UNIVERSITI TEKNOLOGI MARA

PROSIDING



Seminar INSIGHTS 2002
Universiti Teknologi MARA

BRC (Biro Penyelidikan & Perundingan), Universiti Teknologi MARA, 40450 Shah Alam

Biro Penyelidikan dan Perundingan
Universiti Teknologi MARA
40450 Shah Alam, Selangor
Malaysia

© 2001. Biro Penyelidikan dan Perundingan. Hak cipta Terpelihara.

Sila majukan sebarang pertanyaan kepada:

Penolong Naib Canselor(Penyelidikan)
Biro Penyelidikan dan Perundingan
Universiti Teknologi MARA
40450 Shah Alam
Selangor, Malaysia

Tel : 603 5544 2094 / 2093 / 2092
Fax : 603 5544 2096
e-mel : azniz132@salam.uitm.edu.my



Jawatankuasa Penyunting

Penaung	:	Y. Berbahagia Datuk Prof. Dr. Ibrahim Abu Shah Naib Canselor Universiti Teknologi MARA
Penasihat	:	Yang Berbahagia Dato' Prof. Dr. Abdul Halim Mohd.Nawawi Timbalan Naib Canselor (Pembangunan dan Penyelidikan Universiti Teknologi MARA
Ketua Penyunting	:	Prof. Dr. Azni Zain Ahmed Penolong Naib Canselor(Penyelidikan) Universiti Teknologi MARA
Penyunting	:	Prof. Madya Dr. Zainon Mohd. Noor Koordinator Penyelidikan (Sains dan Teknologi) Prof. Madya Dr. Rosmimah Mohd.Noor Kooordinator Penyelidikan (Sains Kemasyarakatan dan Kemanusiaan)
Lembaga Penyunting:		Prof. Madya Ir. Dr. Hj. Abdul Rahman Omar Dr. Rashidah Abd. Rahman Cik Dapeah Ahmad

EXPERT SYSTEMS AND LEARNING

Zaidah Ibrahim
Azlinah Mohamed
Faculty of Information Technology and Science Quantitative

ABSTRACT

Standard expert systems do not have learning capabilities. They require the software designer to incorporate qualitative knowledge of the domain, either in the knowledge base or in the inference process. It would be extremely useful if a program can produce its own knowledge and expands the content of the knowledge base or adapt to new situations. Thus, this research is concerned with the development of a PC-based expert system prototype with learning capabilities.

INTRODUCTION

An expert system is a computer system that emulates the decision-making capability of a human expert (Giarratano and Riley, 1998). It is considered as one of the Artificial Intelligence (AI) application areas because the method of problem solving is predominantly based on heuristics instead of algorithms as the conventional program solves problems.

Typical components of an expert system

The typical components of an expert system is illustrated in figure 1.1 and brief explanations for each component is given below:

i. User interface
The mechanism by which the user and the expert system communicate.

ii. Explanation facility
Explains the reasoning of the system to a user.

iii. Working memory
A global database of facts used by the rules.

iv. Inference engine
Make inferences by deciding which rules are satisfied by facts or objects, prioritizes the satisfied rules and executes the rule with the highest priority.

v. Agenda
A prioritized list of rules created by the inference engine, whose patterns are satisfied by facts or objects in working memory.

vi. Knowledge acquisition facility
An automatic way for the user to enter knowledge in the system instead of having the knowledge engineer explicitly code the knowledge.

Knowledge base, inference engine and database components are part of an intelligent agent component.

