Artificial Intelligence and Expert Systems

Angulakshmi S
PG Student
Department of Information Technology
PSNA College of Engineering and Technology,Dindigul
angukamu@gmail.com

Abstract:-Artificial Intelligence is a branch of science, which deals with helping machines, finds solutions to complex problems in a more human like fashion. Artificial Intelligence's scientific goal is to understand intelligence by building computer programs that exhibit intelligent behavior.

This paper presents some back ground and potential of Artificial Intelligence and its implementation in various fields. We discuss issues that have not been studied in detail within the expert systems setting, yet are crucial for developing theoretical methods and computational architectures for automated reasons. The tools that are required to construct expert systems are discussed in detail.

Keywords: AI, Expert System, Black board systems, Bayesian networks,System Shell

1.INTRODUCTION

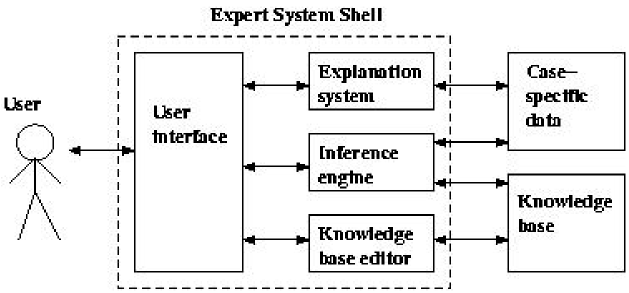
Artificial intelligence is the study of ideas to bring into being machines that respond to stimulation consistent with traditional responses from humans, given the human capacity for contemplation, judgment and intention. Each such machine should engage in critical appraisal and selection of differing opinions within itself. Produced by human skill and labor, these machines should conduct themselves in agreement with life, spirit and sensitivity, though in reality, they are imitations.

An AI program that models the nuances of the human thought process or solves complicated real-world problems con be complex. Languages for building AI programs have the capacity to that take care of low level computing processes, thus allowing the developer to focus on requisite complexity. AI programs work with concepts expressed in words, phrases, or sentences. Therefore, the ability to handle symbolic data is an important feature .To develop an AI system, a programmer tries numerous ways of implementing each constituent function. Rather than go through a lengthy edit-compile-debug cycle to test each type or quantity of data that follows through a program, so they must adapt to fly. A language that incorporates flexible data structures allows this happen in an easy natural way.

2.EXPERT SYSTEMS

An expert system is an interactive computer-based decision tool that uses both facts and heuristics to solve difficult decision problems based on knowledge acquired from an expert. By definition, an expert system is a computer program that simulates the thought process of a human expert to solve complex decision problems in a specific domain. The growth of expert systems is expected to continue for several years. With the continuing growth, many new and exciting applications will emerge. An expert system operates as an interactive system that responds to questions, asks for clarification, makes recommendations, and generally aids the decision-making process.

Expert systems provide expert advice and



guidance in a wide variety of activities, from computer diagnosis to delicate medical surgery.

Complex decisions involve intricate combination of factual and heuristic knowledge. In order for the computer to be able to retrieve and effectively use heuristic knowledge, the knowledge must be organized in an easily accessible format that distinguishes among data, knowledge, and control structures. For this reason, expert systems are organized in four distinct levels:

3. THE ARCHITECTURE OF EXPERT SYSTEMS

1. Knowledge base- consists of problem solving

2. Working memory- refers to task-specific data for the problem under consideration.

3. Inference engine- is a generic control mechanism that applies the axiomatic knowledge in the knowledge base to the task-specific data to arrive at some solution or conclusion.

4.User interface - the code that controls the dialog between the user and the system.

