CHAPTER 3

OVERVIEW OF EXPERT SYSTEM

3.1 Introduction to Expert System

Expert system is one of the likelihood of the applied Artificial Intelligence or AI. It is the set of rule based command or reasoning system. The Encata encyclopedia [31] give the meaning of expert system as follow.

"Expert System, a type of computer application that makes decisions or solves problems in a particular field by using knowledge and analytical rules defined by experts in the field. Human experts solve problems using factual knowledge and reasoning ability. In an expert system, these tools are contained in two separate but related components, a knowledge base and an inference engine. The knowledge base provides specific facts and rules about the subject, and the inference engine provides the reasoning ability that enables the expert system to form conclusions"

From the meaning above, expert system relate to 'problems in particular field', solving by human experts, 'knowledge base and reasoning ability', 'inference engine, and 'computer application'. Those key words will be discussed in later section.

In developing an expert system, there are three persons involving; human experts or expert domains, system developer, and end users. The human experts or expert domains are the person with high level of expertise in the problem area. Normally, the human experts have to concentrate to their work for many years. The human expert is the principle sources of knowledge acquisition. Their expertise will be transform into the computer program.

The system developer is the person who transform the expertise of human expert into the computer program. He has to design the system for easy to use, ability to maintenance, and further knowledge acquisition.

The last person involved in the expert system is the end user. The user will gain the benefits from expert system. He use expert system to solve their problem, consult for some advice, or learning by problem simulation. He will be asked by the system to supply the data or fact or assumption about the problem.

3.2 Architecture of Expert System

An expert system architecture comprises three main components: the knowledge base, the inference engine, and the user interface. The knowledge base is the most important part of the expert system. The expertise of human experts is transform into the knowledge base. It provides the data for giving the recommendations, indicating the potential sources of malfunction, and considering machine status, for instance. It is specific to each application. The basic architecture of an expert system is shown in figure 3.1.

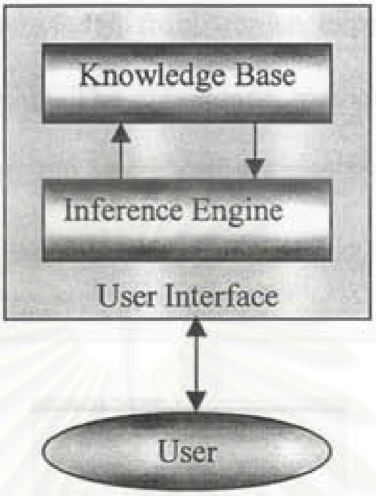

Figure 3.1 Basic architecture of an expert system

The inference engine is the mechanism used to controls the information and makes the assertions, hypotheses and conclusions. It is also controlled by the searching strategies to find the appropriate path and rules.

The inference engine is independence to the problem. So, it is not different between the two expert systems.

The user interface is the way that expert system communicate to the user via the screen, keyboard, or some graphic user interface (GUI)

3.3 Knowledge Acquisition

Knowledge acquisition is time consuming process since the system developer has to acquire as much as knowledge from human experts before transform those knowledge to a computer program. The experience, methodology of an expert have to be arranged to systematic reasoning base. Various way to acquire the knowledge are shown as below.

- Interviewing the human experts
- Experimentation
- Related literatures
- Training documents
- Internet

3.4 Knowledge representation

In AI, the crucial thing about knowledge representation languages is that they should support inference [Alison, 32]. We can't represent explicitly everything that the system might ever need to know - some things should be left implicit, to be deduced by the system as and when needed in problem solving; representing everything explicitly would be extremely wasteful of memory.

There is a tradeoff between inferential power (what we can infer) and inferential efficiency (how quickly we can infer it) [Alison, 32]. Both conflict advantage should be taken into account when designing the system.

In general, a good knowledge representation language should have at least the following features [Alison, 32]:

1. It should allow you to express the knowledge you wish to represent in the language.
2. It should allow new knowledge to be inferred from a basic set of facts.
3. It should be clear, and have a well-defined syntax and semantics.

In making a decision to represent knowledge, one should realize that representing some common-sense things in logic can be very hard. [Alison, 32]. Unfortunately, there is no common way to decide what knowledge should be presented. It depends on developer's judgement.

The idea of structured objects is to represent knowledge as a collection of objects and relations, the most important relations being the subclass and instance relations.