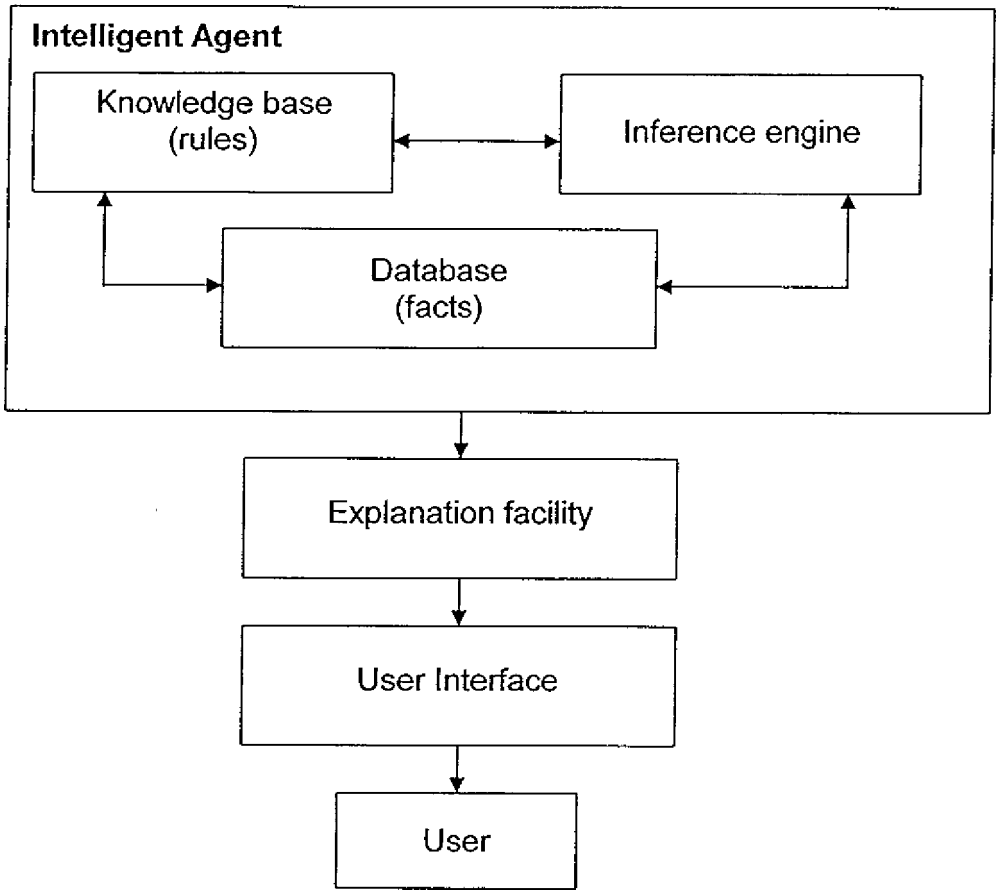


Figure 1 Typical components of an expert system.

In order to integrate learning capability, we need to understand how human reasons. A human may reason based on one of the following factors:

1. use specific a priori rule, for instance "If A occurs then B may happen";
2. use heuristic or "rule of thumb", for instance "If you hear thunder then bring an umbrella";
3. use past experience, "If you do not hear thunder then still bring an umbrella";
4. use expectations or observe patterns of behavior, for instance "If you are not as active as usual then you are not feeling well".

Computer's reasoning model is based on our model of reasoning. Reasoning based on heuristic can be simulated using a knowledge base where the knowledge base may consist of rules represented as if-then statements. Past experience can be stored as cases in a case base while expectations of patterns of behavior can be simulated using artificial neural network where patterns of input are observed to determine the output (Brown and O'Leary, 1995). This research chooses the second and third approach of human reasoning that is reasoning using heuristic and store past experiences as cases in a case base. Future research will be done for the other approaches. Figure 1.2 illustrates the integration of learning facility based on both of the approaches mentioned earlier.

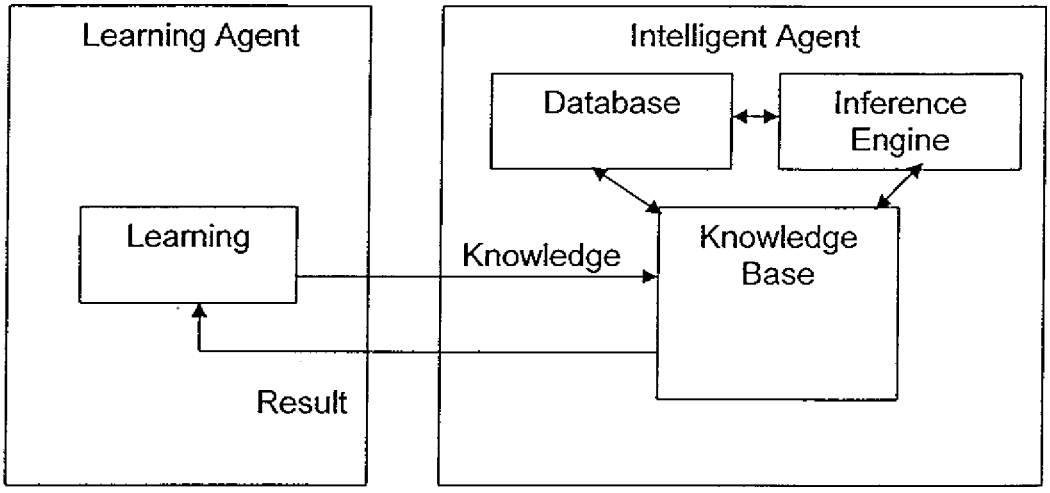


Figure 2 Learning component in an expert system.

