ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 6 June, 2013 Page No. 1914-1918

A REVIEW ON KNOWLEDGE-BASED EXPERT SYSTEM

Inderpal Singh
Computer Science and Engineering Department
DAV Institute of Engineering and Technology
Jalandhar
Er.inderpal13@gmail.com

Abstract-- ***A Knowledge-based expert system use human knowledge to solve problems that normally would require human intelligence. Expert systems are designed to carry the intelligence and information found in the intellect of experts and provide this knowledge to other members of the organization for problem solving purposes. With the growing importance of human resource management and increasing size of the organizations, maintenance of employee related data and generating appropriate reports are the crucial aspects of any organization. Therefore more and more organizations are adopting computer based human resource management systems (HRMS). This paper explains the architecture, knowledge representation techniques of the knowledge-based expert system.***

Keywords-Knowledge-based Expert System, Knowledge Base, Inference Engine

Introduction: What is an Expert System?

An expert system is a computer program that represents and reasons with knowledge of some specialist subject with a view to solving problems or giving advice.

To solve expert-level problems, expert systems will need efficient access to a substantial domain knowledge base, and a reasoning mechanism to apply the knowledge to the problems they are given. Usually they will also need to be able to explain, to the users who rely on them, how they have reached their decisions.

They will generally build upon the ideas of knowledge representation, production rules, search, and so on that we have already covered.

Often we use an expert system shell, which is an existing knowledge independent framework into which domain knowledge can be inserted to produce a working expert system. We can thus avoid having to program each new system from scratch.

Typical Tasks for Expert Systems

There are no fundamental limits on what problem domains an expert system can be built to deal with. Some typical existing expert system tasks include:

1. The interpretation of data Such as sonar data or geophysical measurements

2. Diagnosis of malfunctions Such as equipment faults or human diseases

3. Structural analysis or configuration of complex objects Such as chemical compounds or computer systems

4. Planning sequences of actions Such as might be performed by robots

5. Predicting the future Such as weather, share prices, exchange rates

Characteristics of Expert Systems

Expert systems can be distinguished from conventional computer systems in that:

1. They simulate human reasoning about the problem domain, rather than simulating the domain itself.

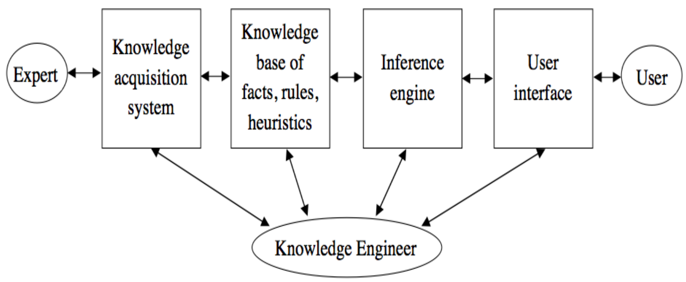
2. They perform reasoning over representations of human knowledge, in addition to doing numerical calculations or data retrieval. They have corresponding distinct modules referred to as the inference engine and the knowledge base.

3. Problems tend to be solved using heuristics (rules of thumb) or approximate methods or probabilistic methods, which, unlike algorithmic solutions, are not guaranteed to result in a correct or optimal solution.

4. They usually have to provide explanations and justifications of their solutions or recommendations in order to convince the user that their reasoning is correct.

The Architecture of Expert Systems

The process of building expert systems is often called knowledge engineering. The knowledge engineer is involved with all components of an expert system:



ARCHITECTURE OF EXPERT SYSTEM

Building expert systems is generally an iterative process. The components and their interaction will be refined over the course of numerous meetings of the knowledge engineer with the experts and users. We shall look in turn at the various components.

Knowledge Acquisition

The knowledge acquisition component allows the expert to enter their knowledge or expertise into the expert system, and to refine it later as and when required.

Historically, the knowledge engineer played a major role in this process, but automated systems that allow the expert to interact directly with the system are becoming increasingly common.

The knowledge acquisition process is usually comprised of three principal stages:

1. Knowledge elicitation is the interaction between the expert and the knowledge engineer/program to elicit the expert knowledge in some systematic way.

2. The knowledge thus obtained is usually stored in some form of human friendly intermediate representation.

3. The intermediate representation of the knowledge is then compiled into an executable form (e.g. production rules) that the inference engine can process.

Knowledge Elicitation

The knowledge elicitation process itself usually consists of several stages:

1. Find as much as possible about the problem and domain from books, manuals, etc. In particular, become familiar with any specialist terminology.

