TM-0377

An Abstract Signal Flow Model Based Diagnosis Expert System Building Method for Telecommunication Network Maintenance

by
Y. Koseki, S. Wada, and T. Nishida
(NEC)

August, 1987

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

An Abstract Signal Flow Model Based Diagnosis Expert System Building Method for Telecommunication Network Maintenance

Yoshiyuki Koseki, Shin-ichi Wada, and Tetsurou Nishida
NEC Corporation

ABSTRACT

This paper describes an expert system building method for telecommunication network maintenance. The telecommunication network in the future is going to be a widely integrated system, so that identifying a faulty portion would require a very wide range of expertise. The method described utilizes the target system *design-knowledge* represented on an *abstract-signal-flow model*, as well as skilled maintenance technicians' *maintenance-knowledge*. All kinds of knowledge are uniformly described in a network-based representation, which is more flexible than the conventional rule-based representations. Based on this method, a prototype trouble-shooting expert system for a digital electronic switching system has been developed. The system helps maintenance technicians to locate defective equipment and repair it.

1. INTRODUCTION

The telecommunication network is growing to a widely integrated and complex system, so that its maintenance is becoming to require a wide range of expertise. Since the network has to be maintained operational for a long time period, it is difficult to keep sufficient numbers of highly skilled technicians. Moreover, it is becoming impossible to train them to be able to maintain modern complex network systems. Therefore, network equipments, such as switching systems, need to provide sophisticated functions for easy operation and maintenance. The goals of these functions are:

- To help technicians to isolate a trouble and to repair it *rapidly* and *correctly*,
- To shorten the maintenance technicians *training period*,

and ultimately,

- To let untrained persons be able to maintain the network.

Although many efforts have been made toward improving operation and maintenance functions, network maintenance still needs highly trained and skilled technicians. To ease their work load and eventually to achieve automatic maintenance, recent expert system technology, derived from Artificial Intelligence research, has been attracting attention.

Early diagnosis expert systems, like MYCIN, were rule-based systems which used a set of empirical symptom-cause rules [Shortliffe76]. They seemed promising, but in reality, their capabilities are limited because they do not utilize any design knowledge, such as internal structures of target systems.

As a contrast to this approach, the so-called first-principle-based approach was developed [Genesereth84, Davis84]. This employs structural and behavioral descriptions of the target system, and reasons for the diagnosis, without having empirical knowledge. This approach seems a good alternative. However, the authors do not believe it can be applied efficiently to complex systems like telecommunication networks, because it does not utilize skilled maintenance technician's experienced knowledge.

This paper proposes a diagnosis expert system building method which utilizes design knowledge as well as skilled maintenance technician's empirical knowledge. Using both kinds of knowledge, it helps maintenance technicians to find defective components, and to repair them. This paper describes:

- Research results on expert knowledge study based on interviews with a number of maintenance technicians
- An *abstract-signal-flow model* to describe design knowledge of target systems
- Trouble-shooting strategy employed in the method
- Knowledge representation schemes for various kinds of knowledge
- A prototype trouble-shooting system, based on the method for digital switching systems [Koseki87a].

The examples and illustrations are taken from the switching system trouble-shooting expert system, but they are relevant to network maintenance problems.

2. EXPERT'S TROUBLE-SHOOTING KNOWLEDGE

Through interviews conducted with skilled maintenance technicians, the authors found that their diagnosis methods generally follow the flow diagram, shown in Fig. 1.

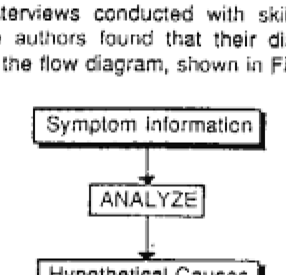


Fig. 1 General Diagnosis Flow

Given the symptom information, the expert thinks of hypothetical causes which could possibly have caused the defect. He enumerates these causes by using structure and behavior knowledge about the target system, and applies a probability order to them, based on his accumulated experience.

With the list of possible causes, he thinks of the subsequent actions to be performed. If the number of the causes is relatively small, the subsequent action may be repairing, such as replacing a printed wiring board. If many causes are still suspected, he thinks of a test to reduce the number of causes. The test to be performed is chosen according to its effectiveness and cost.

After the test execution, he discriminates between the possible causes by interpreting the test result. A structure-and-behavior model is also used at this stage. He repeats this test-and-reduce cycle until he gets a small number of suspected causes.

