An AI Tool and Its Applications to Diagnosis Problems

Hiroshi NARAZAKI, Toshiharu IWATANI, Kayako OMURA, Yoshihisa OTSUKA and Masami KONISHI

Electronics Research Laboratory, Kobe Steel, Ltd., Takatsuka-dai, Nishi-ku, Kobe, Hyogo-ken, 673-02 Japan.

(Received on May 9, 1989; accepted in the final form on September 8, 1989)

An expert system shell is developed for a personal computer using C language. It is designed for diagnosis problems such as fault diagnosis and process operation guidance. The tool is suitable for the knowledge that is represented in the form of the hierarchical tree and causal relationship matrix. Further by using the tool it is easy to build expert systems which cooperatively work with the existing software by passing or receiving data, and which evokes or is evoked by external programs. Therefore the tool is appropriate to construct on-line expert systems. Application examples are shown to prove the usefulness of the tool.

KEY WORDS: artificial intelligence; expert system; diagnosis; AI tool; prototyping; on-line application.

1. Introduction

The expert system (ES) technology is a major fruit of the recent development of AI technology. Numerous discussions have been made on the ES technology and its applications to industrial problems, such as process monitoring, fault diagnosis, and process control.¹⁾ The ES technology is expected to open a way to integrate the algorithmic processing and the heuristics into an intelligent information processing system by offering methods to support the acquisition, representation and processing of various types of knowledge.

Formerly, it was widely believed that the AI technique, including the ES technique was difficult to use in practical applications because the AI program often imposed a heavy burden for the computer, and the introduction of the AI application required a large cost. However, the recent development of the computer hardware and software is changing that belief into a mere prejudice. In the near future AI applications which would have required an expensive LISP machine a few years ago will be available on a 32-bit personal computer.

As the ES techniques spread widely, the establishment of the ES building methodology is strongly needed. In most cases the difficulties for the successful ES development exist in the formalization of the problem in terms of ES paradigms and the acquisition of the knowledge lurking in the expert's mind. Various principles have been presented so far to deal with those difficulties.²⁻⁶⁾

There are many AI tools to help building the expert systems. However most of them have been designed for use in laboratories. Even the most sophisticated tools have problems with the practical abilities such as the inference speed, or the interface with the existing software and hardware. Therefore we developed an AI tool, named IDMON, for the diagnosis problem so as to be used in the field in com-

bination with the existing system. The basic concept of IDMON is based on Ref. 7) and we add some powerful facilities for practical usefulness. By using the tool, it is possible to construct an expert system which cooperatively works with other external softwares, or to develop on-line expert systems whose inference processes are evoked by external events. Our tool is developed by using C language for the efficiency of the inference speed. In this paper the knowledge representation and processing method of IDMON are stated. The application examples to a blast furnace and inspection machine are also introduced.

2. Knowledge Representation in IDMON

2.1. Knowledge Elements in IDMON

IDMON needs three types of objects to represent the knowledge: (1) hypotheses, (2) facts, and (3) rules.

The hypotheses and facts represent statements and they have the certainty factors (CFs) as their values. The CF value is an integer and it is so scaled that "0" means "false", "100" means "true", and "50" means "unknown". The CF of the hypothesis is internally decided by the inference procedure while that of the fact is externally decided by the user inputs, external programs, data files, etc. Fig. 1 shows an example of AND/OR tree. The tree has one root which is analogically the same as the "Top Event" in the Fault Tree (FT)⁸⁾, the intermediate nodes which correspond to intermediate hypotheses, and the bottom nodes which denote conclusions. The logical combination of the facts specifies which path to follow at each intermediate node. The inference engine tries to find a path from the root to a bottom node, along which all of the logical conditions of the facts are true.

The above three types of objects are defined in the

Fig. 1.
Example of AND/OR tree.

"A $\wedge$ B"
C
means that if both A and B are true then C is true

knowledge base, which is given in the following format.

