International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

General Article

Expert System of AI

Sangeeta Rani[Á*]

[Á]BLS Institute of Technology Management, Bahadurgarh. India

Accepted 20 Sept 2014, Available online 01 Oct 2014, **Vol.4, No.5 (Oct 2014)**

Abstract

Expert System are the knowledge intensive programs that solve problems in a domain that requires considerable amount of technical expertise. It is one of the largest area of applications of artificial intelligence is in expert systems, or knowledge based systems as they are often known. This type of system seeks to exploit the specialized skills or information held by of a group of people on specific areas. It can be thought of as a computerized consulting service. It can also be called an information guidance system. Such systems are used for prospecting medical diagnosis or as educational aids. They are also used in engineering and manufacture in the control of robots where they inter-relate with vision systems. The initial attempts to apply artificial intelligence to generalized problems made limited progress as we have seen but it was soon realized that more significant progress could be made if the field of interest was restricted

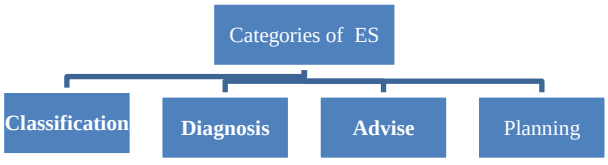
.

Keywords: *Knowledge Base, Inference Engine, User Interface, Shell, Domain Expert, Acquisition, Explanation Facility , External Interface ,Planning, Diagnosis*

Introduction

ES 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 Of course, the term intelligence covers many cognitive skills, including the ability to solve problems, learn, and understand language; AI addresses all of those. But most progress to date in AI has been made in the area of problem solving -- concepts and methods for building programs that reason about problems rather than calculate a solution.AI programs that achieve 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 first expert system was developed in 1965 by Edward Feigenbaum and Joshua Lederberg of Stanford University in California, U.S. Dendral, as their expert system was later known, was designed to analyze chemical compounds. Taken together, they represent the most widespread type of AI application.

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. Building an expert system is known as knowledge engineering and its practitioners are called knowledge engineers. The knowledge engineer must make sure that the computer has all the knowledge needed to solve a problem. The knowledge engineer must choose one or more forms in which to represent the required knowledge as symbol patterns in the memory of the computer -- that is, he (or she) must choose a knowledge representation. He must also ensure that the computer can use the knowledge efficiently by selecting from a handful of reasoning methods. In short we can say that Expert systems are computer applications which embody some non-algorithmic expertise for solving certain types of problems. For example, expert systems are used in diagnostic applications servicing both people and machinery. They also play chess, make financial planning decisions, configure computers, monitor real time systems, underwrite insurance policies, and perform many other services which previously required human expertise. There are 4 Categories of Expert system:-Classification, Diagnosis, Advice, Panning

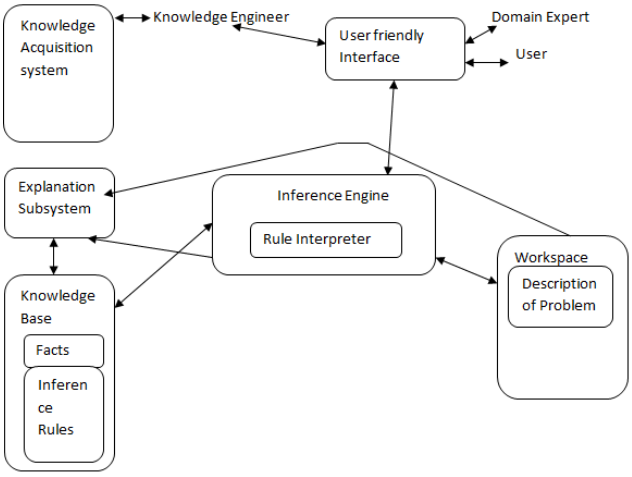


*Corresponding author: **Sangeeta Rani**

Architecture of Expert System

The development environment includes the activities and support that are necessary to acquire and represent the knowledge as well as to make inferences and provide explanations. The major players in this environment are the knowledge engineer and the domain expert who act as builders. Once the system is completed it is used for consultation by the no expert user via the consulate We first describe the components of expert systems that are as follows:

- Knowledge Base
- User Interface
- Inference engine
- Explanation Facility
- Knowledge acquisition Facility



Knowledge Base:- Knowledge-Based systems were first developed by Artificial Intelligence researchers. These early knowledge-based systems were primarily expert systems. The knowledge base of expert systems contains both factual and heuristic knowledge. Factual knowledge is that knowledge of the task domain that is widely shared, typically found in textbooks or journals, and commonly agreed upon by those knowledgeable in the particular field. Knowledge engineering. A Knowledge-based system (KBS) is a computer program that reasons and uses a knowledge base to solve complex problems.