Production systems consist of a set of if-then rules, and a working memory. The working memory represents the facts that are currently believed to hold, while the if-then rules typically state that if certain conditions hold (e.g, certain facts are in the working memory), then some action should be taken (e.g., other facts should be added or deleted).

3.4.1 Structured Objects

A semantic net is really just a graph, where the nodes in the graph represent concepts, and the arcs represent binary relationships between concepts.

There was much discussion about what the nodes and relations really meant. People were using them in subtly different ways, which led to much confusion. It wasn't clear what the nodes and links really meant. It was difficult to use nets in a fully consistent and meaningful manner, and still use them to represent what you wanted to represent.

Frames are a variant of nets that are one of the most popular ways of representing non-procedural knowledge in an expert system. In a frame, all the information relevant to a particular concept is stored in a single complex entity, called a frame. Superficially, frames look pretty much like record data structures. However frames, at the very least, support inheritance. They are often used to capture knowledge about typical objects or events, such as a typical bird, or a typical restaurant meal.

One final useful feature of frame systems is the ability to attach procedures to slots. So, if we don't know the value of a slot, but know how it could be calculated, we can attach a procedure to be used ***if needed***, to compute the value of that slot.

3.5 Inference Strategies

3.5.1 Backward Chaining

Backward Chaining, often referred to as hypothetical reasoning, starts with a specific hypothesis, or set of hypotheses, called the agenda. The agenda structures knowledge and controls backward-chaining event processing by ordering hypotheses, or goals, in a numbered, hierarchical outline. The backward-chaining inference engine works backward from the agenda, pursuing a hypothesis via its search order strategies. Ddiagram in figure 3.1 shows the backward-chaining process.

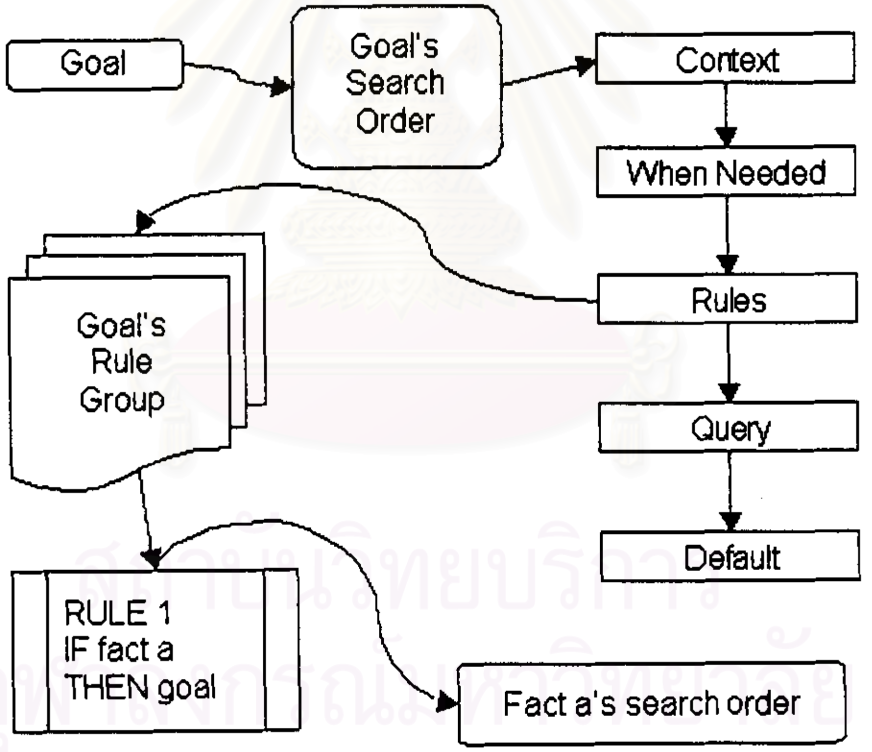

Figure 3.2 Process of Backward Chaining

(Source: rulesmachine [19])

The backward-chaining inference engine begins at the first goal or hypothesis in the knowledge base's agenda. The goals may have different name in different expert system shell. In the Level5 Object, for instance, the goal can be an attribute of the domain class or of a user-defined class or instance.

The inference engine then backchains through the goal's search order. The search order may not be the same in different expert shell. The search order tells the backward-chaining inference engine where, and in what order to obtain the value needed to verify the goal. In the Level5 Object, the goal can obtain that value from any combination of the following: the session context, a WHEN NEEDED method, the knowledge base's rules, and end-user query, or the default value as shown in figure 3.2.