(2.1) Definition of hypothesis and facts:

{%< optional parameters >%}< name >
< explanation >

Definition of rules:

{%< optional parameters >%}< name >
< parent hypothesis >;
< logical condition of facts >;
< child hypothesis 1 >, { - < child hypothesis 2 > },

where { } means that it is optional and can be omitted, and
% represents a delimiter of the optional parameters.

In Fig. 2 the examples of knowledge definitions are shown. ' % ' in the rule definitions represents a root node. In the definition formulae, the optional parameters offer interfaces to/from the external software or data. Details will be mentioned later. The name is a label of an object and the explanation gives its description. When the system wants to ask the user if a certain fact is true or false, the system displays the explanation of the fact to prompt the user to answer. The rule description gives the tree structure. If the logical condition in a rule is satisfied, the CF value propagates from the parent hypothesis to < child hypothesis 1 >, or in case that the condition is false, to < child hypothesis 2 >. In other words, the logical condition is a gate or a switch which controls the flow of CF value from node to node.

2.2. Hierarchy and Relation

There are two kinds of knowledge representation for the diagnosis problems. One is a tree representation and the other is a cause-symptom matrix representation.

The tree representation is useful to describe the process of classification and distinction, or the " check, and see " procedure. It is so easy to arrange the order of checks and tests that the representation is suitable to tell the user what to do first, and how to respond to the result. However the logical structure

(i) Hypothesis definitions

α (This is an intermediate hypothesis.)
 β (This is a conclusive hypothesis.)
 γ (This is another conclusive hypothesis.)

(ii) Fact definitions

A (Fact A is true.)
B (Fact B is true.)
C (Fact C is true.)

(iii) Rule definitions

r1 (%A; α)
r2 (α ;B; β)
r3 (%C; γ)

Fig. 2. Example of knowledge definitions.

should be enough definite and hierarchical to make the tree representation successful, which is often difficult in practical problems. Furthermore in most cases the expert's knowledge is not so definite or rigid to be arranged in a tree format. For example it sometimes occurs that any of certain tests or questions can be done in an arbitrary order and that an expert makes inferences based on a set of findings, not by each finding. This means that an expert does not question or test with the purpose of distinction for each case. His intuition tells him what is to be performed on the basis of collection of findings. In that case each question has not a clear logical purpose to distinct a certain situation from others. If a tree is constructed in that case, then it is a standardization of the expert's task, which is another role of the expert system.

The matrix representation is easy to make in the nonstructural knowledge stated above, and it superficially seems that the representation is suitable when the knowledge base maintenance is carried out by the field experts. But it cannot perform an efficient and intelligent diagnosis if there are many conclusions

and facts, because the representation never tells the structure or hierarchy among the facts and hypotheses. Sometimes the user may feel that the system asks him what is seemingly irrelevant to the situation, or the conclusions are not adequately refined. Therefore some procedures to extract a logical structure from the table is needed, for instance when the analysis of diagnosability finds groups of conclusions which can be definitely judged true or false by the findings.⁹⁾

In practical applications the mixed structure of both representations is required. In the mixed structure the tree representation defines categories of hypotheses and conclusions, and if the knowledge is not well refined and structured in a certain category, then the matrix representation is used. An example is shown in Fig. 3. In the tree representation, the inference engine searches for a path from the root node to the bottom node, while in the matrix representation, it collects the facts and amasses the certainty factor for possible conclusions, based on the cause-symptom relations in the matrix, and selects the most plausible conclusion. In case of the mixed structure the engine searches an adequate category, calculates the CF values of conclusions in that category, and selects the most plausible one as a conclusion.

As stated before, if the main purpose of the expert system is to tell the users how to carry out the diagnosis procedure, namely the standardization, the tree representation is appropriate. There are other cases where the relation of causes and consequences are not logically clear and given by fuzzy expressions such as "the fact has much, something, little, or nothing to do with a certain hypothesis", then the matrix representation is inevitable. In IDMON, both types of rule representations are available. The examples of

the two representations are shown in Figs. 4 and 5.