Intelligent knowledge base diagnosis prototype

Reasoning strategies used in rule-based expert systems include backward chaining, forward chaining, various strategies, problem reduction, pattern-matching procedures, unification, event-driven methods and hybrids. The two approaches commonly used to make inferences are forward chaining and backward chaining. In the forward chaining, the rules are examined one after another in a certain order. The forward chaining reasoning also known as data-driven, bottom up, and antecedent-driven inference. It is best used to solve problems in which data is to be used as the starting point problem solving. In the backward chaining, the inference engine starts from the end.

This research uses forward chaining method where it goes forward from antecedents to the conclusion that it generates and is applied to diagnosis an aircraft. Each rule in the knowledge base of this research has one rectification, picture and detailed descriptions of the steps involved in the rectification.

Inductive learning systems do not require the software designer to incorporate all of the qualitative knowledge of the domain in the knowledge base of in the inference process. Given some examples (or pre-classified samples) of the domain, these systems have the ability to determine by themselves, an approximation to the underlying model of the task domain.

Relationship between the symptoms and the diagnoses can be determined where the system is said to have induced the appropriate classification rules. In other words the system has learned the diagnosis problem and able to provide the diagnosis even though the examples were not seen during training. This can be done through generalization process (Balakrishnan and Honavar, 1998).

Lets refer to table 1 for an illustration of the integration of inductive learning in an expert system that diagnoses an aircraft. Initially, the knowledge base consists of four separate rules with four separate diagnoses. But, if the rule is a combination of symptom 1 and symptom 2, the expert system could not produce the diagnosis since this rule is not included in the knowledge base. But, with inductive learning, the expert system can be trained with new rules. We can teach the expert system that if rule 1 and rule 2 occur, then, the result should be diagnosis 1 and diagnosis 2.

Table 1 Sample rules in the knowledge base.

Rule	**Symptom**	**Diagnoses**
1	Problem in normal operation of wheel brake system	Refer to text in Bk2 pg 2 Refer to picture in Bk 3 pg 2
2	Brake jam	Refer to text in Bk2 pg 3 Refer to picture in Bk 3 pg 3
3	Brake fails to hold	Refer to text in Bk2 pg 4 Refer to picture in Bk 3 pg 4
4	Brake feels spongy	Refer to text in Bk2 pg 5 Refer to picture in Bk 3 pg 5

With this training, the system should be able to handle other combinations of rules, for instance, rule 1 and rule 3 should produce diagnosis 1 and 3.

The process of case-based problem solving can be seen as show in figure 3. This process consists of six different parts that must be focused on, that are, from the case key-in by user until the process on how the case is being stored into the appropriate case storage.

A case is a problem situation that occurs in the aircraft. Usually, the expert engineer is the one that determines cases but in this research the developer is the one that determines the case. In this system a case entered by users are examined by the system. Each word(s) entered by the users are divided into an individual word and stored in an indexed list.

For example, suppose that user enters : "Jam when parking". Each word in the sentence actually is considered as a case (search by the user). Then the system counts the total number of words in the sentence. Finally these three words are stored in a temporary file list for the next process. Table 2 shows how the words or cases entered by the user are stored in the temporary file.

By using the above example, the system will retrieve the first related row in the column of *case name* (by referring to their related *problem code*) in the database then make a comparison with all the three words or cases. The word "Jam" will be examined by the system, which will match data with the selected case name in the database. If the word is exactly the same with the word in the database, then the system will enter a 1 to that row and vice versa. Then, the system will repeat the same step for the second word until all the word(s) in a sentence is examined.