The term is broad and is used to refer to many different kinds of systems. The one common theme that unites all knowledge based systems is an attempt to represent knowledge explicitly via tools such as ontologies and rules rather than implicitly via code the way a conventional computer program does. A knowledge based system has two types of sub-systems: a knowledge base and an inference engine. The knowledge base represents facts about the world, often in some form of subsumption ontology.

In fact the term is often used synonymously with expert systems. The difference is in the view taken to describe the system. Expert system refers to the type of task the system is trying to solve, to replace or aid a human expert in a complex task. Knowledge-based system refers to the architecture of the system, that it represents knowledge explicitly rather than as procedural code. While the earliest knowledge-based systems were almost all expert systems, the same tools and architectures can and have since been used for a whole host of other types of systems. i.e., virtually all expert systems are knowledge-based systems but many knowledge-based systems are not expert systems.The first knowledge-based systems were rule based expert systems. One of the most famous was Mycin a program for medical diagnosis. These early expert systems represented facts about the world as simple assertions in a flat database and used rules to reason about and as a result add to these assertions. Representing knowledge explicitly via rules had several advantages:

Acquisition & Maintenance:- Using rules meant that domain experts could often define and maintain the rules themselves rather than via a programmer.

Explanation:- Representing knowledge explicitly allowed systems to reason about how they came to a conclusion and use this information to explain results to users. For example, to follow the chain of inferences that led to a diagnosis and use these facts to explain the diagnosis.

Reasoning.:-Decoupling the knowledge from the processing of that knowledge enabled general purpose inference engines to be developed. These systems could develop conclusions that followed from a data set that the initial developers may not have even been aware of. As knowledge-based systems became more complex the techniques used to represent the knowledge base became more sophisticated. Rather than representing facts as assertions about data, the knowledge-base became more structured, representing information using similar techniques to object-oriented programming such as hierarchies of classes and subclasses, relations between classes, and behavior of objects. As the knowledge base became more structured reasoning could occur both by independent rules and by interactions within the knowledge base itself. For example, procedures stored as demons on objects could fire and could replicate the chaining behavior of rules Another advancement was the development of special purpose automated reasoning systems called classifiers. Rather than statically declare the subsumption relations in a knowledge-base a classifier allows the developer to simply declare facts about the world and let the classifier deduce the relations. In this way a classifier also can play the role of an inference engine. The most recent advancement of knowledge-based systems has been to adopt the technologies for the development of systems that use the Internet. The Internet often has to deal with complex, unstructured data that can't be relied on to fit a specific data model. The technology of knowledge-based systems and especially the ability to classify objects on demand is ideal for such systems. The model for these kinds of knowledge-based Internet systems is known as the Semantic Web. In order to accomplish feats of apparent intelligence, an expert system

relies on two components: knowledge base and an inference engine. A knowledge base is an organized collection of facts about the system's domain. An inference engine interprets and evaluates the facts in the knowledge base in order to provide an answer. Typical tasks for expert systems involve classification, diagnosis, monitoring, design, scheduling, and planning for specialized endeavors. Facts for a knowledge base must be acquired from human experts through interviews and observations. This knowledge is then usually represented in the form of "if-then" rules (production rules): "If some condition is true, then the following inference can be made (or some action taken)." The knowledge base of a major expert system includes thousands of rules. A probability factor is often attached to the conclusion of each production rule, because the conclusion is not a certainty. For example, a system for the diagnosis of eye diseases might indicate, based on information supplied to it, a 90 percent probability that a person has glaucoma, and it might also list conclusions with lower probabilities. An expert system may display the sequence of rules through which it arrived at its conclusion; tracing this flow helps the user to appraise the credibility of its recommendation and is useful as a learning tool for students.Human experts frequently employ heuristic rules, or "rules of thumb," in addition to simple production rules. For example, a credit manager might know that an applicant with a poor credit history, but a clean record since acquiring a new job, might actually be a good credit risk. Expert systems have incorporated such heuristic rules and increasingly have the ability to learn from experience. Nevertheless, expert systems remain supplements, rather than replacements, for human experts.The knowledge base is a collection of rules or other information structures derived from the human expert. Rules are typically structured as If/Then statements of the form:

IF <antecedent> THEN <consequent>

The antecedent is the condition that must be satisfied.
When the antecedent is satisfied, the rule is triggered and is said to "fire". The consequent is the action that is performed when the rule fires.
Agenda:-When rules are satisfied by the program, they are added to a queue called the agenda. The agenda is an unordered list of all the rules whose antecedents are currently satisfied.Knowledge bases are typically not ordered, because order tends to play very little role in an expert system. Rules may be placed on the agenda in any order, and they may be fired in any order once they are on the agenda.To Create Knowledge Base we use Knowledge Engineering .

Knowledge Engineering:- is the art of designing and building expert systems, and knowledge engineers are its practitioners. Gerald M. Weinberg said of programming in The Psychology of Programming: "'Programming,' -- like 'loving,' -- is a single word that encompasses an infinitude of activities" (Weinberg 1971). Knowledge engineering is the same, perhaps more so. We stated earlier that knowledge engineering is an applied part of the science of artificial intelligence which, in turn, is a part of computer science. Theoretically, then, a knowledge engineer is a computer scientist who knows how to design and implement programs that incorporate artificial intelligence techniques. The nature of knowledge engineering is changing, however, and a new breed of knowledge engineers is emerging. Today there are two ways to build an expert system.:-They can be built from scratch. They built using a piece of development software known as a "tool" or a "shell."

Before we discuss these tools, let's briefly discuss what knowledge engineers do. Though different styles and methods of knowledge engineering exist, the basic approach is the same: a knowledge engineer interviews and observes a human expert or a group of experts and learns what the experts know, and how they reason with their knowledge. The engineer then translates the knowledge into a computer-usable language, and designs an inference engine, a reasoning structure, that uses the knowledge appropriately. He also determines how to integrate the use of uncertain knowledge in the reasoning process, and what kinds of explanation would be useful to the end user. Next, the inference engine and facilities for representing knowledge and for explaining are programmed, and the domain knowledge is entered into the program piece by piece. It may be that the inference engine is not just right; the form of knowledge representation is awkward for the kind of knowledge needed for the task; and the expert might decide the pieces of knowledge are wrong. All these are discovered and modified as the expert system gradually gains competence. The discovery and cumulation of techniques of machine reasoning and knowledge representation is generally the work of artificial intelligence research. The discovery and cumulation of knowledge of a task domain is the province of domain experts. Domain knowledge consists of both formal, textbook knowledge, and experiential knowledge -- the expertise of the experts.Tools, Shells, and Skeletons

Compared to the wide variation in domain knowledge, only a small number of AI methods are known that are useful in expert systems. That is, currently there are only a handful of ways in which to represent knowledge, or to make inferences, or to generate explanations. Thus, systems can be built that contain these useful methods without any domain-specific knowledge. Such systems are known as skeletal systems, shells, or simply AI tools. Building expert systems by using shells offers significant advantages. A system can be built to perform a unique task by entering into a shell all the necessary knowledge about a task domain. The inference engine that applies the knowledge to the task at hand is built into the shell. If the program is not very complicated and if an expert has had some training in the use of a shell, the expert can enter the knowledge himself. Many commercial shells are available today, ranging in size from shells on PCs, to shells on workstations, to shells on large mainframe computers. They range in price from hundreds to tens of thousands of dollars, and range in complexity from simple, forward-chained, rule-based systems requiring two days of training to those so complex that only highly trained knowledge engineers can use them to advantage. They range from general-purpose shells to shells custom-tailored to a class

of tasks, such as financial planning or real-time process control.Although shells simplify programming, in general they don't help with knowledge acquisition. Knowledge acquisition refers to the task of endowing expert systems with knowledge, a task currently performed by knowledge engineers. The choice of reasoning method, or a shell, is important, but it isn't as important as the accumulation of high-quality knowledge. The power of an expert system lies in its store of knowledge about the task domain -- the more knowledge a system is given, the more competent it becomes.