From the Level5 Object online manual [19], if the goal can be concluded through rules, the inference engine backchains through the goal's rule group (the set of rules that can conclude the goal). If an attribute in the antecedent, of IF side, of a rule can conclude the goal, then the inference engine backchains through that attribute's search order in an attempt to find its value. This process continues until the goal is finally reached.

3.5.2 Forward Chaining

"Forward chaining, also referred to as data-driven reasoning, is an inference strategy that starts with known data or conditions and determines what can be concluded from that data" [19].

Unlike backward chaining, forward chaining is used in application where data is already available or when a system needs to react in real time to changing conditions. It starts at the known fact before go through related rules. The process of forward chaining can be seen in figure 3.3.

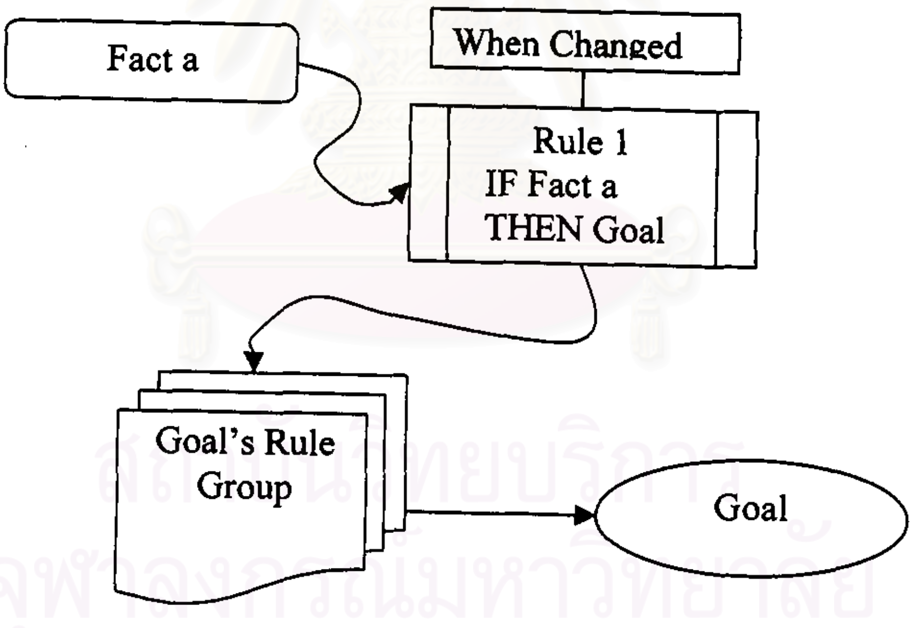

Figure 3.3 Process of Forward Chaining

Forward chaining can be used in a scheduling, animation, or process monitoring and control application.

According to the Level5 Object [19], forward-chaining inferencing occurs as follows:

- ✓ When the value of an attribute changes in the session context, the inference engine checks to see if there is a WHEN CHANGED method. If there is, the inference engine fires the method.
- ✓ The forward-chaining inference engine then examines the antecedents to see if they match the changed attribute value. This process of relating values in the context to patterns in a knowledge base's demons is called *pattern matching*. If a value is needed to evaluate whether a demon can fire, and it is currently undetermined, the inference engine does not backchain to determine its value. Instead, the demon fails.
- ✓ All demons whose antecedents match values in the context are marked to fire. The selected set of demons is called the *conflict set*.
- ✓ The inference engine then fires the demons in the conflict set based on the demon strategy the developer specifies. (See below for further information on demon strategy.)
- ✓ If any values change as a result of a demon or WHEN CHANGED firing, the inference engine immediately rescans the demon's antecedents for other demons that are eligible to fire

3.5.3 Forward Chaining and Backward Chaining

To select the inferencing strategies, it depends on the properties of rules in knowledge base and initial facts. The backward chaining is much more efficient when the goal or hypothesis is clearly identified. It can avoid drawing conclusions from irrelevant facts. However, sometimes backward chaining can be very wasteful; there may be many possible ways of trying to prove something. The worst case is trying almost all of them before finding one that works [32].

Backward chaining is suitable for proving a single fact that giving a large set of initial fact while forward chaining is better if a small set of initial facts is known. Lots of rules would be eligible to fire if forward chaining is used in proving large set of initial facts. However, both inferencing strategies can be used in the same application. It is mixed mode chaining that may suitable for large size and complex application.