Fig. 5 shows the rule description in IDMON of the tree/matrix representation depicted in Fig. 4. The fact conditions are expressed with symbols such as " , ", " + ", and " - " which correspond to logical operators such as " and ", " or ", and " not ", respectively. For example "fact A is true and (fact B is not true or fact C is true)" is expressed by "A, -B+C". When a rule gives a row of a matrix, the CF values of facts separated by " , " are summed to a total CF value where the combination weight can be specified such as $70*f1$. This means that the CF value should be calculated as $50+0.7 \times (\text{CF value of } f1 - 50)$ where the deviation of the fact's CF value from the unknown level, 50, is multiplied by 0.7. This weight expresses the strength of causal relationship between the fact and the conclusive state.

The interpretation of a rule, namely whether it is a portion of a tree or a row of a matrix, can be decided by the optional parameter to the rule. More about the optional parameter will be discussed in more detail later.

3. Inference

3.1. Inference Algorithms

In IDMON, the inference is carried in a forward chaining manner. The system keeps the "current hypothesis" which corresponds to the node on which the inference process is putting an eye. The inference engine selects a set of rules whose parent hypothesis is the same as the current hypothesis, and the fact conditions are tested by Q & A with users, the reference of external files, or the execution of external programs. Two types of inference algorithms are

Fig. 3.
Mixed structure of tree and matrix.

Fig. 4.
Example of mixed structure.

provided.

One type is that the engine tests the conditions of relevant rules and calculates the CF values of the conclusive hypotheses, and the current hypothesis is replaced by the hypothesis of the largest CF value among them. The other is that, if the engine finds a rule in which the CF value of a conclusive hypothesis is above a certain level, then the current hypothesis is renewed to the new hypothesis. The former type of inference is similar to the width-first search where the CF is regarded as a performance index. The latter is similar to the depth-first search. It is totally optional which type of algorithm is to be chosen. The depth-first search is preferred when the knowledge is represented in a tree while the width-first search is suitable for the cause-symptom matrix representation. Our experiences tell that if the inference is carried out through Q & A between the expert system and the user, the depth-first search is more appropriate. The width-first search would give the user an impression that the questions are redundant and the systems is not approaching to the conclusion.

3.2. Inference for On-line Expert System

Some applications require on-line expert systems. Some features are added to IDMON for the purpose of the on-line use:

- (1) IDMON can be evoked by a manager program or an external program.
 - (2) IDMON starts inference after loading the CF values of facts from the initial data file.
 - (3) The optional parameters which terminate the inference and return the control to the manager program are available for conclusive nodes.
- By using the above facilities, IDMON can be evoked periodically or by some external events. In Chap. 6 we will show examples of the on-line inference.

4. Optional Utilities

It is seldom the case that the expert system is constructed only by the inference procedure. We usually

```

r1 (%;A;α)
r2 (%;-A;β_or_γ)
% x % r3 (β_or_γ;70*C;β)
% x % r4 (β_or_γ;30*B,90*C;γ)
    
```

Fig. 5. Rules expressing the knowledge in Fig. 4.

need cooperation between the conventional data-processing software and the inference software. The expert system calls or is called by other softwares and it should share some data. Even if the expert system is targeted to work with the user through Q & A processes only, it is sometimes required for the expert system to display the graphical information or the manuals which help the user to understand and give correct answers to the questions made by the system. Such utility never performs the inference but it is practically necessary. In IDMON the user can put various optional functions for the hypothesis, the fact and the rule to realize those facilities. It is possible to attach commands which work as a help facility, if-asked demon, or if-found demon. For example the user can specify an alarm program which is evoked when the CF value of a certain hypothesis node is found to be above a specified value.

These facilities can be used by specifying optional parameters to hypotheses, facts, or rules. Some of those optional facilities are listed below.

(1) Major Optional Parameters for Facts

- Specification of commands to be executed when the CF value of the fact is asked (If-asked demon facility).
- Linkage to a text file to be browsed when the CF value of the fact is asked to the user (Help facility).
- Linkage to a file from which the CF values of facts are loaded or to which the data are stored.