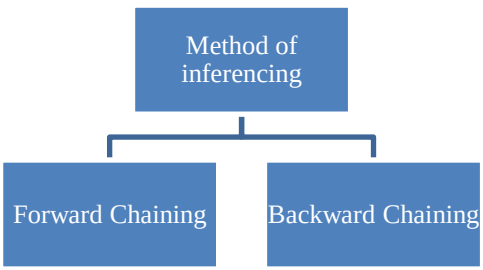
Bricks and Mortar*:*-The fundamental working hypothesis of AI is that intelligent behavior can be precisely described as symbol manipulation and can be modeled with the symbol processing capabilities of the computer.In the late 1950s, special programming languages were invented that facilitate symbol manipulation. The most prominent is called LISP (LISt Processing). Because of its simple elegance and flexibility, most AI research programs are written in LISP, but commercial applications have moved away from LISP.In the early 1970s another AI programming language was invented in France. It is called PROLOG (PROgramming in LOGic). LISP has its roots in one area of mathematics (lambda calculus), PROLOG in another (first-order predicate calculus).PROLOG consists of English-like statements which are facts (assertions), rules (of inference), and questions. Here is an inference rule: "If object-x is part-of object-y then a component-of object-y is object-x."Programs written in PROLOG have behavior similar to rule-based systems written in LISP. PROLOG, however, did not immediately become a language of choice for AI programmers. In the early 1980s it was given impetus with the announcement by the Japanese that they would use a logic programming language for the Fifth Generation Computing Systems (FGCS) Project. A variety of logic-based programming languages have since arisen, and the term prolog has become generic.

User Interface: A user interface is the method by which the expert system interacts with a user. These can be through dialog boxes, command prompts, forms, or other input methods. Some expert systems interact with other computer applications, and do not interact directly with a human. In these cases, the expert system will have an interaction mechanism for transactions with the other application, and will not have a user interface. OR we can say that the user interface is the means of communication between a user and the expert systems problem-solving processes. A good expert system is not very useful unless it has an effective interface. It has to be able to accept the queries or instructions in a form that the user enters and translate them into working instructions for the rest of the system. It also has to be able to translate the answers, produced by the system, into a form that the user can understand. Careful attention should be given to the screen design in order to make the expert system appear 'friendly' to the user. The acceptability of an expert system depends to a great extent on the quality of the user interface. The user can enter commands, and respond to questions. The system responds to commands, and asks questions during the inferencing process.More advanced interfaces make heavy use of pop-up menus, windows, mice.. If the machine supports it, graphics can also be a powerful tool for communicating with the user. This is especially true for the development interface which is used by the knowledge engineer in building the system.

Inference Engine:- The inference engine is the main processing element of the expert system. The inference engine chooses rules from the agenda to fire. If there are no rules on the agenda, the inference engine must obtain information from the user in order to add more rules to the agenda. It makes use of knowledge base, in order to draw conclusions for situations. It is responsible for gathering the information from the user, by asking various questions and applying it wherever necessary.seeks information and relationships from the knowledge base and to provide answers,predictions and suggestions the way a human expert would.Where the user interacts with the expert system. In other words where questions are asked, and advice is produced. As well as the advice that is output, the user interface can output the justification features of an expert system. This is either How justification - where the system justifies its reasoning for providing a piece of advice or Why justification - where the system justifies why a particular question is being asked.Justification allows the user piece of mind about why a question is asked or a piece of advice is provided, and can increase their confidence in taking such advice. It also makes it easier for the programmer of the system to ensure that it works correctly as it will flag up areas where the expert system provides advice that is not intended by the programmer.

The Shell or Inference Engine: The inference engine is the program that locates the appropriate knowledge in the knowledge base, and infers new knowledge by applying logical processing and problem-solving strategies. This applies the facts to the rules and determines the questions to be asked of the user in the user interface and in which order to ask them. This is the 'invisible' part of the expert system, which is active during a consultation of the system (when the user chooses to run the program). An expert system can use 2 different methods of inferencing:-

Forward Chaining and Backward Chaining.



Backward Chaining*:*-A Backward Chaining system is also Known as goal driven system works with the system assuming a hypothesis of what the likely outcome will be, and the system then works backwards to collect the evidence that would support this conclusion. Expert systems used for planning often use backward chaining.