3.5.4 Mixed Mode Chaining

The mixed-mode chaining is the combination of backward chaining and forward chaining. It is useful in large or complex application, which some parts require backward chaining while some parts require forward chaining.

For example, forward chaining can be used within a primarily backward-chaining application for data validation and range checking or to trigger or monitor event [32].

3.6 Developing Tools Selection

To develop the expert system, many of developing tools can be used. The high level language such as C, FORTRAN can be used, but the programming may take more time since the system developer has to write expert system structure from nothing. The developing tools which are very popular such as Microsoft's Visual BASICS also can be used. It is more convenient when compare to the high level language.

The most appropriate tools are the reasoning based shell, such as VPExpert, Level5 Object. They are built especially for reasoning based programming.

In this research, Level5 Object is used because its many points of advantages. For example, Level5 Object can link with the SQL format data base. It can also present the graphic user interface. Moreover, its inference engine is very capable for complex programming.

3.7 The Level5 Object

Rulemachines [19] has explained the Level5 Object as follows:

LEVEL5 Object is a hybrid application development tool that integrates object-oriented techniques and expert system technology with traditional, procedural programming. LEVEL5 Object's knowledge base may contain class declarations, backward-chaining demons, an agenda, methods, displays, and object-oriented databases.

LEVEL5 OBJECT captures and codifies valuable expertise in the form of objects, business rules, triggers, and smart forms.

Objects in a knowledge base are created via class declarations. Class declarations define the structure of the objects contained in a knowledge base, and, as instances, hold and retain the application's data values. Rules and demons describe the operational logic, rules-of-thumb, and cause-and-effect relationships needed to make decisions and trigger certain events or actions during a session. An agenda schedules the main events and the sequences of events or procedures that follow. Methods establish procedures for determining values, and displays interact with the end user by prompting for values, showing graphics and animation, or reporting data. Object-oriented databases are managed by LEVEL5 Object's Object-Oriented Database Management System (OODBMS) which allows a knowledge base to obtain the structure and attribute values of a class from an external database. The OODBMS maintains all objects in LEVEL5 Object identically, regardless of their source.

3.8 Basic Steps in Developing an Expert System

Hufemia [9, p.8-10] presented the basic steps in constructing expert systems as below.

Step 1: Identification of a problem. *An appropriate problem should be identified and selected to start with. In some cases, a feasibility study may be required to establish the real need. For it to be feasible, it should be justified, necessary and appropriate.*

***Step 2: Selection of experts**. Documented knowledge for a particular domain are easily accessed through handbooks, manuals, databases, and others. However, undocumented knowledge – which is equally, if not more, important for an expert system – can only be found and acquired through human experts. Care must be taken in selecting and identifying the experts. Their heuristic knowledge is much more complex than the documented ones.*

***Step 3: Selection of hardware and software.** Traditionally, computer tools were scarce and expensive to obtain; expert systems needed workstations and specialized computer hardware to run. Today, however, microcomputers are used to serve the same purpose. Several commercial expert system shells are now available so that developers do not have to write the program from 'scratch'. An expert system shell is a computer program that does not contain the knowledge base, which must be created by the knowledge engineer. It only contains the inference engine and the knowledge development tools. That is, one expert system shell can be used to create several expert systems by encoding different knowledge bases, without having to write the inference engine.*

Practically, tool selection is affected by whatever is already available. Although in principle, tool selection should consider the application and the knowledge that would be presented and the capability of the tool. For small systems, PCs should be sufficient hardware.

For expert systems written without the aid of shells, hardware selection is influenced mainly by the requirement of the programming language being used for the particular application.

Step 4: Acquisition, representation and inferencing of knowledge*. Acquisition of knowledge can be started after the first three steps have been satisfied. This knowledge is then represented in the computer by translating it to a language that the computer understands. The inference engine is then created as the acquisition process continues.*

Step 5. Development of an expert system prototype. *A prototype should be developed prior to completing the whole system. A prototype can be run just like the true expert system, but with a very limited capability. This is carried out to ensure that the development goes well as intended by the user before a lot of resources are wasted. This also helps the developer to decide on the structure of the knowledge base easier.*

Step 6 Performance evaluation, improvement and validation. *The prototype, as well as later improvements of the expert system, is subjected to tests and evaluations for performance. This is done using historical and proven cases provided by the users. Performance evaluation also determines the limitations of the system. Their limitations are usually the subject of further improvements. Improvements are made to update the existing rules with new ones to make the system more complete. Care should also be taken when updating to ensure that no rules contradict other rules.*

Step 7. User's acceptance and training. *Acceptance by the users depends on the capability of the system to handle problems. It will also depend on behavioral and psychological factors. For example, people may become intimidated by the system and think that their jobs are no longer secure.*

Training shall also be done to make sure that the users know how to input the information needed by the expert system before solving any problem. It will also familiarize them with the capabilities and limitations of the system.

