TR-663

An ATMS-based Knowledge Verification System for Diagnostic Applications

by

T. Tanaka & K. Ishikawa (Toshiba)

July, 1991

ICOT

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

(03)3456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

TITLE

An ATMS-based Knowledge Verification System for Diagnostic Applications

Tatsuji Tanaka and Keiko Ishikawa

Heavy Apparatus Engineering Laboratory
Toshiba Corporation

1.Toshiba-cho, Fuchu-shi, Tokyo 183, JAPAN

Masatoshi Yamao

Computer Application Systems Department
Toshiba Corporation Fuchu Works

1.Toshiba-cho, Fuchu-shi, Tokyo 183, JAPAN

ABSTRACT

Knowledge acquisition and verification is a critical bottleneck of expert systems. In particular, it is difficult to confirm and maintain the consistency of a large-scale knowledge base.

A Knowledge Verification System (KNOV) has been developed for diagnostic applications which helps the process that makes problem solving knowledge complete and consistent. KNOV is a meta-system that regards the diagnostic knowledge as assumption and verifies it by assumption-based reasoning using Assumption-based Truth Maintenance System (ATMS). An architecture with the following features is proposed for knowledge verification:

1)assumption-based reasoning with dynamic testing
2)meta-knowledge definition for verification
3)knowledge consistency using ATMS

KNOV has been implemented by Extended Self-contained Prolog on Personal Sequential Inference machine (PSI-II) developed by Institute for New Generation Computer Technology, Japan (ICOT).

The paper shows a verification example using KNOV for the Electric Power System's diagnosis knowledge base. The validity and effectiveness of the ideas for knowledge verification are confirmed by applying KNOV to the diagnosis system of Electric Power Systems and a Computer-Center Fault Recovery System which are used in the field.

1. Introduction

In building an expert system, methods and tools are needed for extracting knowledge from experts, structuring the knowledge, verifying its consistency, and validating it based on the needs of the user. This continuous process is called knowledge acquisition and knowledge verification. It is the bottle-neck of building expert systems[1-9].

1.1 Knowledge verification

The purpose of knowledge verification is to solve the problem correctly by refining incomplete or conflicting knowledge into complete and consistent knowledge[10,11]. The verification process is shown in Fig.1; it comprises the following steps:
1) Error detection and localization, 2) Modification of the conflict knowledge, 3) Verification of the modified knowledge.
In each of these steps, experts use the heuristics and strategies for verification which correctly and efficiently detect and repair the knowledge base error.

The knowledge used in the verification process is as follows:

1. Test case and conflict detection knowledge are used to detect inconsistencies in the knowledge base.
2. Information for knowledge generation is used to generate alternative knowledge and dissolve the inconsistency in the knowledge base.

This reflects deep knowledge in the application's domain, such as the functional model and behavioral model and design knowledge of the system. This knowledge is usually ambiguous and not explicitly represented by the expert.

There are two methods of verifying the knowledge base: static testing and dynamic testing[10]. Static testing analyzes the relationship between the knowledge description element (such as the antecedent condition and the consequent proposition of the rule and related facts), and detects the logical conflict between them[12]. Although syntactic errors and logical conflicts of knowledge description can be detected using static testing, semantic errors cannot. In this case, dynamic testing is needed, which actually infers the knowledge base and detects conflicts within it by checking the inferring process and the result.

1.2 Diagnosis system

It is important to restrict the problem domain when acquiring and verifying the knowledge. Diagnosis of industrial plant and process control applications is chosen as the problem domain. The knowledge structure in this domain is the same as state abstraction in a generic task[9]. The knowledge in this task represents the relationship between the functional element and the system function, and reflects the structure of the system.

In this paper Electric Power Systems diagnosis is chosen for an example[17]. The Electric Power Systems transmit electric power from the generator to consumers and consist of transmission lines, transformers, and buses, each of which is connected with a connective switch comprising circuit breakers and line switches. The connective switch is equipped with protective relays to detect accidents in the Electric Power Systems. Relays open the connective switch to cut off the faulty area from the normal supply area to prevent it from extending. When an accident occurs, the diagnosis system for the Electric Power Systems detects the faulty area and the kind of fault from data of the activated relays and the state of the Electric Power Systems.

Fig.2(a) shows an example of Electric Power Systems configuration and the situation when the faulty accident happened. The diagnosis system determines the faulty facility and the kind of fault based on the function of each protective relay, why they act, when they act, and what factors cause their actions. Accident case-1 in Fig.2(a) indicates the accident with '78s' relay activated in both 'sw2' and 'sw4'. This accident situation is caused by a fault in b_bus. Fig.2(b) shows a diagnosis rule concerning with the relay '78s'.