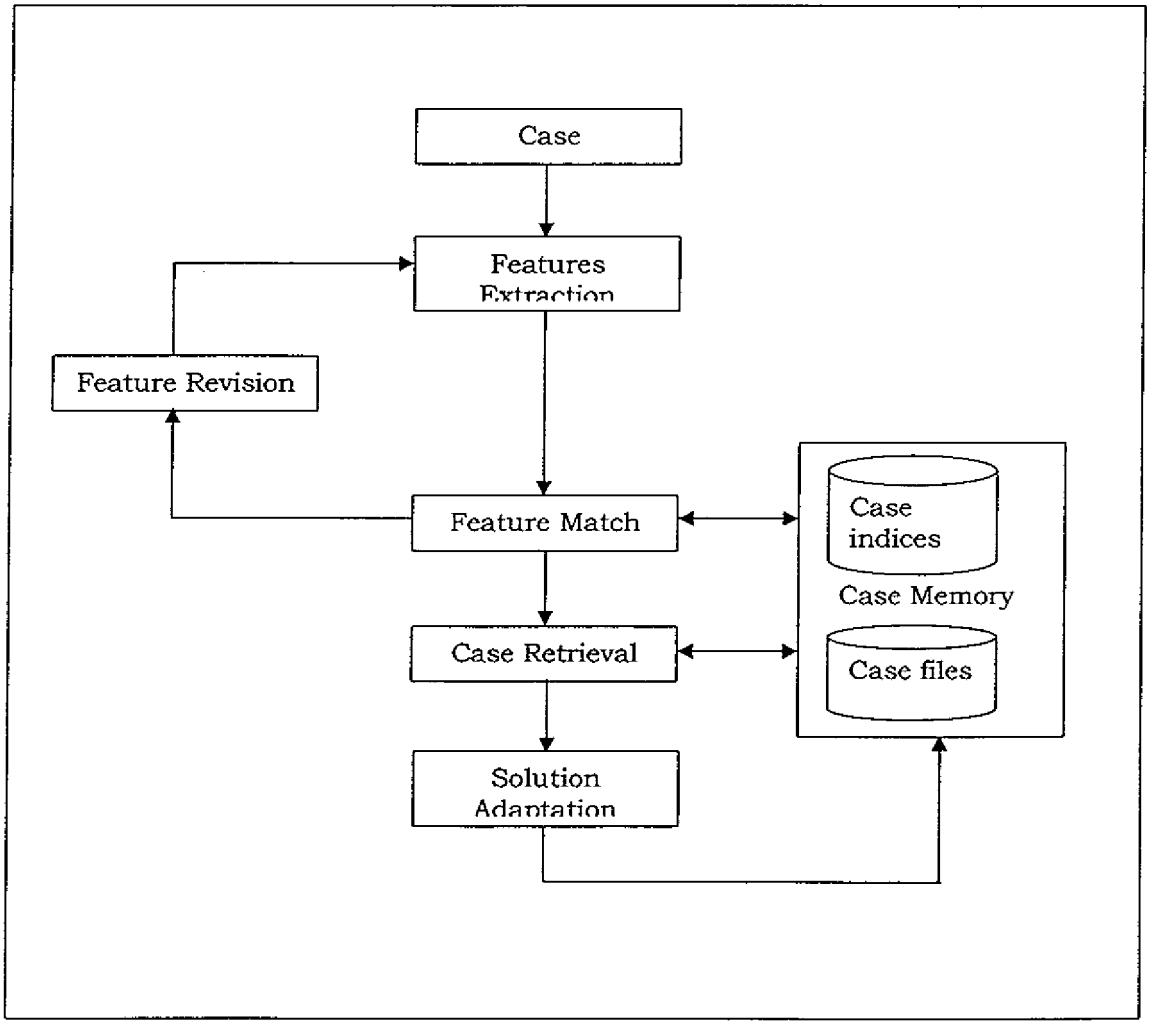


Figure 3 Process of case-based problem solving (Bing Chiang Jeng and Ting Peng Liang, 1995).

Table 2 Example of data store in temporary file.

Index	Word / Case
0	Jam
1	When
2	Parking

If the searching process is finished with the first selected case name, then the system will select the next case name in the database by referring to the problem code to be examined. This process will continue until the whole case name in the related problem codes are examined.

Problem code is identified by which problem area code covered in the problem situation. Examples of problem area are wheel brake, vibration and rotor brake. Each of these problems is given its own problem code. Case name is the problem situation regarding the problem area. Referring to the above example, if the word "Jam" matches with the case name selected in the database, the system will give a temporary value for that case name.

The temporary value(s) are used in this system to calculate the percentage of the case matched with the selected case name. The result will be produced by the system, if only one or more exactly search word(s) by user matches with the selected case name in the database. The system will ignore the case name with the non-matched word with selected case name.

The retrieved case names will be ranked by using percentage. Below is an example on how the percentages are calculated.