3.1 Knowledge base

A knowledge base is the nucleus of the expert system structure. The knowledge base may be a specific diagnostic knowledge base compiled by a consulting firm, and the end user may supply the problem data. Knowledge base is not a database. The traditional data base environment deals with data that have a static relationship between the elements in the problem domain. Knowledge engineers, who translate the knowledge of real human experts into rules and strategies, create it. These rules and strategies can change depending on the prevailing problem scenario. The knowledge base provides the expert system with the capability to recommend directions for user inquiry. It is usually stored in terms of if–then rules.

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.

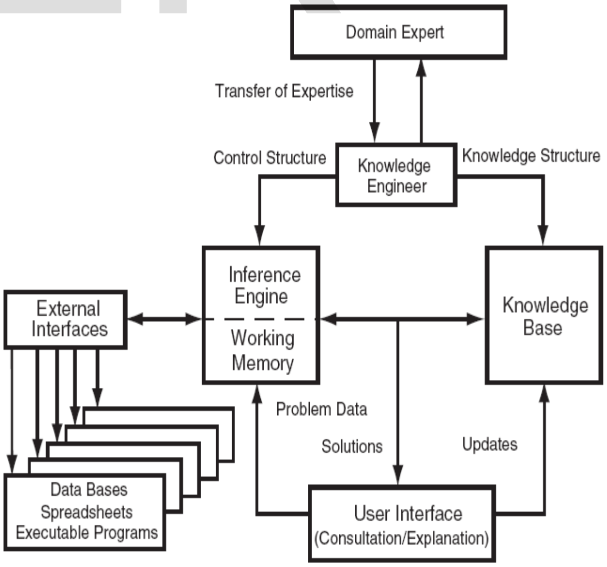
Heuristic knowledge is the less rigorous, more experiential, more judgmental knowledge of performance. In contrast to factual knowledge, heuristic knowledge is rarely discussed, and is largely individualistic. It is the knowledge of good practice, good judgment, and plausible reasoning in the field. It is the knowledge that underlies the "art of good guessing."

3.2 Inference engine

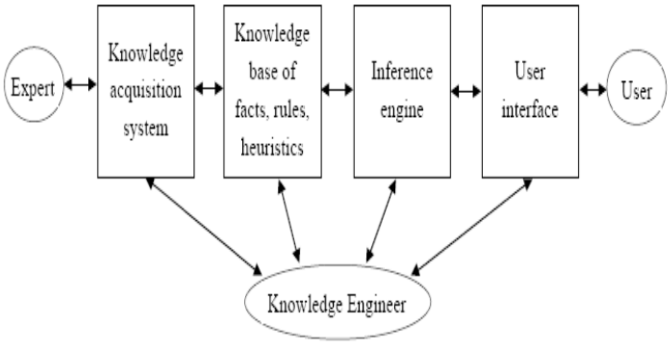
The basic functions of inference engine are:

1. Match the premise patterns of the rules against elements in the working memory. Generally the rules will be domain knowledge built into the system, and the working memory will contain the case based facts entered into the system, plus any new facts that have been derived from them.

2. If there is more than one rule that can be applied, use a conflict resolution strategy to choose one to apply. Stop if no further rules are applicable.



3. Activate the chosen rule, which generally means adding/deleting an item to/from working


Expert
Knowledge acquisition system
Knowledge base of facts, rules, heuristics
Inference engine
User interface
User
Knowledge Engineer

memory. Stop if a terminating condition is reached, or return to step 1.

3.3 User interface

The Expert System user interface usually comprises of two basic components:

✓ **The Interviewer Component**

This controls the dialog with the user and/or allows any measured data to be read into the system. For example, it might ask the user a series of questions, or it might read a file containing a series of test results.

✓ **The Explanation Component**

This gives the system's solution, and also makes the system's operation transparent by providing the user with information about its reasoning process. For example, it might output the conclusion, and also the sequence of rules that was used to come to that conclusion. It might instead explain why it could not reach a conclusion. So that is how we go about building expert systems. In the next two weeks we shall see how they can handle uncertainty and be improved by incorporating machine learning.

3.4 Explanation system

Almost all expert systems also have an explanation subsystem, which allows the program to explain its reasoning to the user.