2. Knowledge verification architecture

The knowledge verification architecture for diagnostic applications is developed. The architecture has the following main features:

1)assumption-based reasoning with dynamic testing

2)meta-knowledge definition for verification

3)knowledge base consistency maintenance using ATMS

The knowledge verification architecture is shown in Fig.3. This system regards the diagnostic knowledge to be verified as assumptions. It verifies the assumptions and refines the unreliable knowledge base into a consistent one by using meta-knowledge for verification. Verification meta-knowledge consists of a test-case set, conflict detection knowledge and assumption generation knowledge.

The system is a meta-system which contains an inference engine for diagnosis, which is controlled by the verification meta-knowledge. It consists of a problem solver which performs assumption-based reasoning for verification, and a knowledge manager which maintains knowledge base consistency using ATMS[13].

The problem solver consists of an assumption test part and an assumption generation part. The assumption test part checks and detects the inconsistency in the knowledge base by treating it as a set of assumptions using the test-case set and the conflict detection knowledge. If any inconsistency is detected, the assumption generation part creates alternative knowledge to replace the inconsistency by using assumption generation knowledge, so that the modified knowledge base is consistent.

The knowledge manager receives inference data and inconsistency information from the problem solver, and stores them in terms of the inference network, and maintains their consistency using ATMS. In the assumption generation phase, this information, stored in the inference network, is used to avoid generating the rules that have already been checked and registered as inconsistent.

3. Meta-Knowledge for verification

The test-case set, conflict detection knowledge, and assumption generation knowledge are used to control the verification process. The contents of the verification knowledge is investigated and its representation is defined. However, this knowledge is given and its correctness and validity are determined by an expert in the problem domain. By representing this knowledge explicitly, an expert can estimate the completeness of the knowledge base and accumulate the heuristics and strategies needed to refine the conflicting knowledge base as a rule.

3.1 Meta-knowledge for assumption test

When verifying the assumptions (knowledge), an expert prepares the criteria which shows the correct result of inference. One of these criteria is the test case. The following items must be defined as a test case:

1)equipment and device organization and their status in the target plant control system

2)phenomena and situations concerning faults or accidents

3)the cause of faults or accidents, such as breakdown of devices and parts

To determine the conflict contained in the knowledge base, the more abstract criteria than the test case is needed. This criteria is formalized as conflict detection knowledge. The knowledge defines the inconsistency of the inference process with the status of the target system to be diagnosed. Inconsistency detection of diagnostic knowledge is performed efficiently by using a test-case set together with conflict detection knowledge.

The test specification must be determined from the viewpoint of the functions required for the system. The test-case set and error detection knowledge are determined by the purpose of the diagnostic system. They must have the following characteristics: 1)correctness, 2)completeness (covering all faults or accidents), 3)consistency (no mutual conflicts).

3.2 Meta-knowledge for assumption generation

Assumption generation is assumed to be a diagnostic rule creation process so that the created rules satisfy all the test cases. This process comprises the rule reconstruction by selecting alternative primitives so that the modified knowledge infers the correct result defined by the test case. Primitives are the basic elements of the rule and the conceptual knowledge of the problem domain.

For assumption generation, the problem solver creates alternative rules to the conflict rule using the following assumption generation knowledge, which represents the expert's idea for rule refinement:

1)concreteness: explore the rule in detail which serves the specific situation

2)abstraction: generalize the specified rule or abstract the concrete rule which has common or similar characteristics

3)selection of alternatives: select alternative primitives on behalf of conflict primitives

Assumption generation knowledge also includes the diagnostic knowledge structure as the constraints for rule generation. This meta-knowledge reflects the deep knowledge on which the problem domain depends. Fig.4 shows an example of the assumption generation knowledge and process for Electric Power Systems diagnosis. In this example assumption generation knowledge is represented by the hierarchical structure of conceptual knowledge of the domain of electric power systems. Using this knowledge, the system replaces the primitive "at the side of the line" in the initial rule with "at the supply side of the line" and "at the receiving side of the line" and two specific rules are generated. First the system searches the concept "line" in assumption generation knowledge, which is broken down into more concrete concept "supply side" and "receive side". New rules are then created.