Step 8. Installation approaches. *Once the expert system reaches an acceptable quality and reliability, it can be installed and used by the users. Depending on the situation, human experts may want to sit with the expert system for sometime before letting it work on its own.*

Step 9. Documentation and maintenance plans. *Expert systems can be continuously developed to contain everything about a domain and about the particular firm using it. Therefore it is important that such systems are properly maintained and kept to avoid unauthorized use.*

3.9 Literature survey

Related literature survey in expert system field is presented hereunder. Furthermore, apart from traditional research paper, many web sites that provide useful information in expert system field and its application are presented as well.

Asuwapongpatana [1] developed an expert system for the selection of air conditioning system design for Thailand environment, which is different from other developing countries environment. The selection of air conditioning depend on many factors, such as installation area, heat source in such an area, type of work, and forth. He claims that the developed expert system can satisfy the user who responsible for air conditioning system design. . The backward chaining inference strategies was employed in this thesis.

Atiraklapvarodom [2] developed an expert system for practice and trouble shooting of the vertical shoulder coated leveling process of a printed circuit board. In the process, printed circuit board has to be coated by machine. Many factors affect the process such as temperature, humidity, coating material concentration, and forth. They can lead to unacceptable quality and failure in the process. The developed expert system can help the operators in controlling the process.

Bentotage [3] mentions about using Export System approach for highway construction. He developed both data bases and rule bases for the highway construction scheduling and duration estimation. The limitation of this study is the expert knowledge was gain from only one construction project. In real situation, highway construction projects are also complex and different. The shared experience from other highway projects would be more advantage. The Expert System that used in this study was "VP Expert". The author did not mention about version but he claimed about its limitations. The VP expert lack to prepare the arrow diagrams. It also has a limitation of forty characters for any conclusion message display included complete and correct description of activities.

Chindaratana [5] developed an expert system for the scheduling of production in PCB industry. The expert system can provide the appropriate schedule for each operation. The complex problems of scheduling can be solved by breaking down them into controllable sub-problems. Then the planner can use the developed expert system in effective planning in each sub-task.

Hufemia [9] had developed an Export System interfaced to AutoCAD design. This developed system can help to assist in determining and locating the appropriate fixture types in design. Moreover, it can help in slight modifications to generic features and manually checking to ensure the output is correct. The limitation of the developed expert system is it acquires knowledge from a few available projects while at each construction design company use different tools and techniques. Therefore, the expert system need further adjustment and development before using in another companies.

Jiaranuchart [12] developed an expert system for the preliminary evaluation of telephone switching systems in economics and technical aspect in system capacity and service performance. The system was developed by M1 expert shell. Its knowledge base has two major parts; the preliminary evaluation in economics and technical aspects and detailed technical evaluation for technical aspects in the system capacity and system performance. The result of the research is the expert system can reduce evaluator's skill requirement and evaluation time.

Juneja [14] developed 'CNC EXPERT' for diagnosis failure in 'OKUMA CNC'. The problem come from irresponsible CNC machine seller who do not have enough knowledge in their machine and always leave the problems to its customer. The available trouble shooting is unclear and does not cover all the problem area, especially the controller of the CNC machine. Although the machine has its self-diagnosis system, it can not diagnose some problem. So, the developed expert system is needed in diagnosis the failure. Backward chaining is used in the expert system.

Nitikhunkasem [16] use an Expert System apply in maintenance management at the Siam Gypsum Industry. He had developed Expert System, Named EXPERT-MM, to support in three main functions : maintenance policy suggestion, diagnosis, and maintenance scheduling. He used dBASE-III plus, turbo Pascal and rule-based expert system shell-EXSYS which support backward chaining architecture as development tools. The user interface is still on text mode, due to the limitation of software and hardware at that time. The other limitation is limit of memory in computer cannot support huge size of the expanded knowledge in the future.