3.5 Knowledge base editor

Some systems also have a knowledge base editor, which help the expert or knowledge engineer to easily update and check the knowledge base.

3.6 Working memory

The working memory represents relevant data for the current problem being solved.

4.PROPOSED WORK

4.1 Knowledge engineering

It is the art of designing and building expert systems, and knowledge engineers are its practitioners. As stated earlier that knowledge engineering is an applied part of the science of artificial intelligence, which, in turn, is a part of computer science. Today there are two ways to build an expert system. They can be built from scratch, or 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.

4.2Programming Languages

Expert systems are typically written in special programming languages. The use of languages like LISP and PROLOG in the development of an expert system simplifies the coding process. The major advantage of these languages, as compared to conventional programming languages, is the simplicity of the addition, elimination, or substitution of new rules and memory management capabilities. The programming languages used for expert systems tend to operate in a manner similar to ordinary conversation. We usually state the premise of a problem in the form of a question; with actions being stated much as when we verbally answer the question, that is, in a ‘‘natural language’’ format. If, during or after a consultation, an expert system determines that a piece of its data or knowledge base is incorrect or is no longer applicable because the problem environment has changed, it should be able to update the knowledge base accordingly. This capability would allow the expert system to converse in a natural language format with either the developers or users. Some of the distinguishing characteristics of programming languages needed for expert systems work are:

- Efficient mix of integer and real variables
- Good memory-management procedures
- Extensive data-manipulation routines
- Incremental compilation
- Tagged memory architecture
- Optimization of the systems environment
- Efficient search procedures

4.3 Expert System Shell

An expert system shell is a program that provides the framework required for an Expert System, but with no knowledge Base. The shell provides an inference engine, perhaps a user interface for providing knowledge or some means of reading data in from files. Some shells are self-contained, while others can be extended by using other programming languages. Indeed, some are effectively programming languages have their own right! The advantage of using a shell is that you can focus on solving the problem at hand rather than trying to make an Expert System to scratch. The disadvantage is that you’re stuck with the guts of the shell.

4.4 Types of expert systems

There are various types of expert system technology available. What to use depends upon the nature of the problem and software and what is easiest! Seriously, inmost simple applications of Expert systems the choice e of technology is not as important as one might think.

The basic types of system are:

- Decision trees
- Forward chaining
- Back ward chaining
- State machines
- Bayesian networks
- Black board systems
- Case based reasoning

Any of these basic technologies can be implemented from a suitable shell or from scratch. In some other cases technologies such as fuzzy logic or Neural networks can be integrated in to the Expert system to create amore efficient Expert.

4.5 Decision tree

This technology is best suited in creating Expert systems whose primary role is in diagnosis of problems. In a tree system the user is prompted with a question and each question having a number of responses, which indicate the next question to ask. Eventually the tree is organized to a point where an answer is given by the system or it gives up and effectively says ‘I don’t know’. Behind the scenes the knowledge base is a series of questions, responses and questions to ask based on those responses and the ‘end points where the answer is represented.

4.6 Forward chaining

A forward chaining expert system starts from the symptoms and then runs through its knowledge base until it gets to an answer. The difference between this system and the decision tree method above is that in forward chaining the input data is presented at the starting of the inference engine runs through it's rules in an arbitrary order until no further changes take place in the internal variables of the system reached 'steady state' this may take several runs through the rules, as it is possible that one rule firing off may influence the behavior of another rule that's already been run. so the rules are all run again until no more rules are executed

4.7 Backward chaining

A back ward chaining expert system starts with a possible result and then reviews the inputs available to see if the evidence matches. As you find an input that DOESN'T fit the proposed result, that result is disregarded and a different potential solution selected. If your knowledge base is large, taking this approach may be faster than the sequential examination of all inputs and rules as required by the tree or forward chaining mechanism

4.8 State machine

A state machine is a software 'object' that can hold a number of different 'states', each state being represented by a particular combination values stored with in the object. How the object behaves, and switches between the states, depends up on the current state it's in. Each state can have different inputs and output values to another state, so one state may in to account the value and another state may disregard it.