4. Knowledge consistency maintenance based on ATMS

The diagnostic rules are defined by any combination of primitives that represents an antecedent condition or consequent proposition element of the rule. The set of rules included in the knowledge module is modeled by an AND/OR circuit consisting of primitives. The detection of a conflict or incomplete element in the knowledge is considered to be a multiple-fault diagnosis of an electric circuit[14]. Thus the ATMS can be used for the detection and maintenance of conflict rule in the knowledge base. In case of knowledge verification, an assumption is an element of diagnostic knowledge (rule or primitive), a model is a judgement by an expert, and behavior prediction is a test case.

In using the ATMS to maintain knowledge consistency, we define the ATMS node as described below, and define justification, environment, context and label to be the same as ATMS[13].

<ATMS node>

a. premises: test case, conflict detection knowledge

b. assumptions: diagnostic rule or primitive

c. derivations from assumptions or others

The basic idea of knowledge consistency maintenance using ATMS is shown in Fig.5. In this case primitives A1 and Z are assumptions. A1 is an antecedent condition and Z is the consequent proposition of the diagnostic rule. Test cases a1, a2, a3 (phenomena), d1, d2 (cause of fault), and conflict detection knowledge are given as premises (facts).

In accordance with the inference progress, the derived node is created and added to the inference network. At this time an inferred assumption set is stored as an environment in the derived node. In Fig.5, two derived nodes which have environments {A1,Z} are created according to the two inferences, which refer to facts a1 and d1, or a2 and d2.

The problem solver tests the assumption and if it detects any conflicts, it registers the conflict assumption set (rule) into the ATMS no-good database. In Fig.5, conflict is detected by inferring rule C with input data d2. Consequently, the node(*) is made to be contradictory, and related assumptions are made to be false from the super- and sub-relationship of the inference network. In this case, the environment {A1,Z} of the conflicting node(*) is registered to the no-good database of the ATMS. This means that the diagnostic rule consisting of the primitives A1 and Z is in conflict. The process of inference is stored using ATMS and the problem solver can control assumption generation, therefore, which means the failed rule generated previously will not be generated again.

Although the number of assumption generations and the number of inferences for testing assumptions increase exponentially with the complexity of the diagnostic knowledge, the test can be managed efficiently using the ATMS mechanism.

5. Implementation

KNOV has been developed based on the knowledge verification architecture. KNOV has the restriction that the conflict part of the knowledge is to exist only in the rule part, not in the primitive or data (facts) part. KNOV has been implemented by ESP on the workstation PSI-II developed by ICOT[15,16].

Fig.6 shows the man-machine interface of KNOV. Window (a) is the control window where a user starts up the verification process and checks verification information. Window (b) shows the result of a diagnostic inference. Window (d) shows the conflict detection knowledge used to detect the conflicting rule shown in window (c). In this example, KNOV generates consistent rules shown in window (e).

6. Evaluation of KNOV

The validity and effectiveness of the system was evaluated as described below and applied to the diagnosis system of Electric Power Systems and Computer-Center Fault Recovery Systems used in the field[17,18].

An evaluation of KNOV is shown in Table 1. The diagnostic knowledge and verification meta-knowledge are shown in Table 1(a). This knowledge is for the Electric Power Systems diagnosis described in this paper. The evaluation results are shown in Table 1(b). We confirmed that all the conflict rules in the knowledge base were detected and modified correctly.

In Table 1(b), each field of assumptions means the following: a) generated rules by assumption generation knowledge, b) conflict rules detected by ATMS, c) conflict rules detected by inference, d) consistent rules verified by ATMS and inference. The processing time for verification is less than 60 seconds in any evaluation case shown in Table 1(b). Thus KNOV can be used in real-sized problems with reasonable performance.

7. Conclusion

An ATMS-based knowledge verification architecture for diagnostic applications was proposed and the KNOV on which it is based was developed. The system which supports a knowledge verification process did not exist and the KNOV is an entirely new system that supports the process. Our research is a step toward building an integrated knowledge base developing environment.

By using the KNOV, the complicated knowledge verification process, which depends on the experience and perception of experts, can be systematized and performed automatically. Thus, the system is very useful for building and maintaining knowledge-based systems. In this system, the verification heuristics and strategies are represented in terms of verification meta-knowledge. Thus the KNOV is so flexible that an expert can define the verification meta-knowledge which depends on an application domain independent from the verification procedure. To generate alternative rules to repair the conflicting knowledge base, verifying their correctness and confirming that they have not caused side effects is the most time-consuming process for experts. The system manages these complicated processes and information for verification systematically by using ATMS.

