview of the kernel of the Expert System.

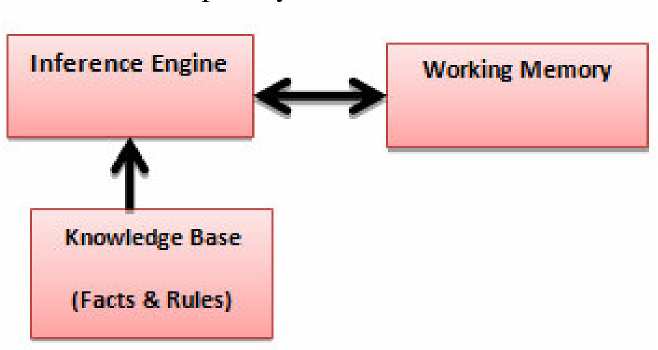

Figure 9. Kernel of Expert System

The knowledge base in an expert system kernel consists of both a fact and a rule base. The fact base contains up-to-date information and data on the state of that portion of the environment that is pertinent to the expert system kernel. The rule base is typically populated with rules of the following form: AB. This is interpreted as "if condition A is satisfied then do B". The "A" portion of the rule is called the antecedent or LHS (Left Hand Side) of the rule. The "B" portion of the rule is called the consequent or RHS (Right Hand Side) of the rule. The condition "A" may be a conjunction of conditions A1, A2, ...,An which must all be satisfied in order to trigger any actions stipulated by B. The relationship between the rule base and the fact base is quite straightforward. If there is a fact in the fact base like "Var1 = n" and there is a rule in the rule base that states that "If Var1 = n then B" then this rule is considered for execution (or firing) [15].

The inference engine (mechanism) is that part of the expert system kernel which supports reasoning about the environment by proper manipulation of its rule and fact bases. Reasoning is the standard technique by which Expert system solve problem. Expert systems are commonly used when an inadequate algorithm or no algorithmic solution exists and reasoning offers the only possibility of solution [16]. There are basically two ways or control strategies by which the inference engine manages rules to arrive at some conclusion or to arrive at a sequence of actions to be taken with respect to the environment. These are forward and backward chaining. Most expert systems support only one control strategy while some support both. For the purpose of clarity, the researcher presents the various methods of inference using the scheme below.

The working memory (WM), in the other hand, represents the set of facts known about the domain. The elements reflect the current state of the world. In an expert system, the Working Memory typically contains information about the particular instance of the problem being addressed. For example, in a medical expert system, the Working Memory could contain the details of a particular patient being diagnosed.

The diagram below [17] shows the architecture of a rule-based ES.

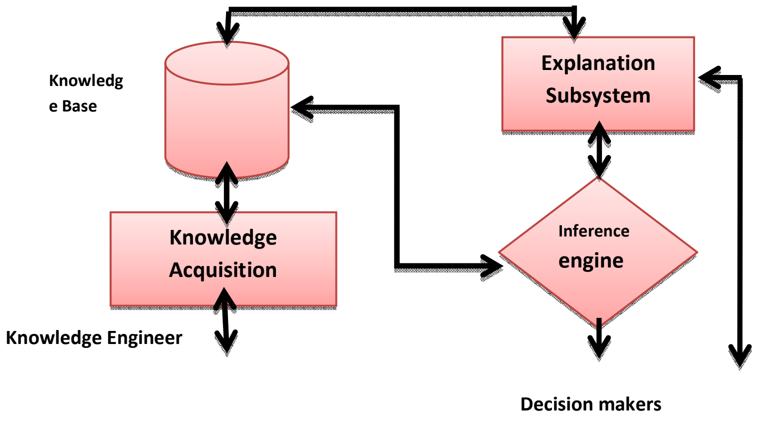

Figure 10a: **Architecture of A Rule-Based Expert System**

3.2 Expert System Organization and Operating Environment

The figure below show the organization and operating environment of a typical Expert system

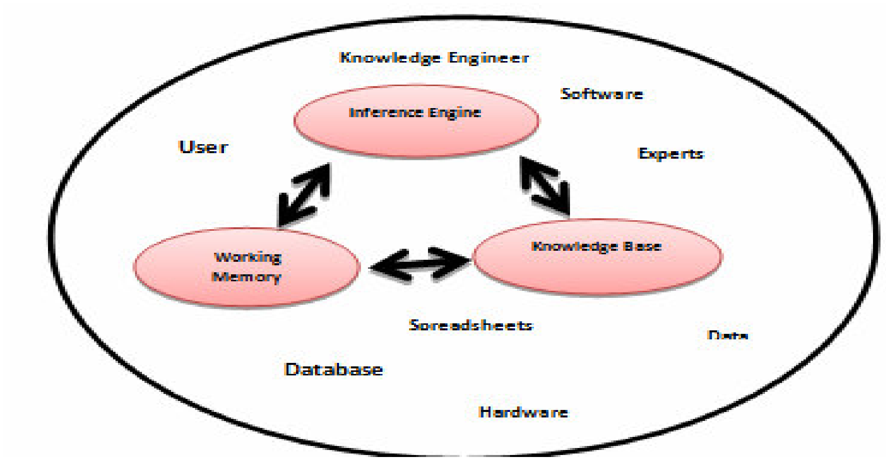

Fig. 10b

4.0 Expert System Application Roadmap

The symbolic processing capability of Artificial intelligence technology leads to many potential application in engineering and manufacturing. With the increase in sophistication of Artificial Intelligence techniques, analysts are now able to use innovative methods to provide viable solutions to complex problems in everyday applications. Figure 10c presents a structural representation of application paths for Artificial Intelligence and Expert Systems.