(2) Major Optional Parameters for Hypotheses

- Specification of a CF value level which the hypothesis is thought to be true.
- Specification of commands to be executed when the hypothesis is inferred to be true, that is, when the CF value is more than the specified value (if-found demon facility).
- Linkage to a text file to be browsed when the hypothesis is found true.
- Termination of IDMON when the hypothesis is established.
- Backtracking when the inferred CF value is less than

the specified value.

- Designation of another knowledge base which the inference process is to use when the specific hypothesis is concluded (hierarchical inference).

(3) Major Optional Parameters for Rules

- CF value of the rule.
- Linkage to a text file to be browsed when the rule is under proof procedure.

5. Knowledge Acquisition Process

The "prototyping approach" is known to be adequate for the development of the expert systems. Here we state an example which shows how the prototyping approach is carried out by using IDMON. The steps to be followed are listed below.

STEP 1: Identification and Classification of the Problems

First the knowledge engineer (KE) should identify what type of problems or situations are to be diagnosed. For example, a KE asks an expert what kind of problems are transmitted to him.

STEP 2: Choice of a Specific Problem as a Prototyping Topic

The KE chooses one problem from those listed in the previous step. A prototype expert system is constructed for that topic. The root node is assigned to that situation.

STEP 3: Trace of Actions to Be Taken

The KE asks the expert for usual action to solve the situation. In this step the facts and hypotheses are listed up.

STEP 4: Explanation of the Experiences

Based on the concrete experiences, the KE asks the expert what he did at a particular occasion. This is different from asking what he usually does. This step gives the structure or causal relations among the hypotheses or the facts. If the action and inference are carried out sequentially to prune the candidates of the conclusions, then the tree is formatted. However if the inference is somewhat intuitively done based on the collection of data, then the matrix representation is more appropriate.

If the goal of expert system is the standardization of the expert's task, then the tree structure should be adopted as a standard.

STEP 5: Test and Modification of the Knowledge Base

The constructed tree or matrix is to be updated through field tests or case studies. It is better to carry out the inference through off-line Q & A procedure. The task constitutes the following activities;

- Addition or removal of new facts or new hypotheses
- Refinement of the explanation sentence of facts
- Modification of a tree structure
- Adjustment of the causal relationship strength of a cause-symptom matrix.

In case of the matrix representation the KE is recommended to extract a tree structure from the matrix,

which is important to improve the knowledge base.

STEP 6: Expansion of the Knowledge Base

The KE expands the knowledge base for other problems listed in STEP 1 and follows the same STEPS from 2 to 5.

If the knowledge base grows large for each problem, then the knowledge base is divided into units by using the hierarchical inference option for conclusive hypotheses.

STEP 7: Linkage to External Software or Data

If there exist some programs or data files that should be evoked to find the CF value of the facts, should be executed when a certain hypothesis is inferred to be true, should be referenced during the inference procedure, or should be browsed during the Q & A process by the user, then the expert system is linked to those programs or data files by using the optional parameters for the hypotheses, facts, and rules.

STEP 8: Introduction of Expert System into Practice

Finally the expert system is introduced into practice after developing some interface programs that are required for the practical use. If it is an interactive system, then its usage is to be explained to users. If it is an on-line expert system, then it is connected with the existing system.

6. Applications

6.1. Action Guidance of a Blast Furnace¹⁰⁾

This system performs an operational guidance of the heat level control of a blast furnace.

Fig. 6 shows a block diagram of the action guidance expert system of the blast furnace. IDMON is used in the block surrounded by a dotted line. The logged data of the blast furnace is processed and the CF value of the process situation is calculated by using the fuzzy inference method. The result is passed to IDMON for the judgment of action to take based on the decision tree defined in the knowledge base. In Fig. 7 an example of advisory messages of this system is shown. These messages are displayed on the CRT beside the operator. The CRT messages tell what is the situation of the blast furnace and what to do.