Even with the benefits, acquiring verification meta-knowledge is the problem which is left for future considerations in practical use of the KNOV. Verification meta-knowledge is a higher abstraction of diagnostic knowledge and methods of its use. Thus, it is difficult to show explicitly how to acquire, formalize and represent the meta-knowledge for each application.

Acknowledgements

This research has been carried out as a part of the Fifth Generation Computer Project. We would like to thank Kenji Ikoma, the chief of the Fifth Research Laboratory of ICOT, for helpful discussions and encouragement.

References

[1] H.Hayes-Roth, D.A.Waterman and D.B.Lenat, Building Expert Systems, Addison-Wesley, 1983.

[2] B.R.Gaines, "An Overview of Knowledge-Acquisition and Transfer," Int. Journal of Man-Machine Studies, vol.26, pp.453-472, 1987.

[3] R.Davis and D.B.Lenat, Knowledge-Based Systems in Artificial Intelligence, McGraw-Hill, 1982.

[4] J.Boose, "Personal Construct Theory and the Transfer of Human Expertise," in Proc. of the National Conf. on Artificial Intelligence, Austin,Texas, 1984, pp.27-32.

[5] J.H.Boose and J.M.Bradshow, "Expertise Transfer and Complex Problems: Using AQUINAS as a Knowledge-Acquisition Workbench for Knowledge-Based Systems," Int. Journal of Man-Machine Studies, vol.26, pp.3-28, 1987.

[6] J.Kahn, S.Nowlan and J.MacDermott, "Strategies for Knowledge Acquisition," IEEE Trans. of Pattern Analysis and Machine Intelligence, V.PAMI-7,No.5,pp.511-522, 1985.

[7] L.Esheiman and J.MacDermott, "MOLE : A Knowledge Acquisition Tool That Uses its Head," in Proc. of the National Conf. on Artificial Intelligence, 1986,

pp.950-955.

[8] J.MacDermott, G.Dallemage, G.Klinker, D.Marques, and D.Tung, "Explorations in How to Make Application Programming Easier," in Proc. of the First Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop, 1990, pp.134-147.

[9] B.Chandrasekaran, "Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design," IEEE Expert, pp.23-30, Fall 1986.

[10] D.O'Leary and R.O'Keefe, "Verifying and Validating Expert Systems," Tutorial: MP4 of IJCAI-89, 1989.

[11] T.J.O'Leary, M.Goul, K.E.Moffitt and A.E.Radwan, "Validating Expert Systems," IEEE Expert, pp.51-58, June 1990.

[12] T.A.Nguyen, W.A.Perkins, T.J.Laffey and D.Pecora, "Checking an Expert Systems Knowledge Base for Consistency and Completeness," in Proc. of 9th IJCAI, 1985, pp.375-378.

[13] J.De Kleer, "An Assumption-Based TMS," Artificial Intelligence, vol.28, pp.127-162, 1986.

[14] J.De Kleer and B.C.Williams, "Diagnosing Multiple Faults," Artificial Intelligence, vol.32, pp.97-130, 1987.

[15] S.Konda and T.Chikayama, "ESP Programming," ICOT Tech. Rep. TM-134, 1985.

[16] Y.Oota and K.Inoue, "Assumption-Based Reasoning System APRICOT/0 Users Manual," ICOT Tech. Rep. TM-676, 1989.

[17] S.Moriguchi, T.Tanaka and K.Ishikawa, "Expert Shell for Power Systems Diagnosis," in Proc. of Int. Workshop on Artificial Intelligence for Industrial Applications, Hitachi City, Japan, May 1988, pp.585-590.

[18] S.Ishii, T.Abe, Y.Hidai and A.Tanaka, "An Operations Advisor for On-line Computer Banking System with Graphics Interface," in Proc. of IEA/AIE-89, 1989, pp.568-576.