4.9 Case based reasoning

Here the system requires a knowledge base made up of previous 'cases or instances of the problem, with solution that was found in a result that took place. Rather than creating a set of rules, you just write an inference Engine that can look for similarities between previous situations ad the current one.

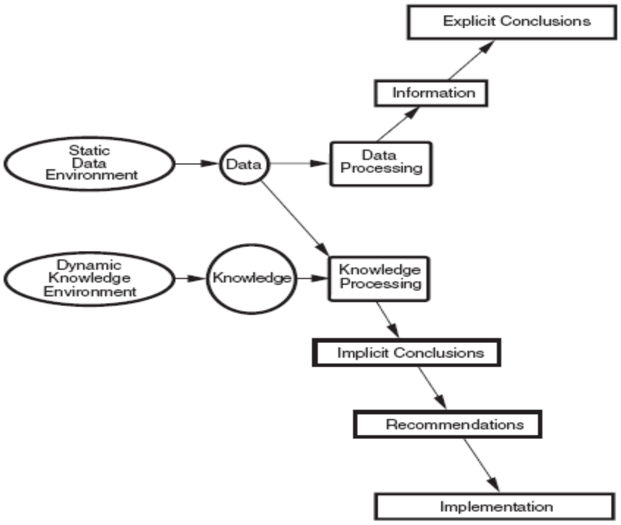
4.10 Bayesian networks

A problem with simpler Expert Systems is that if they cannot find answer that matches a set of circumstances in the inference engine and in Knowledge Base then they give no answer. A system built around a Bayesian Network will give a best-fit answer with probabilities attained. The knowledge base for such a system consists of a table of inputs and out pts with the probability that a particular input will contribute to a particular output

4.11 Black board systems

Black board systems work by having lots of small components that identify particular events or results based on specific inputs. Each component then communicates its result to a controlling system, which is also receiving data from other components. The individual components might be fully fledged Expert Systems in their own right, or other technologies such as Neural Networks

5. TRANSITION FROM DATA PROCESSING TO KNOWLEDGE PROCESSING



What data has been to the previous generations of computing, knowledge is to the present generation of computing. Expert systems represent a revolutionary transition from the traditional data processing to knowledge processing. Figure illustrates the relationships between the procedures for data processing and knowledge processing to make decisions. In traditional data processing the decision maker obtains the information generated and performs an explicit analysis of the information before making his or her decision. In an expert system knowledge is processed by using available data as the processing fuel. Conclusions are reached and recommendations are derived implicitly. The expert system offers the recommendation to the decision maker, who makes the final decision and implements it as appropriate. Conventional data can now be manipulated to work with durable knowledge, which can be processed to generate timely information, which is then used to enhance human decisions.

5.1 The Need for Expert Systems

Expert systems are necessitated by the limitations associated with conventional human decision-making processes, including:

1. Human expertise is very scarce.
2. Humans get tired from physical or mental workload.
3. Humans forget crucial details of a problem.
4. Humans are inconsistent in their day-to-day decisions.
5. Humans have limited working memory.
6. Humans are unable to comprehend large amounts of data quickly.
7. Humans are unable to retain large amounts of data in memory.
8. Humans are slow in recalling information stored in memory.
9. Humans are subject to deliberate or inadvertent bias in their actions.
10. Humans can deliberately avoid decision responsibilities.
11. Humans lie, hide, and die.