If there remain a number of possible causes, but he runs out of effective tests, he must try one of the repair actions. He chooses most probable causes based on his past experience. If the symptom disappears after the repair action, the trouble-shooting is considered to be finished. If the symptom still remains after the repair, he has to try the next possible repair in the indicated sequence, until the symptom disappears or until he runs out of possible repairs.

To perform this trouble-shooting work, experienced technicians must have the following kinds of knowledge:

- Design knowledge: structural and functional description of the target system
- Maintenance knowledge: symptom analysis strategies, test procedures, and repair procedures
- Control knowledge: strategy for deciding on a subsequent action

3. TROUBLE-SHOOTING METHOD

The expert system trouble-shooting method follows the diagnosis flow shown in Fig. 1. This section describes the detailed procedures in each step in the flow.

3.1 Abstract-Signal-Flow Model

An *abstract-signal-flow* model is used for suspect generation by symptom analysis and suspect discrimination by test result interpretation.

The model is represented in a directed graph as shown in Fig. 2. This example shows four kinds of signal flow to realize a dial tone connection in a switching system. In this example, the line scan signal for a telephone line comes up to the central processor unit (CPU) through the signal flow (a). To make an interconnection between the originating party and the terminating party, three abstract-signal-flows, (a), (b), and (c), are concerned.

In network maintenance problems, this model should be represented at functional block level; that is, each node on the flow is a functional block. This is because the repair actions are usually to replace printed wiring boards, but not to replace individual LSIs on a board. The entire network representation forms a network of functional blocks connected to each other by signal flows.

(a) Line scan signal

Line -> LC -> E/G Conv -> LM Intf ->
Scn Ctl -> Ins -> SMUX -> Drp -> Q Ctl ->
Bus Ctl -> BIU -> CPU

(b) Tone source control order

CPU -> BIU -> Bus Ctl -> SMC ->
SPC Intf -> T2 CtlM -> T2Sw

(c) Originating subscriber control order

CPU -> BIU -> Bus Ctl -> Ord TX -> SRDQ ->
SDMUX -> Drp -> DrpQ -> SIO -> LM Intf

(d) Voice signal path

Tone Gen -> PMUX -> Ins -> SMUX -> Pad ->
T1 -> S1 -> JHW1 -> JHW2 -> S2 -> T2 ->
SDMUX -> Drp -> PDMUX -> DLSw ->
E/G Conv -> LC -> Line

Fig. 2 Dial tone connection signal-flows

3.2 Symptom Analysis

Most of the symptoms can be analyzed by using the abstract-signal-flow model. For example, to send a dial tone to an originating subscriber, the four abstract-signal-flows, shown in Fig. 2, must not be defective. If there is a no-dial tone symptom observed, there must be a failure in a functional block along with these signal flows. Therefore all of the functional blocks along the flows are enumerated as suspected causes.

When the symptom is a parity-error or a pilot-error, the suspected blocks are considered as being along the signal flow from the parity/pilot generator to the checker. In this case, a relatively small number of blocks is determined as suspects.

Other causes, such as software problems which cannot be inferred by using the flow model are also enumerated. This is done by directly searching the related causes in the knowledge network described in the following session.

3.3 Effective Test Selection and Execution

After the suspect generation is carried out, all of the tests, which are effective to reduce the number of the suspected units, are selected. The word *test*, means any actions to reduce the number of suspected causes here. Therefore, tests may be to investigate the symptom concentration, to run a diagnostic test, to use special test equipments, and so on.

These tests are sorted out by priority. The one with the highest priority is recommended to the user as the most effective test for the trouble. Therefore, novice users can mechanically choose the most highly recommended test, while expert users may ask for a list of all possible tests and select one himself.

Most of the test results can be interpreted by using the abstract-signal-flow model. This is because most of the tests used in network maintenance are signal path tests. A signal path test applies a test signal to a specified point in a network and the signal output at the other point is observed. The test result indicates whether or not functional blocks along a certain signal path are faulty.

Test results in the other categories are interpreted, by executing special procedures provided for the tests themselves.

4. KNOWLEDGE REPRESENTATION

All kinds of knowledge are uniformly represented in a network structure. A network consists of a set of *objects* and their *relations,* as shown in Fig. 3.