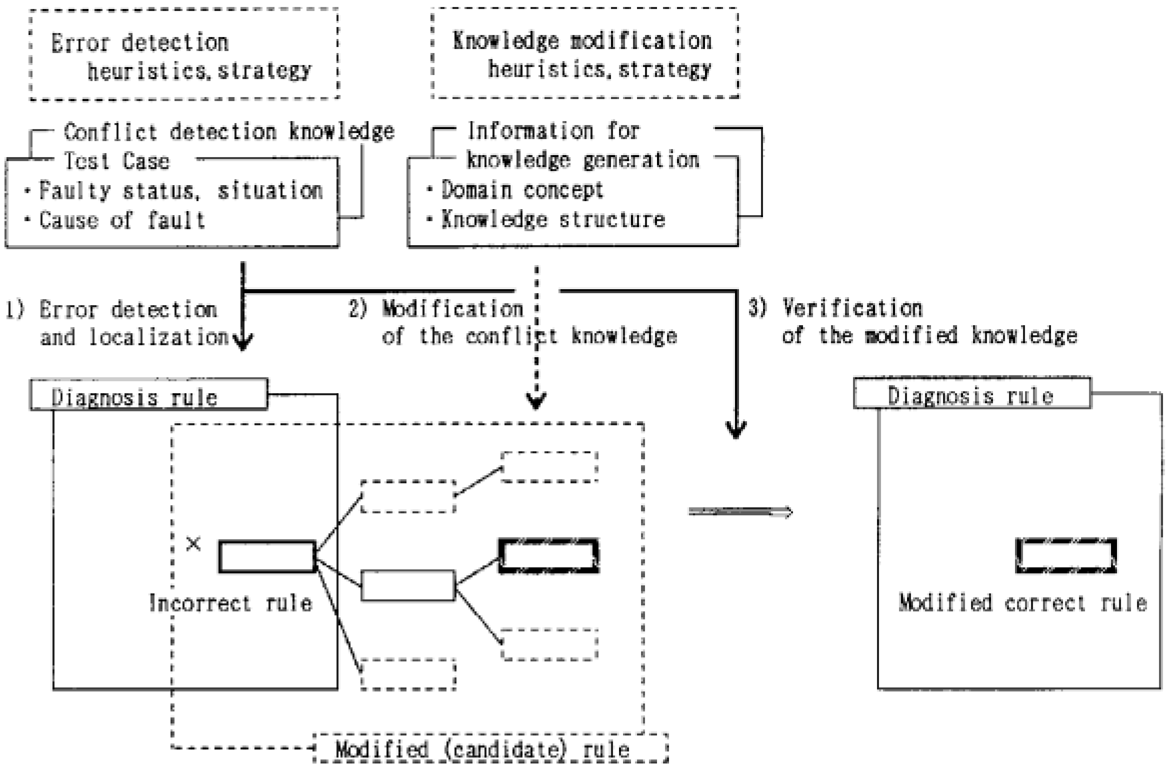
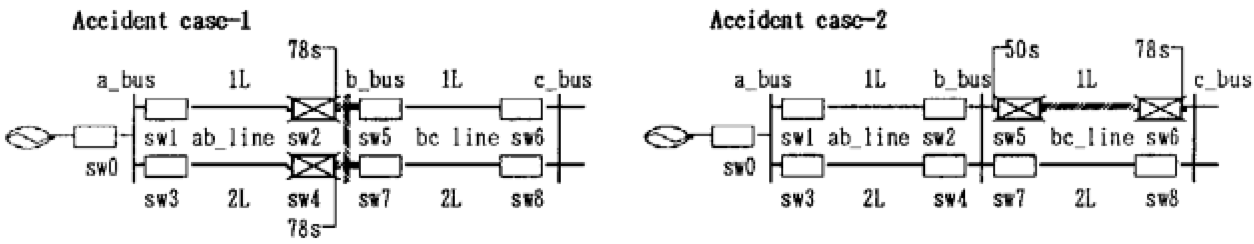


Fig. 1 Knowledge Verification.



Faulty Situation:	activated relay	..78s, 78s
	opened switch	..sw2, sw4
Faulty Cause:	facility	..b_bus
	kind	..short

Faulty Situation:	activated relay	..50s, 78s
	opened switch	..sw5, 78s
Faulty Cause:	facility	..bc_line_1L
	kind	..short

(a) Electric Power Systems Configuration and Daignosis

```
rule 24:
        IF      relay (78s) acts
                at the side of bus (BUS),
        THEN    faulty kind is (short)
                faulty area is bus (BUS).
```

(b) Diagnosis Rule

Fig. 2 Electric Power Systems Diagnosis.