Coupled with these human limitations are the weaknesses inherent in conventional programming and traditional decision-support tools. Despite the mechanistic power of computers, they have certain limitations that impair their effectiveness in implementing human-like decision processes. Conventional programs:

1. Are algorithmic in nature and depend only on raw machine power
2. Depend on facts that may be difficult to obtain
3. Do not make use of the effective heuristic approaches used by human experts
4. Are not easily adaptable to changing problem environments
5. Seek explicit and factual solutions that may not be possible

5.2 The Applications of Expert Systems

An expert system may be viewed as a computer simulation of a human expert. Expert systems are an emerging technology with many areas for potential applications. Past applications range from MYCIN, used in the medical field to diagnose infectious blood diseases, to XCON, used to configure computer systems. These expert systems have proven to be quite successful. Most applications of expert systems will fall into one of the following categories:

5.3 Diagnosis and Troubleshooting of Devices and Systems of All Kinds

This class comprises systems that deduce faults and suggest corrective actions for a malfunctioning device or process. Medical diagnosis was one of the first knowledge areas to which ES technology was applied (for example, see Shortliffe 1976), but diagnosis of engineered systems quickly surpassed medical diagnosis. There are probably more diagnostic applications of ES than any other type. The diagnostic problem can be stated in the abstract as: given the evidence presenting itself, what is the underlying problem/reason/cause?

5.4 Planning and Scheduling

Systems that fall into this class analyze a set of one or more potentially complex and interacting goals in order to determine a set of actions to achieve those goals, and/or provide a detailed temporal ordering of those actions, taking into account personnel, materiel, and other constraints. This class has great commercial potential, which has been recognized. Examples involve airline scheduling of flights, personnel, and gates; manufacturing job-shop scheduling; and manufacturing process planning.

5.5 Configuration of Manufactured Objects from Subassemblies

Configuration, whereby a solution to a problem is synthesized from a given set of elements related by a set of constraints, is historically one of the most important of expert system applications. Configuration applications were pioneered by computer companies as a means of facilitating the manufacture of semi-custom minicomputers (McDermott 1981). The technique has found its way into use in many different industries, for example, modular home building, manufacturing, and other problems involving complex engineering design and manufacturing.

5.6 Financial Decision Making

The financial services industry has been a vigorous user of expert system techniques. Advisory programs have been created to assist bankers in determining whether to make loans to businesses and individuals. Insurance companies have used expert systems to assess the risk presented by the customer and to determine a price for the insurance. A typical application in the financial markets is in foreign exchange trading.

5.7 Knowledge Publishing

This is a relatively new, but also potentially explosive area. The primary function of the expert system is to deliver knowledge that is relevant to the user's problem, in the context of the user's problem. The two most widely distributed expert systems in the world are in this category. The first is an advisor, which counsels a user on appropriate grammatical usage in a text. The second is a tax advisor that accompanies a tax preparation program and advises the user on tax strategy, tactics, and individual tax policy.

5.8 Process Monitoring and Control

Systems falling in this class analyze real-time data from physical devices with the goal of noticing anomalies, predicting trends, and controlling for both optimality and failure correction. Examples of real-time systems that actively monitor processes can be found in the steel making and oil refining industries.