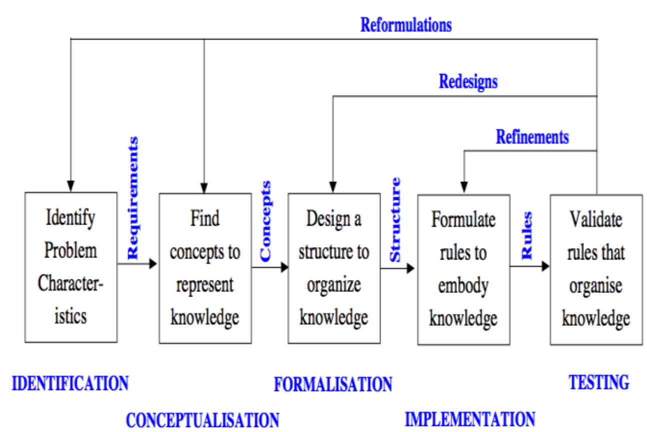
2. Try to characterise the types of reasoning and problem solving tasks that the system will be required to perform.

3. Find an expert (or set of experts) that is willing to collaborate on the project. Sometimes experts are frightened of being replaced by a computer system.

4. Interview the expert (usually many times during the course of building the system). Find out how they solve the problems your system will be expected to solve. Have them check and refine your intermediate knowledge representation.

5. This is a time intensive process, and automated knowledge elicitation and machine learning techniques are increasingly common modern alternatives.Stages of Knowledge Acquisition

The iterative nature of the knowledge acquisition process can be represented in the following diagram (from Jackson, Section 10.1):



KNOWLEDGE ELICITATION PROCESS

Levels of Knowledge Analysis

Knowledge identification: Use in depth interviews in which the knowledge engineer encourages the expert to talk about how they do what they do. The knowledge engineer should understand the domain well enough to know which objects and facts need talking about.

Knowledge conceptualization: Find the primitive concepts and conceptual relations of the problem domain.

Epistemological analysis: Uncover the structural properties of the conceptual knowledge, such as taxonomic relations (classifications).

Logical analysis: Decide how to perform reasoning in the problem domain. This kind of knowledge can be particularly hard to acquire.

Implementational analysis: Work out systematic procedures for implementing and testing the system.

CAPTURING TACIT/IMPLICIT KNOWLEDGE

One problem that knowledge engineers often encounter is that the human experts use tacit/implicit knowledge (e.g. procedural knowledge) that is difficult to capture.

There are several useful techniques for acquiring this knowledge:

1. **Protocol analysis:** Tape-record the expert thinking aloud while performing their role and later analyse this. Break down the their protocol/account into the smallest atomic units of thought, and let these become operators.

2. **Participant observation:** The knowledge engineer acquires tacit knowledge through practical domain experience with the expert.

3. **Machine induction:** This is useful when the experts are able to supply examples of the results of their decision making, even if they are unable to articulate the underlying knowledge or reasoning process.

Which is/are best to use will generally depend on the problem domain and the expert.

Representing the Knowledge

We have already looked at various types of knowledge representation. In general, the knowledge acquired from our expert will be formulated in two ways:

1. Intermediate representation – a structured knowledge representation that the knowledge engineer and expert can both work with efficiently.

2. Production system – a formulation that the expert system's inference engine can process efficiently.

It is important to distinguish between:

1. Domain knowledge – the expert's knowledge, which might be expressed in the form of rules, general/default, values, and so on.

2. Case knowledge – specific facts/knowledge about particular cases, including any derived knowledge about the particular cases.

The system will have the domain knowledge built in, and will have to integrate this with the different case knowledge that will become available each time the system is used.

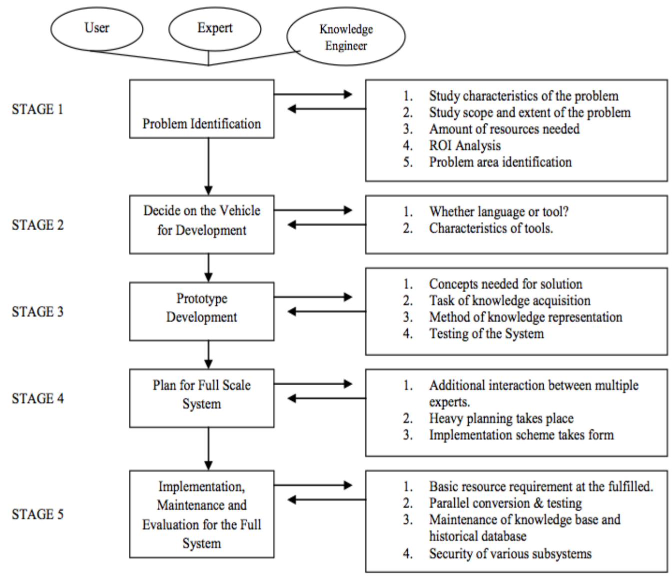


Fig.2: Expert System Life Cycle

There are five major stages in the development of an expert system. Each stage has its own unique features and a correlation with other stages.

