ASSIGNMENT - 8

AIM : Developing an book recommender (a book that the reader should read and is new) Expert system.

OBJECTIVE :

- To understand the basic concept of expert system, it's architecture and characteristics.
- To develop an book recommender expert system.

SOFTWARE REQUIREMENTS :

- Linux Operating System
- Java Compiler
- Eclipse IDE

MATHEMATICAL MODEL :

Consider a following set theory notations related to a program. The mathematical model M for an book recommender expert system is given as below,

M={S,So,A,G}

Where,
S= State space.
So= Initial State. In this case, the books that are already available are display according to the their rating and release rate.
A= Set of Actions/Operators.The simplest formulation defines the actions as selecting different types of books like novel, literature, science fiction, Java or HTML.
G=Goal state., In case of book recommender system goal state is to display top rated books alongwith their release date and rating.

THEORY :

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. 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.

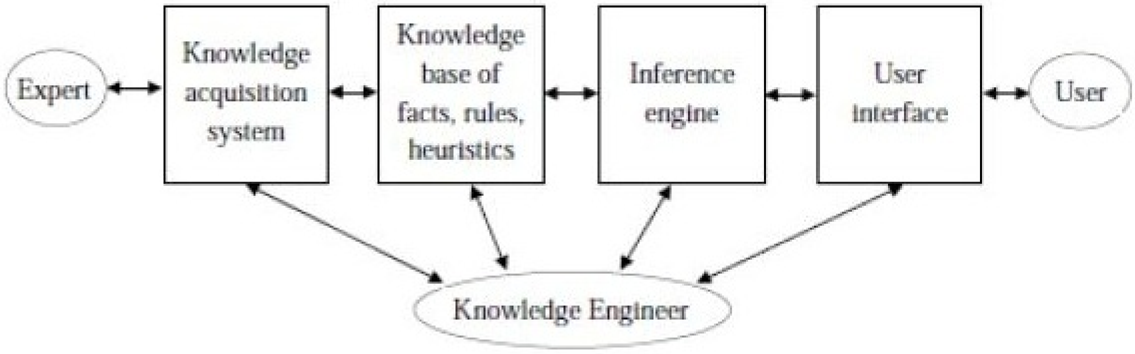
5. Predicting the future : Such as weather, share prices, exchange rates However, these days, conventional computer systems can also do some of these things.

Characteristics of Expert Systems :

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.



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. 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.

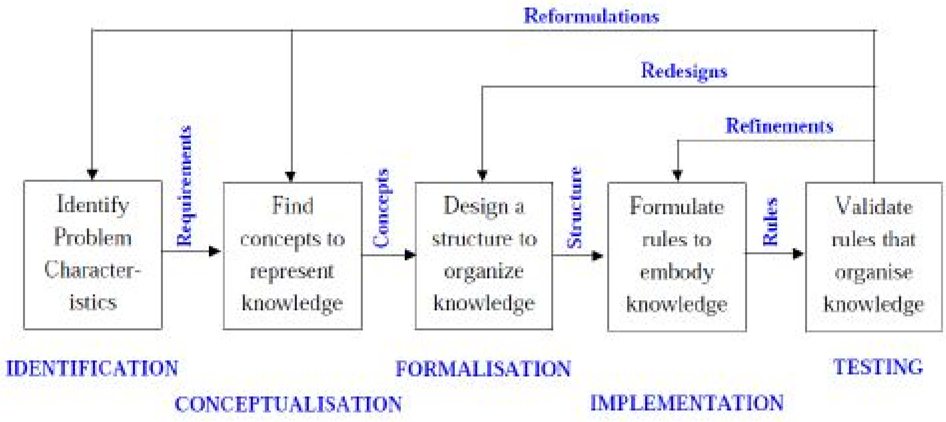
Knowledge Acquisition :

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.



Stages of Knowledge Acquisition :

The iterative nature of the knowledge acquisition process can be represented in the above diagram :

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.

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.

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.

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

Levels of Knowledge Analysis :

1. **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.

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

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

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

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

Capturing Implicit Knowledge :

One problem that knowledge engineers often encounter is that the human experts use 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.

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 systems inference engine can process efficiently.

It is important to distinguish between:

1. Domain knowledge the experts 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.

The Inference Engine :

The steps in the basic Recognize Act Cycle:

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 memory. Stop if a terminating condition is reached, or return to step 1.Early production systems spent over 90% of their time doing pattern matching, but there is now a solution to this efficiency problem:

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 systems solution, and also makes the systems 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.

An expert system can use 2 different methods of inferencing - Forward Chaining and Backward Chaining.

A Backward Chaining system (a 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.

A Forward Chaining expert system (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.

CONCLUSION :

Thus, we have developed an book recommender Expert system.

Roll No.	**Name of Student**	**Date of Performance**	**Date of Submission**	**Sign.**
		/ /2015	/ /2015	