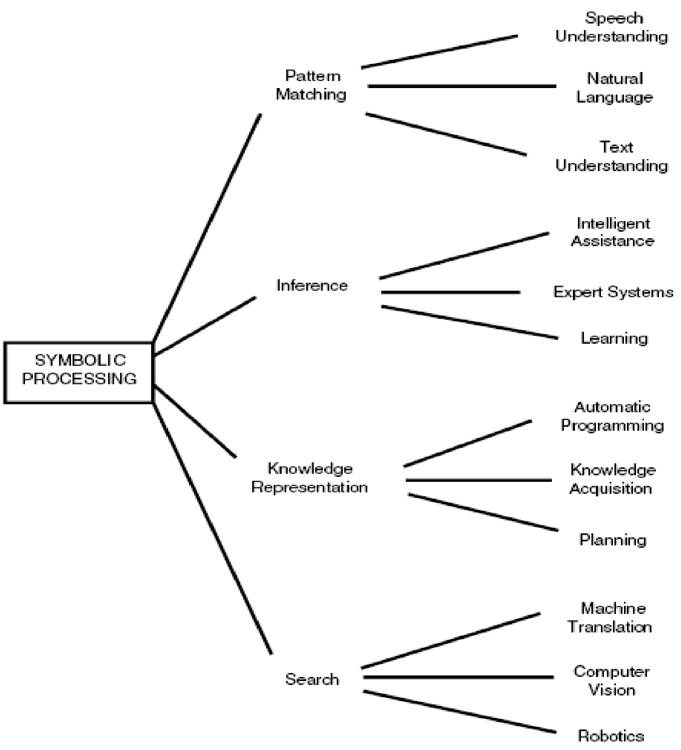
6. DESIGN AND MANUFACTURING

These systems assist in the design of physical devices and processes, ranging from high-level conceptual design of abstract entities all the way to factory floor configuration of manufacturing processes.

Applications that are computational or deterministic in nature are not good candidates for expert systems. Traditional decision support systems such as spreadsheets are very mechanistic in the way they solve problems. They operate under mathematical and Boolean operators in their execution and arrive at one and only one static solution for a given set of data. Calculation intensive applications with very exacting requirements are better handled by traditional decision support tools or conventional programming. The best application candidates for expert systems are those dealing with expert heuristics for solving problems.

6.1 Application Roadmap

The symbolic processing capabilities of AI technology lead to many potential applications in engineering and manufacturing. With the increasing sophistication of AI techniques, analysts are now able to use innovative methods to provide viable solutions to complex problems in everyday applications. Figure presents a structural representation of the application paths for artificial intelligence and expert systems.



6.2 Benefits of Expert Systems

Expert systems offer an environment where the good capabilities of humans and the power of computers can be incorporated to overcome many of the limitations discussed in the previous section. Expert systems:

1. Increase the probability, frequency, and consistency of making good decisions
2. Help distribute human expertise
3. Facilitate real-time, low-cost expert-level decisions by the non-expert
4. Enhance the utilization of most of the available data
5. Permit objectivity by weighing evidence without bias and without regard for the user's personal and emotional reactions
6. Permit dynamism through modularity of structure
7. Free up the mind and time of the human expert to enable him or her to concentrate on more creative activities
8. Encourage investigations into the subtle areas of a problem

7. CONCLUSIONS

However a good expert system is expected to grow as it learns from user feedback. Feedback is incorporated into the knowledge base as appropriate to make the expert system smarter. The dynamism of the application environment for expert systems is based on the individual dynamism of the components. This can be classified as follows:

• *Most dynamic: Working memory*. The contents of the working memory, sometimes called the data structure, changes with each problem situation. Consequently, it is the most dynamic component of an expert system, assuming, of course, that it is kept current.

• *Moderately dynamic: Knowledge base*. The knowledge base need not change unless a new piece of information arises that indicates a change in the problem solution procedure. Changes in the knowledge base should be carefully evaluated before being implemented. In effect, changes should not be based on just one consultation experience. For example, a rule that is found to be irrelevant less than one problem situation may turn out to be crucial in solving other problems.

• *Least dynamic: Inference engine*. Because of the strict control and coding structure of an inference engine, changes are made only if absolutely necessary to correct a bug or enhance the inferential process. Commercial inference engines, in particular, change only at the discretion of the developer. Since frequent updates can be disruptive and costly to clients, most commercial software developers try to minimize the frequency of updates. Artificial intelligence has a long way to go yet before it can achieve its goals, but the discoveries made by research in this area justify its continuation. There are intelligent techniques that have been developed though and it is doubtless that these will continue to be developed and similar new discoveries made. However, true intelligence in machines appears to be, for now at least, beyond our reach. Only time will tell whether this remains to be so.

REFERENCES:

- www.pcai.com
- www.acm.org
- www.alanturing.net
- www.oniformation.com
- www.exampleessays.com