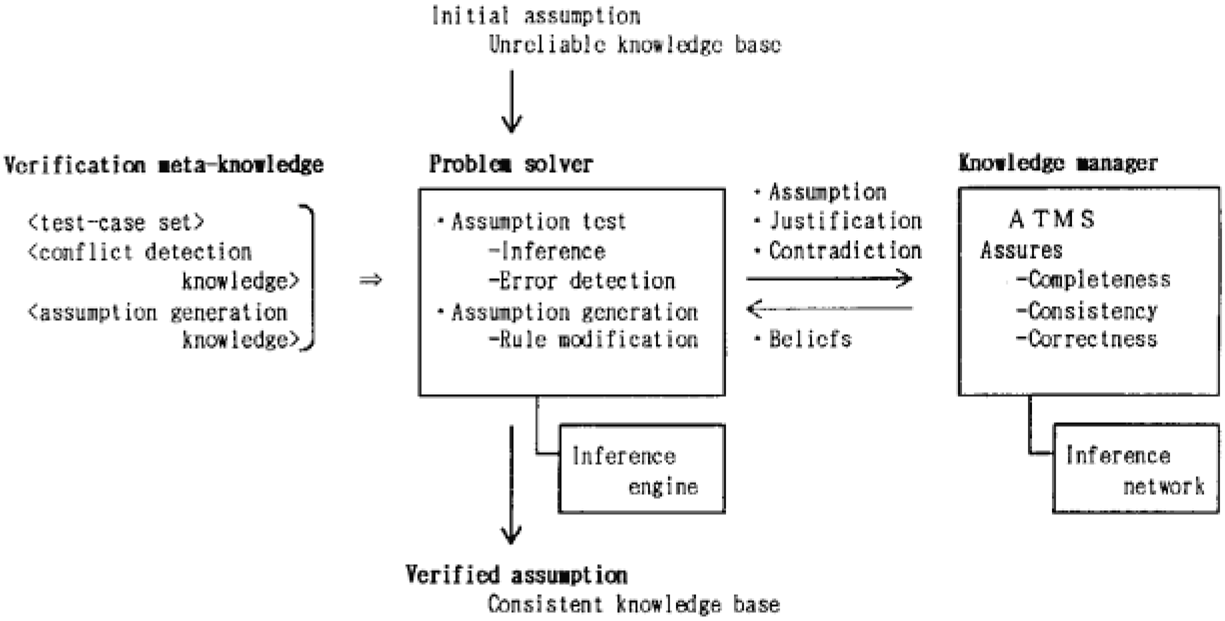


Fig. 3 Knowledge Verification Architecture.