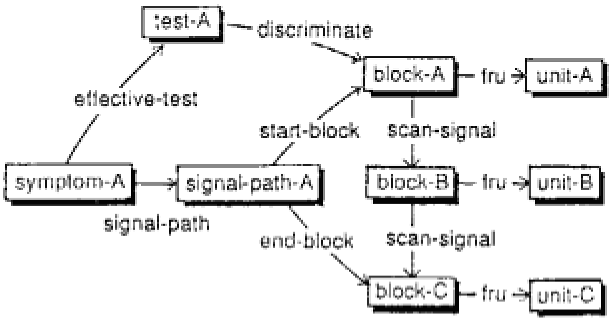


Fig. 3 A semantic network

Each *object* represents a certain lump of knowledge pertaining to a concept or a physical object. That is, it may represent a functional block, a field-replaceable-unit (fru), a symptom, or a test.

Relations literally represent relations between these objects. Some relations are used for diagnostic inference. For instance, relations representing signal flows among functional blocks *(scan-signal* in Fig. 3), are used to enumerate suspected blocks according to the symptom information.

Some relations form a hierarchy of objects. For instance, an object which represents a general concept of duplicated functional blocks is a *super object* of individual functional-block objects. Objects to represent knowledge about tests and symptoms also form a similar hierarchy.

Through these hierarchical relations, lower level objects can *inherit* knowledge concerning their super objects. Therefore, common knowledge about a class of objects must be represented only once in the class representing object. In this way, a compact knowledge representation was achieved.

4.1 Design knowledge representation

Design knowledge is a logical and physical structure description , based on the abstract-signal-flow model.

The logical structure description represents properties of all the functional blocks and the logical relations (signal flows) between the blocks. Functional blocks are represented as objects, and logical connections are represented as relations as shown in Fig. 4.

The physical structure description is also described in the design knowledge. Physically replaceable units are also represented as objects in the network, connected to functional blocks by relations. These relations are used to infer actual locations of physical units, such as packages and cables.

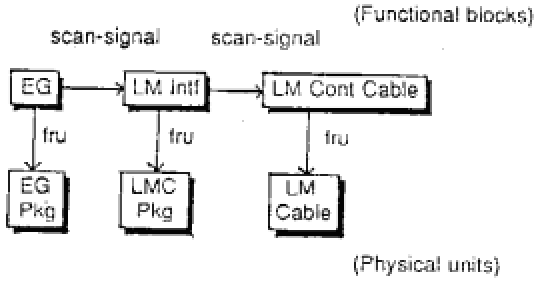
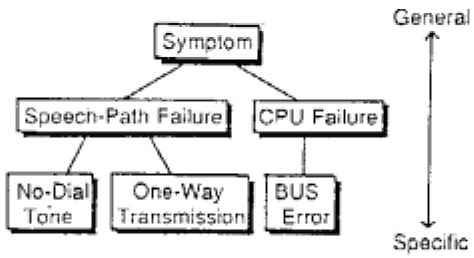


Fig. 4 Design knowledge

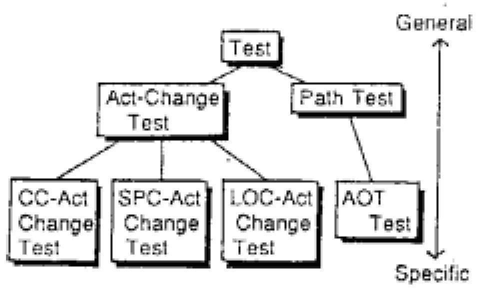
4.2 Maintenance Knowledge Representation

This is heuristic knowledge which distinguishes the expert system from a conventional diagnosis program. Maintenance knowledge consists of symptom knowledge, test knowledge, and their relations with functional blocks in design knowledge.

Individual bits of test and symptom knowledge form a hierarchical tree. As shown in Fig. 5, a higher object for each concept tree is general knowledge of symptom or test. Lower objects describe individual symptom or test knowledge. Utilizing the multiple inheritance mechanism through the relations, attributes in a higher object can be used by lower objects. Therefore, the knowledge-base becomes compact and modifiable.



(a) Symptom knowledge hierarchy



(b) Test knowledge hierarchy

Fig. 5 Maintenance knowledge

Symptom objects hold symptom specific knowledge such as procedures to gather symptom parameters for suspected block generation.

Test objects hold test specific knowledge such as conditions to be effective, test execution procedures, and result interpretation strategies.

4.3 Control Knowledge

Unlike other rule-based diagnosis expert systems, this system does not use a general inference mechanism like forward reasoning or backward reasoning. Instead, it uses more domain specific control mechanisms, written in Prolog procedures.