Stage 1: Identification of the problem

In this stage, the expert and the knowledge engineer interact to identify the problem. The major points discussed before for the characteristics of the problem are studied. The scope and the extent are pondered. The amount of resources needed, e.g. men, computing resources, finance etc. are identified. The return-of- investment analysis is done. Areas in the problem, which can give much trouble, are identified and a conceptual solution for that problem and the overall specification is made.

Stage 2: Decision about the mode of development

Once the problem is identified, the immediate step would be to decide on the vehicle for development. The knowledge engineer can develop the system from scratch using a programming language like PROLOG or LISP or any conventional language or adopt a shell for development. In this stage, various shells and tools are identified and analyzed for the suitability. Those tools whose features fit the characteristics of the problem are analyzed in detail.

Stage 3: Development of a prototype

Before developing a prototype, the following are the prerequisite activities:

Decide on what concepts are needed to produce the solution. One important factor to be decided here is the level of knowledge (granularity). Starting with coarse granularity, the system development proceeds towards fine granularity.

After this, the task of knowledge acquisition begins. The knowledge engineer and the domain expert interact frequently and the domain-specific knowledge is extracted.

Once the knowledge is acquired, the knowledge engineer decides on the method of representation. In the identification phase, a conceptual picture of knowledge representation would have emerged. In this stage, that view is either enforced or modified.

When the knowledge representation scheme and the knowledge is available, a prototype constructed. This prototype undergoes the process of testing for various problems and revision of the prototype takes place.

Stage 4: Planning for a full-scale system

The success of the prototype provides the needs impetus for the full-scale system. In prototype construction, the area in the problem which can be implemented with relative ease is first chosen. In the full-scale implementation, sub-system development is assigned (l group leader and schedules are drawn. Use of Gantt chart, PERT or CPM techniques are welcome.

Stage 5: Final implementation, maintenance and evolution

This is the final life cycle stage of an expert system. The full scale system developed is implemented at the site. The basic resource requirements at the site are fulfilled and parallel conversion and testing techniques are adopted. The final system undergoes rigorous testing and later handed over to the user.

Maintenance of the system implies tuning of the knowledge base because knowledge, the environment and types of problems that arrive are never static. The historical database has to be maintained and the minor modifications made on inference engine have to be kept track off. Maintenance engulfs security also.

Evaluation is a difficult task for any AI programs. As mentioned previously, solutions for AI problems are only satisfactory. Since the yardstick for evaluation is not available, it is difficult to evaluate. However, utmost what one can do is to supply a set of problems to the system and a human export and .compare the results. When this method was adopted for the system MYCIN, it surpassed human experts.

The User Interface

The Expert System user interface usually comprises of two basic components:
1. 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.

2. 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.

CONCLUSION

Expert systems provide consistent answers for repetitive decisions, processes and tasks and holds significant level of information. The purpose of expert system is not to replace human experts, but to make their knowledge and experience more widely available and permit non-experts to work better. For the success of the expert system proper management of expert system, development and deployment is required. Success factor of an expert system depends on the problem to be solved, which must be qualitative and narrow in aspect. It is observed that domain experts not always able to explain their logic and reasoning. Also an expert system cannot respond creatively like a human expert in unusual circumstances and can automatically modify its knowledge base, or adjust existing rules or add new ones. The knowledge engineer is still responsible for revising and maintaining the system.

REFERENCES

[1] Annual Report of Birla Corporation Limited.

[2] Dennis Ritchi (1996), Artificial intelligence, Tata McGraw- Hill, New Delhi.

[3] Ephraim Turban and Jay E. Aronson (2001), Decision Support systems and intelligent systems, Prentice Hall, New Delhi.

[4] Jackson P. (1999), Introduction to Expert Systems, Harlow, England: Addison Wesley Longman, Third Edition.

[5] V.S.Janakiraman, K Sureshi, (2003), Artificial Intelligence and Expert Systems, Macmillan Series in Computer Science, New Delhi.

[6] Russel S, and P. Norvig, (2002), Artificial Intelligence: A Modern Approach, Second Edition, Prentice- Hall, New Delhi.