Forward Chaining:- A Forward Chaining expert system also Known a data driven system simply gathers facts (like a detective at the scene of a crime) until enough evidence is collected that points to an outcome. Forward chaining is often used in expert systems for diagnosis, advise and classification, although the size and complexity of the system can play a part in deciding which method of inferencing to use.

Architecture:- The logic that an inference engine uses is typically represented as IF-THEN rules. The general format of such rules is IF <logical expression> THEN <logical expression>. Prior to the development of expert systems and inference engines artificial intelligence researchers focused on more powerful theorem prove environments that offered much fuller implementations of First Order Logic. For example, general statements that included universal quantification (for all X some statement is true) and existential quantification (there exists some X such that some statement is true). What researchers discovered is that the power of these theorem proving environments was also their drawback. It was far too easy to create logical expressions that could take an indeterminate or even infinite time to terminate. For example, it is common in universal quantification to make statements over an infinite set such as the set of all natural numbers. Such statements are perfectly reasonable and even required in mathematical proofs but when included in an automated theorem prover executing on a computer may cause the computer to fall into an infinite loop. Focusing on IF-THEN statements (what logicians call Modus Ponens) still gave developers a very powerful general mechanism to represent logic but one that could be used efficiently with computational resources. What is more there is some psychological research that indicates humans also tend to favor IF-THEN representations when storing complex knowledge.[A simple example of Modus Ponens often used in introductory logic books is "If you are human then you are mortal". This can be represented in pseudocode as: Rule1: Human(x) => Mortal(x)

Inference Engine:-In order to execute a rule-based expert system using the method of forward chaining we merely need to fire (or execute) actions whenever they appear on the action list of a rule whose conditions are true. This involves assigning values to attributes, evaluating conditions, and checking to see if all of the conditions in a rule are satisfied. A general algorithm for this might be: while values for attributes remain to be input read value and assign to attribute evaluate conditions fire rules whose conditions are satisfied

Several points about this require consideration. First, some conflict resolution strategy needs to be employed in order to decide which rules are fired first.Our method is to fire the rule which the system designer defined first. Also, we wish to cut down on computational time. To do this we must not do anything which does not absolutely need to be done. This means that conditions are only evaluated at the time they might change and that rules are checked (to see if all of their conditions are satisfied) only when they might be ready to be fired, not before. We shall do this as attributes are assigned values and shall only consider rules and conditions affected by the new attribute assignment.

Knowledge Acquisition:- The Knowledge Acquisition Facility is an analysis and design tool used to model tacit knowledge to be incorporated into an Expert System, Knowledge Base System or as part of an Knowledge Management System. This software suite includes 3 products, Knowledge Domain Analyzer, Knowledge Modeler (Utilizing an expanded UML notation), and the Knowledge Acquisition Unified Framework (The Eclipse Base Knowledge Acquisition Process).

Knowledge Domain Analyzer - This module incorporates rules and decision trees designed to determine if the domain under discussion is suitable for tacit knowledge capture and modeling. This module will interact with the user asking a series of questions, performing the analysis and reporting back on the details of its decision.

Knowledge Modeler - This module will incorporate an expanded Unified Modeling Language (UML) notation and incorporate knowledge modeling rules that will assist the user (Analyst/Designer) in the modeling of tacit knowledge.

Knowledge Acquisition Unified Framework (KAUF) - This software is a framework for identifying, capturing and cataloging knowledge by addressing specific needs of the knowledge engineer during the knowledge acquisition process. These needs include the capability to decompose the knowledge acquisition task into manageable subtasks, focus on a representation of expertise that is natural to domain experts, and recognize the patterns in knowledge and to resolve conflict when aspects of knowledge of a particular domain become uncertain. The Knowledge Acquisition Facility enables a Knowledge Engineer to successfully capture, apply and validate tacit knowledge. The Knowledge Acquisition Facility is constructed to stand alone and can be integrated to utilize various vendor products (i.e. IBM, Ilog, Paradigm Pop, Eclipse). As an iPad/iPhone Application developer we bring knowledge management software to these premier platforms from Apple. Currently our focus is in the areas of People Productivity, Emergency Response, and Medical applications.