In this application, the knowledge is represented as a decision tree. The inference is carried out every 1 min, and the total time required for data acquisition, processing, and inference is less than half a minute. The optional parameters for the hypotheses are used to write down in the file which nodes are passed through.

6.2. Fault Diagnosis of Surface Inspection Machine¹¹⁾

This system performs a troubleshooting of the surface inspection machine. Fig. 8 shows the outline of the system. In this system the knowledge is represented by a tree and matrix. The tree represents the faulty situation category such as "No detection", "Noise at a certain channel", and "Noise at all

Fig. 6. Block diagram of expert system for blast furnace.

Fig. 7. Example of the guidance display.

channels". In each category the cause-symptom relation is represented in the form of matrix. It is difficult to write a tree which identifies a conclusion following a certain path. Then the cause-symptom matrix is constructed, where each element denotes the strength of the relationship. The inference engine first narrows a category and after that the CF values of the conclusions are calculated by referring to the matrix and the most plausible conclusions are shown. The inference procedure is carried out by Q & A with the operator and the external program is used for Q & A process and the result is passed to IDMON.

7. Conclusions

We introduced a personal-computer based AI tool, named "IDMON" which is suitable for the building of diagnosis type expert systems. The tool has power-

ful abilities to work cooperatively with other softwares, and it can be used as an on-line expert system. By using this tool it is possible to construct expert systems such as automatic process monitoring systems, automatic action guidance systems, off-line interactive consultation systems.

We have applied IDMON to the fault diagnosis problem of an inspection machine and the action guidance of the blast furnace. These applications have shown that the knowledge processing facilities in IDMON is quite useful.

The difficulty is said to exist in the knowledge acquisition. The knowledge is usually logically ill-structured, and it is quite difficult to represent the human problem solving process in the form of a tree or a matrix. In a sense the construction of expert systems is inevitably the standardization of the experts' task and a long time of investigation is required

Fig. 8. Fault diagnosis system of surface inspection machine using IDMON.

to know what standardization is appropriate. One of the important things for the successful expert system development is that the problem is adequate for the standardization and merits are brought by the frequent, efficient, or sometimes automatic executions of the inference based on the standardized knowledge base. The prototyping approach using IDMON is quite useful for such standardization process.

REFERENCES

- 1) Knowledge-based System Diagnosis, Supervision, and Control, ed. by S. G. Tzafestas, Plenum Press, New York, (1988).
- 2) Building Expert Systems, ed. by F. Hayes-Roth, D. Waterman and D. Lenat, Addison-Wesley Publ. Co., New York, (1982).
- 3) Expert Systems, ed. by P. Klahr and D. Waterman, Addison-Wesley Publ. Co., New York, (1986).
- 4) L. Brownston, R. Farrell, E. Kant and N. Martin: Programming Expert Systems in OPS5, Addison-Wesley Publ. Co., New York, (1985).
- 5) C. Holsapple and A. Whinston: *Manager's Guide To Expert Systems Using GURU*, DowJones-Irwin, IL, (1986).
- 6) G. Kahn, S. Nowlan and J. McDermott: *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **PAMI-7** (1985), 511.
- 7) H. Kumamoto, K. Ikenishi, K. Inoue and E. J. Henley: *Chem. Eng. J.*, **29** (1984), 1.
- 8) E. Henley and H. Kumamoto: *Reliability Engineering and Risk Assessment*, Prentice Hall, NJ, (1981).
- 9) Y. Ishida, N. Adachi and H. Tokumaru: *Trans. Soc. Instrument and Control Engineers*, **22** (1986), 168.
- 10) K. Matuda, N. Tamura, K. Kadoguchi and T. Yabata: Proc. of 33th Annual Conference of Systems, Control and Information Engineers, Inst. Systems, Control and Inform. Eng., Kyoto, (1989), 151-152.
- 11) M. Konishi, H. Narazaki and K. Omura: *CAD & CIM*, **2** (1988), No. 2, 57-60.