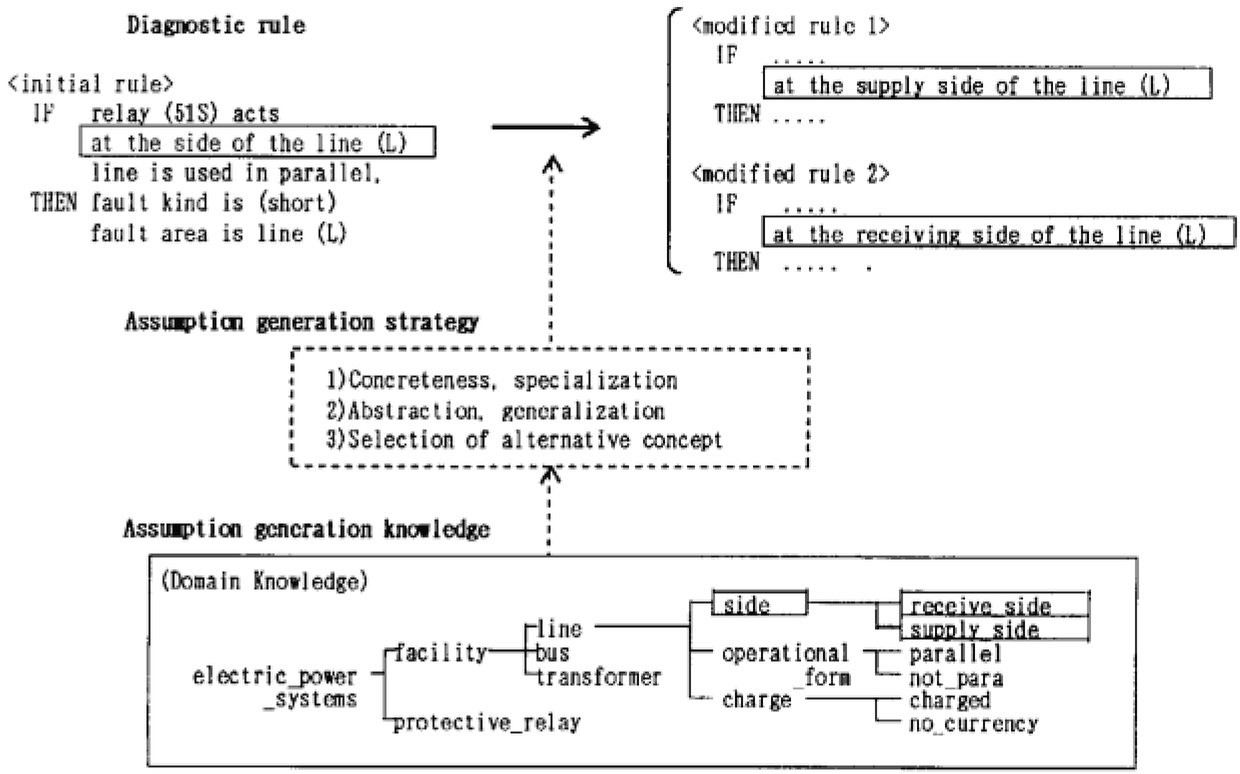
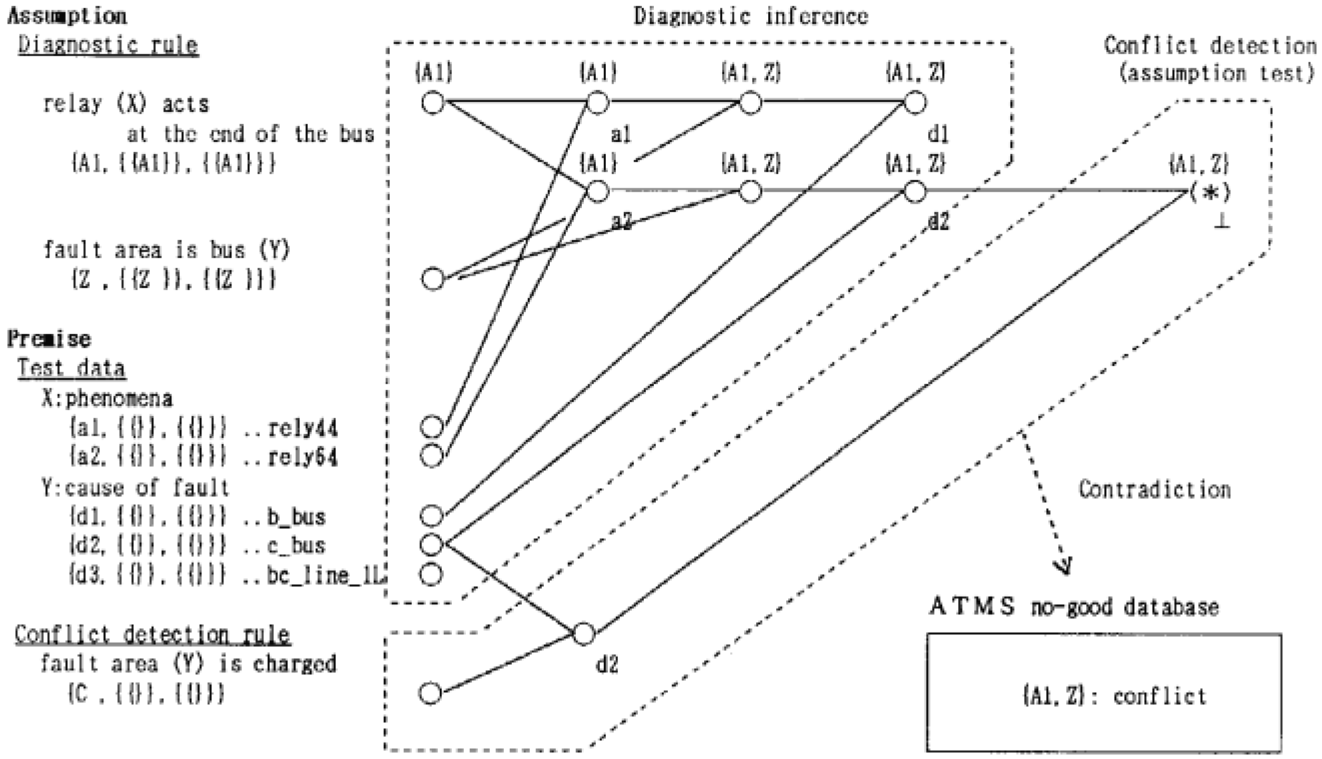


Fig. 4 Assumption Generation.

Diagnostic rule : IF relay (X) acts at the end of the bus, THEN fault area is bus (Y).

Conflict detection rule : IF fault area (Y) is charged, THEN contradiction.