If the entire amount of inference knowledge is written in a rule style utilizing a general reasoning mechanism, the control mechanism must be embedded into the rules, so that knowledge base becomes difficult to be manipulated.

5. PROTOTYPE SYSTEM: SHOOTX

Using the method described so far, a prototype for trouble-shooting digital switching systems was developed [Koseki87a]. This section briefly describes the system.

5.1. System Operation

When a fault is found in a switching system, a maintenance technician starts a trouble-shooting session with the expert system by selecting the appropriate symptom in the menu.

With the logged error messages, as well as system configuration and status information, it infers the suspected causes.

When there are many suspected causes, the expert system suggests appropriate tests in order to discriminate a faulty portion. Some tests, which can be performed by sending commands to the switching system, are automatically carried out and the results are recorded. The test results are then interpreted to reduce the number of suspected causes. This is repeated until a small number of suspected blocks can be identified.

When the expert system runs out of any appropriate tests, it suggests an appropriate unit replacement or other repair action, showing the unit location graphically. It gives detailed instructions on how to replace the unit and send appropriate commands.

5.2 User Interface

A graphics-oriented user interface is provided in a multiple window environment, as shown in Fig. 6.

It shows detailed location of the unit in the windows. The floor layout shows the frame location on the floor, the frame layout shows the module location in the frame, and the module layout shows the package/cable location in the module. User interaction is accomplished through the menu, using the *mouse* pointing device.

5.3 Implementation

An expert system tool environment, called PEACE (PROLOG based Engineering ApplиCations Environment) has been developed and used on an engineering work-station [Koseki87b]. It is a semantic-network based knowledge representation system, built on logic programming language PROLOG. It supports multiple programming paradigms, such as:

- Object-oriented programming style with multiple inheritances through user defined relations between objects.
- Data-oriented programming style on *slots* of objects.

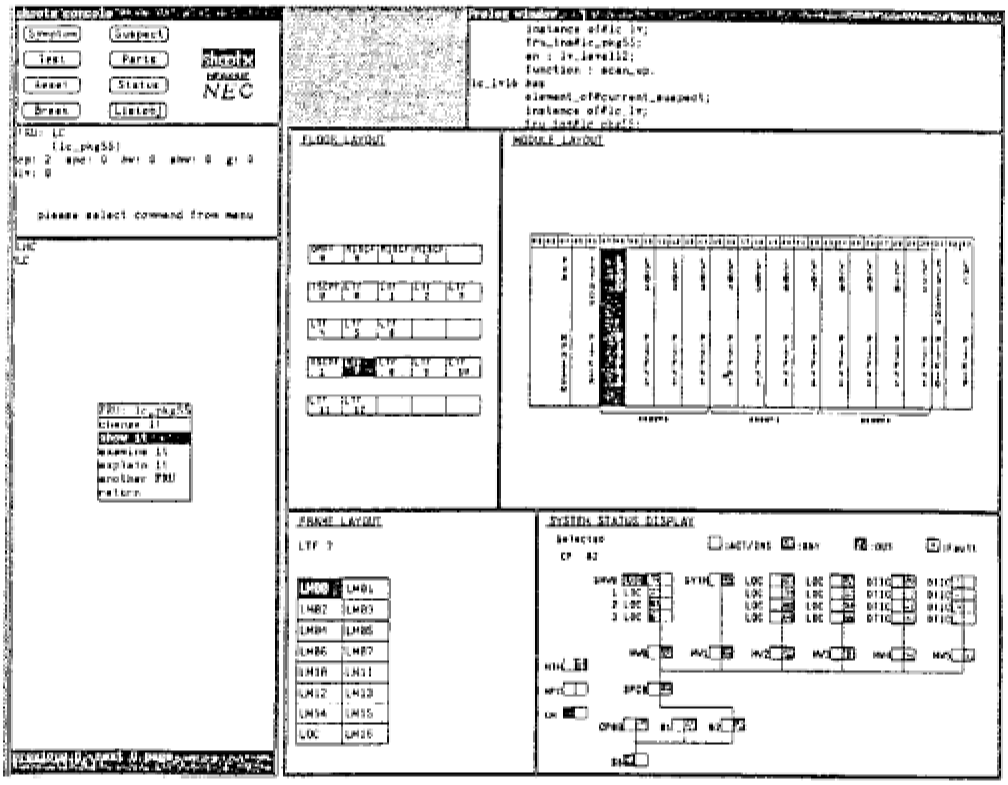


Fig. 6 User interface screen

Logic-programming style using PROLOG pattern matching and a backward chaining mechanism with backtracking.