Methods for knowledge elicitation *:*-On its first processing level our model makes use of three different knowledge elicitation method

Interview :-One of the most important strategies of knowledge engineering is the interview. Grover (1983) distinguishes four different interview techniques for rule acquisition:

(1) **Forward scenario simulation**:-An applicational situation within a problem domain is selected and investigated under laboratory conditions. The expert reports on the relevant terms and concepts and describes the steps in problem-solving, i.e. his or her own reasoning to achieve a goal.

(2) **Goal decomposition** :-The knowledge engineer divides the overall problem into subgoals and asks the expert to describe paths for achieving the subgoals.

(3) **Procedural simulation** :-Grover (1983) uses this umbrella term for protocol analysis. In his opinion

controlling interventions by the knowledge engineer are absolutely necessary.

(4) **Pure reclassification** :-Expert statements are further differentiated and classified into specific objects and relations between objects by means of a dialogue between knowledge engineer and expert. As a result of the interview, object-relations may be reclassified and new taxonomic relations eventually discovered. An interview techniques not mentioned in Grover's classification .

(5) **Laddering**:-The expert is asked to name important concepts of the problem domain in question. These concepts are then used as basis for the interview to follow. Especially supertypes and instances of generic concepts are inquired about,allowing the derivation of a taxonomic structure.The system must liaise with people in order to gain knowledge and the people must be specialised in the appropriate area of activity. For example medical doctors, geologists or chemists. The knowledge engineer acts as an intermediary between the specialist and the expert system. Typical of the information that must be gleaned is vocabulary or jargon, general concepts and facts, problems that commonly arise, the solutions to the problems that occur and skills for solving particular problems. This process of picking the brain of an expert is a specialised form of data capture and makes use of interview techniques. The knowledge engineer is also responsible for the self consistency of the data loaded. Thus a number of specific tests have to be performed to ensure that the conclusions reached are sensible.

Consultation

The system is in this mode when a user is interacting with it. The user interacts by entering data in English and the sytem responds by using its inference engine to carry out the reasoning involved in deriving an answer to the questions posed by the user. The role of the inference engine is to search for facts that match the condition part of the productions that match the action part of the question. There is scope for the use of heuristics in this role. The following example is hypothetical but illustrates the action of typical systems.

System: What is the patient's name?
User: George Washington
System: What is the patient's age?
User: 252 years
System: Is the patient male or female?
User: Male
System: What are the symptoms?
User: Headache dizziness upset stomach high fever
System: When did these symptoms first appear?
User: December 13,1799
System: I conclude that the patient has a viral infection, my recommendation is that he should take two aspirin, drink plenty of fluid, get lots of rest and call me in the morning.George Washington died the next day of a viral infection no antibiotics.

Explanation

This mode allows the system to explain its conclusions and its reasoning process. This ability comes from the AND/OR trees created during the production system reasoning process. As a result most expert systems can answer the following why and how questions

why was a given fact used?
why was a given fact not used?
how was a given conclusion reached?
how was it that another conclusion was not reached?

Explanation Facility:-Most expert systems have explanation facilities that allow the user to ask *why* it asked some question, and *how* it reached some conclusion. These questions are answered by referring to the system goals, the rules being used, and any existing problem data. To illustrate the sorts of facilities commonly provided, we'll elaborate on our simple example above, starting with a simple example dialogue involving *why* and *how* questions and explanations.

System: Is there gas in the fuel tank?
User: Yes.
System: Is it true that the engine turns over?
User: No.
System Is it true that the lights come on?
User: Why?
System: I am trying to prove: problem with battery.
It has been established that it is not the case that the engine turns over.
Therefore if it is not the case that the lights come on then there is a problem
with battery.
Is it true that the lights come on?
User: No.
System: I conclude that there is a problem with the battery.
User: How?
System: This follows from Rule 2:
IF NOT engine_turns_overAND ...
NOT engine_turns_overwas given by the user.
NOT lights_come_onwas given by the user.

Conclusion

In this paper, we have proposed the architecture of the expert system .In this we categorize the expert system so they can used for different application .Various methods as well as rules are used to explain the component of expert system.

References

BW77 Bobrow, D. G. and Winograd, T. (1977)., "An Overview of KRL a Knowledge Representation Language," *Cognitive Science*

BM77 Boyer, R. S. and Moore, J. S. (1977). "A Fast String Searching Algorithm," *Communications of the ACM*, 20

DBS77 Davis, R., Buchanan, B., and Shortliffe, E. H. (1977) "Production Rules as a Representation for a Knowledge-Based Consultation Program," *Artificial Intelligence* .8.