(a) Rule Example



(b) Inference Network

Fig. 5 Knowledge Consistency Maintenance Based on ATMS.

15/E

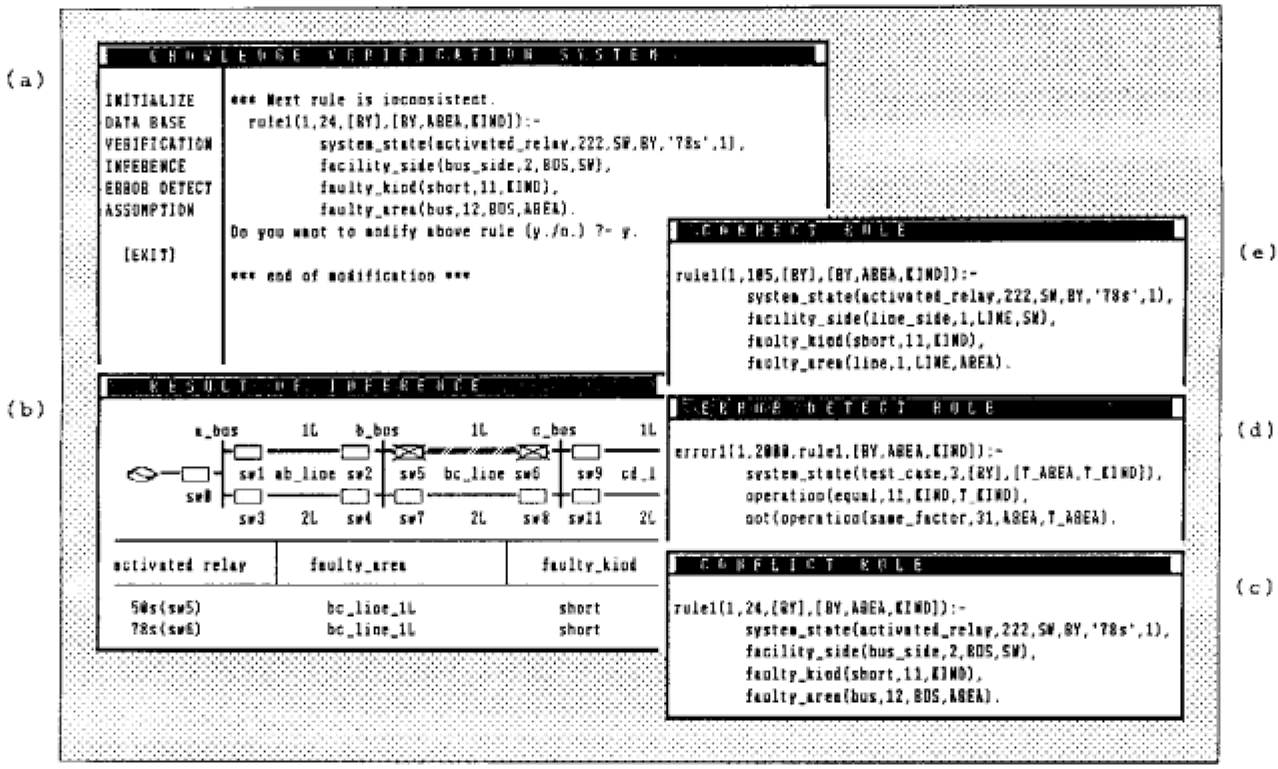


Fig.6 Man-Machine Interface of KNOV.

Table 1 Evaluation of Knowledge Verification.

(a) Diagnostic and Verification Knowledge

Knowledge class	Rule and data	Amount of rules or data
Diagnostic Knowledge	rule module	2 modules
	rule	40 rules
	primitive	73 primitives
	data definition (fact)	114 data (facts)
Verification Meta-Knowledge	test-case set	4 set
	conflict detection knowledge	9 rules
	assumption generation knowledge	64 knowledge (generation:50, constraint:14)

(b) Verification Result

Evaluation Case	Number of Assumptions				Processing Time
	a)	b)	c)	d)	(seconds)
Rule specialization (1)	84	53	28	3	43
Rule specialization (2)	50	26	22	2	51
Rule abstraction	49	38	9	2	28
Selection of alternative concept	101	42	58	1	33

a) = b) + c) + d)

a) .. generated assumptions (rules) by assumption generation knowledge
b) / c) .. conflict assumptions (rules) detected by ATMS / inference
d) .. consistent assumptions (rules) verified by ATMS and inference