Pattanakooha [17] developed the decision support system for overhauling power circuit breakers in electrical substation, called DECCIB. By employing the developed data base system, the DECCIB can assist the user in making a decision to overhaul the power circuit breaker. Data base system was built by collecting substation equipment record and daily data. The decision will made on three criteria; a number of short circuit interruptions, the cumulated short circuit current, and the cumulative square of short circuit current. The DECCIB was developed on Microsoft Access 2.0.

Pochana [18] developed the decision support system for production planning in chicken processing plant. The program was developed on personal computer by using database management system. However, with the limitation of computer technology in that time, the program did not support graphic user interface.

The result of the research is satisfy. The system can reduce planner's skill requirement. The planning schedule can be easily adjust and reduced by the program.

Shuyu [20] choose "MAHO 600E CNC " 5-axis milling machine as the problem area to build the expert diagnosis system. She had focused on major function failures, including the spindle system malfunction, each feed drive system malfunction, the automatic tool changing system malfunction. She had collected knowledge from responsible engineer at AIT workshop, document of the machine, and her real practice experience from the machine.

Sondhi [22] had developed an Export System for assembly planning, called EXAP, that is intend to separate the acceptable components from the unacceptable ones and obtain sequencing of operations for use in a specific assemblies. The developed system is only for uniaxial rotation component.

Sue [23] developed an expert system for plastics processing methods selection. The knowledge base cover product shape, geometry of product, production rate, dimensional tolerance, mechanical strength, constraint relaxation providing less suitable solutions, rasin type, questions based upon the entered information, and so forth. The expert system was developed on expert system shell; the Smart Element version 1.0.

Ubolriabroy [25] developed an expert system for determining cost of ornamental rings. The C language was used in the development. The expert system was further developed from the demonstration of expert system shell prototype. It use the forward chaining strategy.

Vongderri [26] developed an expert system for diagnosis and maintenance on the operations of industrial fire tube boiler, called BODES. The BODES was developed by using M1 expert system shell. It has been dealing with external program in operating the boiler which are automatic feed water control, automatic fuel oil supply control, automatic combustion control, and operating safety system.

http://www.computer.org/intelligent/

This is an official web site of IEEE. It contains useful knowledge in artificial intelligence field. The online 'intelligence system and application' is provided for both its member and free sample for non-member. Moreover, many useful links in artificial intelligence in World Wild Web can be found.

http://www.usc.edu/schools/business/atisp/AI/AI-Bus/sigaibus.htm

This homepage is 'AAAI Special Interest Group on AI in Business'. The topics provided in this web site are the application of AI in business field, especially accounting field. Furthermore, the research and practice topics in artificial intelligence are provided, such as ‘AI and Reengineering’, ‘AI and Agents’, ‘AI and Internet Information Systems’, and forth. In addition, many useful links in AI resource in internet are provided.

http://www.bus.orst.edu/faculty/brownc/

This is the homepage of CAROL E. BROWN, an Associate Professor at Oregon State University. His education includes a BS in art, an MS in accounting and a Ph.D. in computer science. He actively researches, presents and publishes in the accounting and business expert systems area. Therefore, basic tutorial of expert system and its application in accounting field are given in this homepage. The content of AI tutorial is shown as follows.

I. What is Artificial Intelligence (AI)?

II. What are Expert Systems (ES)?

  Functional Components

  Structural Components

III. How do People Reason?

IV. How do Computers Reason?

  IV-1. Frames

  IV-2. Rule Based Reasoning

    IV-2a. Knowledge Engineering

  IV-3. Case-Based Reasoning

  IV-4. Neural Networks

V. Advantages and Disadvantages

VI. Additional Sources of Information

VI-1. Additional Sources on World Wide Web

http://www.bus.orst.edu/faculty/brownc/eswa/eswa.htm

The 'Expert System With Application' or ESWA homepage contain many articles in expert system field. It is also prepared and maintained by Carol E. Brown. Hereunder are some examples articles from ESWA.

"EXACT.- EXpert in Analog Circuit Topology."

Expert Systems with Applications 2(2/3), pp. 137-152.

Samardar, Mojgan; Nevin, Joesph, H; Mazlack, Lawrence J. (1991).