Word enter by the user : Jam when parking
Case name in the database : (i) Jam
: (ii) Ineffective when operated
: (iii) Spongy

The percentage of (i) will be 100% because the whole case name "Jam" is founded, while the percentage for (ii) will be 33.33%. This is because, there is only one word "when" entering by user matched with the case name. The mathematical formula is as follows,

$$\textbf{Result Percentage} = \frac{\textbf{Number of matched word}}{\textbf{Number of word enter by user}} \times \mathbf{100\%}$$

There is no result appear for (iii) because it will return 0% for that case matched.

If there is no matched word entered by user with case name in the database, the system will consider or accept it as a new case but if only the user definitely sure that is the problem of the situation.

In this research inductive learning is done only when two cases from different case name matched with the searched word or case by only the word "and" in between it. Below is the example of how the inductive learning done in this project.

Word enter by the user : Jam and spongy
Case name in the database : (i) Jam
: (ii) Ineffective when operated
: (iii) Spongy

The word "jam" and "spongy" are actually the case names in the database. When the system found the word "and" between them, the system will accept it as a new case and will ask the user whether it is the actual problem or not. Inductive learning process is done only when the users answer is "Yes". Then the system will combine all information about case name "jam" and all information about case name "spongy" to become one new case known as "Jam and Spongy". Then the system will create the new case code for that case name as reference. The result for that new case name also is provided by the system at that time. If users answer is "No", then the system will not accept it as a new case and inductive learning process will not be done in this situation.

SUMMARY

Expert systems would be a suitable approach for diagnosis problems provided that the human experts are able to articulate their knowledge and reasoning methodology to an experienced knowledge engineer. The main weakness of conventional expert systems is their inability to adapt to new situations or rules. This can be improved by integrating the expert system with inductive learning. But, inductive learning system requires a set of representative examples, called the training set which can indicate the relationship between the symptoms and diagnoses. Thus, this requires past diagnoses.

Since interesting domains tend to be large, a learner may only examine a fraction of all possible examples and from this limited experience, the learner must acquire knowledge that will generalize correctly to unseen instances of the domain. Thus, an expert system that can learn should be able to produce results for unknown data or data that is not coded in the knowledge base. What this means is that an expert system that learns should be able to expand its knowledge base. Further research will be made to improve the capability of the integration of expert systems and inductive learning.

ACKNOWLEDGEMENT

The authors would like to thank UiTM for sponsoring this research.

REFERENCES

A.I Mechitov and H.M Moshkovich, (1995). Journal on Knowledge Acquisition Tool for Case-Based Reasoning System., Elsevier Science Ltd.

Bing Chiang Jeng and Ting Peng Liang, (1994). Journal on Fuzzy Indexing and Retrieval in Case-Based Systems. Elsevier Science Ltd.

Chandler J. s. et al., 1990, Integrating Statistical and Inductive Learning Methods for Knowledge Acquisition, Systems With applications in the USA.

Gearge F. Luger and William A. Stubblefield. (2000). "Artificial Intelligence Structured and Strategies for Complex Solving", Third Edition, Addison-Wesley.

Giarratano and Riley, (1999). "Expert System – Principles and Programming", Third Edition, PWS Publishing Company.

Greg Kearsley, (1987). "Artificial Intelligent and Instruction Application and Method", Addison Wesley.

Honavar, J., et al., 1989, Diagnostic Reasoning in Continuous Systems, Division of Engineering Cybernetics, Norwegian Institute of Technology, Trondheim, Norway.

Jae Kyu Lee and Min Yong Kim, (1993). Journal on Case-Based Learning for Knowledge-Based Optimization Modeling System. Pergamon Press Ltd.

Joseph. G., 1998, Expert Systems: Principles and Programming, Third Edition, An International Thomson Publishing Company (ITP).

Karen Ketler, (1993). Journal on Case-Based Reasoning. Pergamon Press Ltd.

Larry D. and Brown C. M., 1997, Textbook of Medical Physiology, Seventh edition, W. B. Saunders Company, Philadelphia.

M. N. Kamel, 1995, A Prototype Rule-Based front End Expert System for Integrity Enforcement in Relational Data bases: An Application to the Naval Aircraft Flight Records Data Base, Information System Group, Naval Postgraduate School, Monterey, CA.

Wylie et al., 1997, Integrated Diagnostic System (IDS) for Aircraft Fleet Maintenance, M50 Montreal Road, Ottawa, Ontario, Canada.

Youngohc Yoon and Angela D. Acree, (1993). Journal on Development of a Case-Based Expert System. Pergamon Press Ltd.