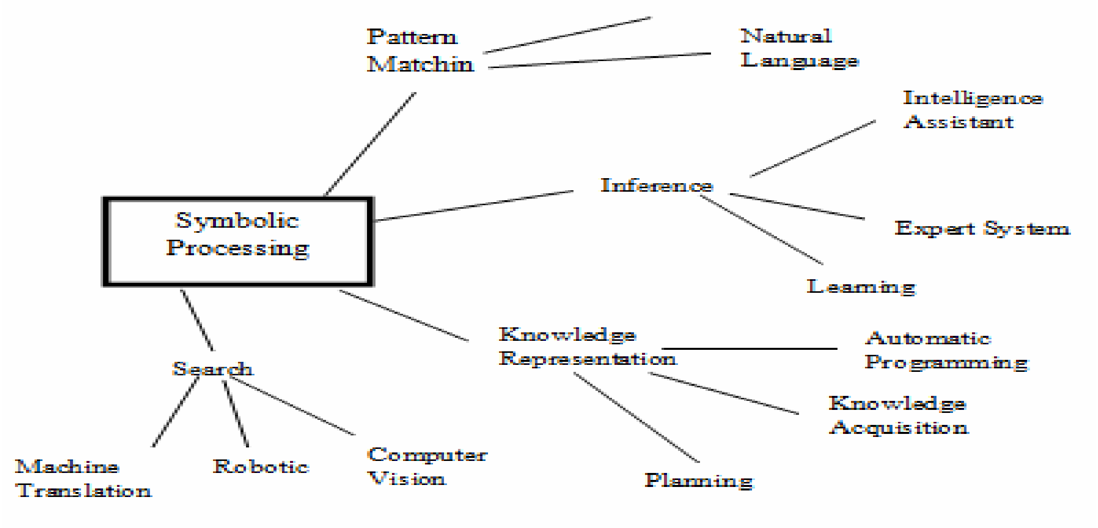

Fig. 10c: Application Roadmap for Expert System

Summary and Conclusion

In this paper we present an overview of Expert System development principles, stages and its building blocks. Efforts have been made to provide comprehensive discussion on the fundamental understanding of the basic ideas of the phases of Expert System development with respect to its methodology, andits principles and stagesof development.Its structures, application roadmap, and building blocks were also presented. The practical benefits of Experts System have been realized by many organizations and the future development of this technology will only increase with time due to the fact that more complex problems, in critical domains can now be addressed.

.

.

References

. [1]. Pomykalski, J. J., Truszkowski, W. F. and Brown, D. E.(1999) "Expert System", In The Wiley Encyclopedia for Electrical and Electronics Engineering (edited by J. Webster), February.

[2]. Jones, D.D. and Barrett, J.R. (1989), "Building Expert Systems", in The ASAE Monograph No.8, ASAE, St. Joseph, MI.

[3]. Petrovic, S. "Decision Support Methodologies", School of Computer Science, University of Nottingham.

[4]. Duke, E.L. and Regenie, V.A. (1985), "Expert Systems Development and Application", in The national Aeronautics and Space Administration", Ames Research Centre, Edwards, Califonia

[5]. Russell, S.J. andNorvig, P. (1995), "Artificial Intelligence*:* A Modern Approach", Englewood Cliffs, NJ, Prentice-Hall.

[2]. Jones, D.D. and Barrett, J.R. (1989), "Building Expert Systems", in The ASAE Monograph No.8, ASAE, St. Joseph, MI.

.[6]. McCulloch, W.S. and Pitts, W. (1943), "A logical Calculus of the ideas Immanent in Nervous activity", Bulletin of Mathematical Biophysics, **5**: 115-137, 1943.

. [7]. Lindsay, R.K., Buchanan, B.G., Feigenbaum, E.A. and Lederberg J.(1980), Applications of Artificial Intelligence for Chemical Inference: The DENDRAL Project", *New* York, NY: McGraw-Hill

.[8]. Shortliffe, E.H. (1976), "Computer-Based Medical Consultations*:* MYCIN", New York, NY: Elsevier.

[9]. Buchanan, B.G. andShortliffe, E.H.,eds. (1985), "Rule-Based Expert Systems: The MYCIN Experiments ofthe Stanford Heuristic Programming Project", Reading, MA: Addison-Wesley.

[10]. Ignizio, J.P. (1991), "Introduction to Expert Systems: The Development and Implementation of Rule-BasedExpert Systems*"*, New York, NY: McGraw-Hill.

[11]. Waterman, S.A. (1986) “A Guide to Expert Systems”, Addison-Wesley Publishing co., Inc., Reading, M.A.

[12]. Turban, E. & Aronson, J. E. (1998), Decision Support Systems & Intelligence Systems. Prentice Hall,

[13]. Sasikumar, M. et al. (2007), “A Practical Introduction to Rule-Based Expert Systems”, Norasa publishing, New Delhi, pp. 141.

[14]. Noran, O. S. “The Evolution of Expert Systems”, School of Computer and Information Technology. http://www.cit.gu.edu.au/˜noran

[15]. Hayes-Roth (1998), “Rule-Based Systems”, Communication of the ACM, 28(9):929.

[16]. Giarratano, J. and Riley, G. (1999), “Expert System: Principles and programming, 3ed”, PWS publishing company, Boston

[17].Adhikar, B., Ansri, Md.H.,Priti, S. and Susma, P. (2008), “Neurology Diagnosis System”, (A project proposal presented to Advanced College of Enginnering and Management, Tribhun University), September

[18] Buchanan, B.G. et al. (1983), “Constructing an Expert System”, Addison-Wesley, Reading, MA.