This article discusses the development of a knowledge-based system that uses some of the -Fundamentals of analog circuit design as a body of knowledge. This system, EXACT (Expert in Analog Circuit Topology), creates new circuit topologies using an established set of well-characterized circuit modules. The outcome depends upon the way the transfer function is subdivided, finding the functional block (blocks) that produce the transfer function, and modifying the circuit to satisfy other constraints. For a given linear transfer function, EXACT is capable of creating up to five circuits using different topologies to satisfy the design requirements by recognizing different functional blocks, which perform the same task. The system is expandable to be able to consider more requirements in its designs.

"The Application of Deep Models in Industrial Expert Systems."

Expert Systems with Applications 2(2/3), pp. 187-194.

Cardefqosa, J.; Alonso, F.; Castellanos, J.; Garcia, J. (1991).

This article differentiates between the four main knowledge levels of a human expert: namely, structural conceptual heuristic, and epistemological knowledge. The conceptual relationships between these various types of knowledge are constituent elements of the deep knowledge encountered in industrial systems. We present advice as to how to recognize such deep knowledge in order to produce a better quality knowledge base Finally, a prototype developed in a industrial area is described, in which the components that affirm the existence of deep knowledge are identified, detailing the concepts mentioned above

"MEDIN:An Expert System for Processing Medical Insurance Claims."

Expert Systems with Applications 2(2/3), pp. 211-218.

Cierniakoski, John J.; De, Pahul; May, Jerrold H. (1991).

The development of an expert system prototype is described The system is intended to assist nurses who process medical insurance claims. The problem is essentially one of classification, where a set of codes and data on the claim form are used to arrive at a decision. We emphasize the lessons learned from knowledge acquisition and implementation and discuss the adequacy of rule based systems for solving similar problems, including a discussion of how the prototype, although functional in itself can be extended to become a fully viable system. In this article we detail the process of building the prototype. In particular, the knowledge elicitation stage-its duration, getting the expert to verbalize her information processing, and generalization from the set of examples available. We also highlight the benefits that insurance companies can derive from similar development efforts. We present the pros and cons of using shell-based packages for similar problems, including their convenient and user-friendly environment.

"PRODE: A Shell for Industrial Diagnosis."

Expert Systems with Applications: An International Journal 8(1), pp. 67-76.

Gini, Giuseppina; Sassaroli, Piera (1995).

PRODE a shell for building expert systems based on prospective diagnosis, provides the domain expert with a knowledge acquisition interface and a suitable inference engine. Two kinds of knowledge contribute to modeling the diagnostic process. Failure knowledge expresses cause effect relations between faults as well as the relations between the result of tests and the "degree of belief" about the presence of a fault. Strategic knowledge contains criteria to guide the selection of the appropriate action. A criterion refers to a particular viewpoint(such as the cost of a test or the est. occurrence of a fault) and its application orders or prunes the set of the actions to take with respect to this viewpoint. The tool for knowledge acquisition provides the expert with a user-friendly interface in order to acquire both kinds of knowledge. Prospective diagnosis involves acquiring new data to increase the knowledge about the system while the strategic criteria guide in this choice. The inference engine proceeds, alternating data acquisition and focusing until the most probable cause of the malfunction is found and a repair is executed.

"Development of a Machine Troubleshooting Expert System via Fuzzy Multi attribute Decision Making Approach."

Expert Systems with Applications 8(1), pp. 187-201.

. ***Liu, Shih-Yaug; Chen, Jen-Gwo (1995).***

Most current machine fault diagnosis systems emphasize on the correctness of the hypothesized result; however in time constrained situations, the efficiency of the diagnostic process becomes more important and should not be overlooked. This paper presents an Integrated Machine Troubleshooting Expert System (IMTES) that enhances the efficiency of the diagnostic process, improves the completeness and consistency of the knowledge base, and assists users in developing and maintaining their diagnostic systems. . IMTES consists of five modules: a diagnostic tree model establishes the hierarchical structure regarding the function or connectivity of the diagnostic system, a fuzzy multi attribute decision making module determines the most efficient diagnostic process and creates a meta knowledge base to control the diagnosis process, a knowledge base module captures the human expertise and deep knowledge to diagnose the possible machine fault, an inference engine module controls the diagnosis process and deals with the uncertainty from the user input and knowledge base itself, and a learning module uses the failure driven learning method to train the knowledge base from the past actual cases. The system has been successfully implemented on MS-Windows environment, and it is written in MS Visual BASIC. To validate the system performance, IMTES is compared to EXACT, an expert system for automobile air compressor troubleshooting, using 50 sample cases. The result shows that IMTES can reduce the number of queries by 20.7%.