Figure 7 shows an example description for a function block in PEACE.

```
eg ::
    super_type # func_block;
    control_path # lm_intf;
    voice_path # dlsw;
    dual_single # single_function_block;
    en_concentration : eg;
    fru # eg_pkg.
```

Fig. 7 An example functional block description

In this description, # denotes the relation with the other object. The inheritance can be made through these multiply defined relations.

6. CONCLUSION

Based on the described method, a prototype system was completed and it has been in a demonstrative evaluation phase. In this period, it has been demonstrated to many NEC operation and maintenance technicians and design experts. By operating it themselves, they have realized that it greatly improves operation and maintenance efficiency.

Unlike other rule-based systems, the different kinds of knowledge are clearly classified and are explicitly represented in a unified network. Therefore, their development and maintenance task is relatively easy.

However, the authors feel that future research is necessary to achieve easier knowledge acquisition. Design knowledge is acquired from system designers, whereas, diagnostic knowledge is acquired from field maintenance technicians. Therefore, knowledge engineers have to understand their knowledge and interpret it to develop the knowledge-base. For efficient and error-free knowledge-base development and maintenance, future research must be made to make knowledge-base directly manipulatable by these different kinds of experts.

Currently, an effort to add some learning capability to the system is being made. It is hoped that this will improve system flexibility in the future.

ACKNOWLEDGEMENTS

The authors thank all of the members in the expert system implementation team for enabling it to be realized. The authors thank Mr. K. Toudou, Mr. H. Igarashi and their colleagues for their patient cooperation on the knowledge acquisition study. Also, deep appreciation is expressed to Mr. A. Kitamura, Mr. T. Ando, Mr. Y. Souda, Mr. H. Mori, and Dr. S. Goto for continuous encouragement and support. The PEACE project is affiliated with the Institute for New Generation Computer Technology (ICOT). The authors would like to thank Mr. Y. Fujii of ICOT for his support and guidance.

REFERENCES

[Shortliffe76]
E. J. Shortliffe, *Computer Based Medical Consultations: MYCIN*, Elsevier, New York, 1976.

[Genesereth84]
M. R. Genesereth, "The Use of Design Descriptions in Automated Diagnosis," *Artificial Intelligence* 24, 1984, pp. 411-436.

[Davis84]
R. Davis, "Diagnostic Reasoning Based on Structure and Behavior," *Artificial Intelligence* 24, 1984, pp. 347-410.

[Koseki87a]
Y. Koseki, S. Wada, T. Nishida and H. Mori, "SHOOTX: A multiple Knowledge Based Diagnosis Expert System for NEAX61 ESS," *proc. of International Switching Symposium 1987*, Phoenix, March 1987, pp. 78-82.

[Koseki87b]
Y. Koseki, "Amalgamating Multiple Programming Paradigms in Prolog," *proc. of Tenth International Joint Conference on Artificial Intelligence*, Milano, August 1987.

BIOGRAPHY

Mr. Yoshiyuki Koseki received his B.S. in Computer Science from Tokyo Institute of Technology in 1979, and his M.S. in computer science from the University of California at Los Angeles in 1981. He joined NEC Corporation in 1981, and is a supervisory research member of the Application System Research Laboratory, C&C Systems Research Laboratories. His research interests are in artificial intelligence applications, including knowledge representation, knowledge acquisition, and learning. He is a member of IEEE and of IPSJ (Information Processing Society of Japan).

Mr. Shin-ichi Wada received his B.S. and M.S. in Electrical and Communications Engineering from Tohoku University in 1982 and in 1984 respectively. He joined NEC Corporation in 1984, and is a research member of the Application System Research Laboratory, C&C Systems Research Laboratories. His research interests are in artificial intelligence applications, including knowledge representation and automated reasoning. He is a member of IPSJ (Information Processing Society of Japan) and of JSAI (Japanese Society for Artificial Intelligence).

Mr. Tetsurou Nishida received his B.S. and M.S. in Information Science from Tokyo Institute of Technology in 1979 and in 1981 respectively. He joined NEC Corporation in 1981, and is a member of the Software Development Department, Integrated Switching Development Division. His research interests are in artificial intelligence applications to telecommunication networks and human oriented interfaces. He is a member of IPSJ (Information Processing Society of Japan).