DGH79 Duda, R. O., Gaschnig, J. G., and Hart, P. E. 1979, "Model Design in the Prospector Consultant System for Mineral Exploration," in *Expert Systems in the Microelectronic Age,* ed. D. Michie, Edinburgh University Press, Edinburgh.

Fo82 Forgy, C. L. (1982)., "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem," *Artificial Intelligence* 19

FM75 Fleisher, H. and Maissel, L. I. (1975)., "An Introduction to Array Logic ," *IBM J. Res. And Dev.*

Ga88 Gajski, D. D. ed (1977.), *Silicon Compilation* Addison-Wesley, Reading, Massachusetts

Gr87 Griffin, N.(1988) "A Fast Architecture for Rule-Based Systems," *MS Thesis, University of Kentucky,*, 1987. Also in *Proc. Southeastern Regional ACM Conf.*

IL89 Igarashi, Y. and Lewis, F. D. ."Expert Systems in Silicon," in preparation.

Mu82 Muroga, S. (1977),*VLSI System Design*, John Wiley & Sons, New York,.

Qu68 Quillian, M. R. (1968), "Semantic Memory," in *Semantic Information Processing*, ed. M. Minsky, MIT Press, Cambridge, Mass.

MS81 van Melle, W., Shortliffe, E. H., and Buchanan, B. G. (1981)., "EMYCIN: A Domain- Independent System that Aids in Constructing Knowledge-Based Consultation Programs," *Machine Intelligence* 3

WK81 Weiss, S. M. and Kulikowski, C. A. (1981)., "EXPERT Consultation Systems: The Expert and Casnet Projects," *Machine Intelligence* 3

WiT75 Winograd, T.(1975), "Frame Representation and the Declarative/Procedural Controversy," in *Representation and Understanding: Studies in Cognitive Science*, ed. A. Collins, Academic Press, New York.

BW77 Bobrow, D. G. and Winograd, T. (1977)., "An Overview of KRL a Knowledge Representation Language," *Cognitive Science*

BM77 Boyer, R. S. and Moore, J. S. (1977). "A Fast String Searching Algorithm," *Communications of the ACM*, 20

DBS77 Davis, R., Buchanan, B., and Shortliffe, E. H. ((1977).) "Production Rules as a Representation for a Knowledge-Based Consultation Program," *Artificial Intelligence* .8

DGH79 Duda, R. O., Gaschnig, J. G., and Hart, P. E. (1979), "Model Design in the Prospector Consultant System for Mineral Exploration," in *Expert Systems in the Microelectronic Age*, ed. D. Michie, Edinburgh University Press, Edinburgh.

Fo82 Forgy, C. L. (1982)., "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem," *Artificial Intelligence* 19

FM75 Fleisher, H. and Maissel, L. I.(. (1975).), "An Introduction to Array Logic ," *IBM J. Res. And Dev*

Ga88 Gajski, D. D. ed (1977.), *Silicon Compilation* Addison-Wesley, Reading, Massachusetts,

Gr87 Griffin, N. (1987) "A Fast Architecture for Rule-Based Systems," *MS Thesis, University of Kentucky,*. Also in *Proc. 1988 Southeastern Regional ACM Conf.*

IL89 Igarashi, Y. and Lewis, F. D. ."Expert Systems in Silicon," in preparation.

Mu82 Muroga, S.,*VLSI System Design* (1977.), John Wiley & Sons, New York,

Qu68 Quillian, M. R. (1968), "Semantic Memory," in *Semantic Information Processing*, ed. M. Minsky, MIT Press, Cambridge, Mass.,.

vMS81 van Melle, W., Shortliffe, E. H., and Buchanan, B. G. ((1981).), "EMYCIN: A Domain- Independent System that Aids in Constructing Knowledge-Based Consultation Programs," *Machine Intelligence* 3

WK81 Weiss, S. M. and Kulikowski, C. A. (1981)., "Expert Consultation Systems: The Expert and Casnet Projects," *Machine Intelligence* 3

WiT75 Winograd, T. (1975.), "Frame Representation and the Declarative/Procedural Controversy," in *Representation and Understanding: Studies in Cognitive Science*, ed. A. Collins, Academic Press, New York