PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1984,
edited by ICOT. © ICOT, 1984

A Knowledge Based System for Plant Diagnosis

Hiroshi Motoda, Naoyuki Yamada
and Kenichi Yoshida

Energy Research Laboratory,
Hitachi Ltd.
1168 Moriyamacho, Hitachi,
Ibaraki 316, Japan

Abstract

A knowledge based system for plant diagnosis is proposed in which both event-oriented and function-oriented knowledge are used. For the proposed system to be of practical use, these two types of knowledge are represented by mutually nested four frames, i.e. the component, causality, criteriality, and simulator frames, and production rules. The system provides fast inference capability for use as both a production system and a formal reasoning system, with uncertainty of knowledge taken into account in the former. Event-oriented knowledge is used in both diagnosis and guidance and function-oriented knowledge, in diagnosis only. The inference capability required is forward chaining in the former and resolution in the latter. The causality frame guides in the use of event-oriented knowledge, whereas the criteriality frame does so for function-oriented knowledge. Feedback nature of the plant requires the best first search algorithm that uses histories in the resolution process. The inference program is written in Lisp and the plant simulator and the process I/O control programs in Fortran. Fast data transfer between these two languages is realized by enhancing the memory management capability of Lisp to control the numerical data in the global memory. Simulation applications to a BWR plant demonstrated its diagnostic capability.

1 Introduction

Complex power plants should be equipped with aids to support safe operation and improve availability. In anomalous situations, the plant operators must observe and interpret many signals displayed on control panels and make appropriate decisions as to what is wrong and how to correct it.

Some model is required to perform diagnosis. Generally, this model describes anomaly propagation and can be regarded as knowledge about cause and consequence relationships of anomaly situations (event-oriented knowledge) as is seen in computer hardware diagnoses [Bennett and Hollander 1981], [Shubin and Ulrich 1982], [McDermott and Brooks 1982]. The possibility has been suggested of making diagnoses of nuclear power plants by collecting many pieces of knowledge about causality relationships [Wells and Underwood 1982], [Yoshida et al. 1983]. The reasoning process, however, is one of searching the state space of classifications. In this regards, the approach requires an explicit expression of causality: cause and result relationship.

A second new approach that has been put forward is to use knowledge about system descriptions, i.e. intended structure and expected behavior (function-oriented knowledge) [Genesereth 1982], [Davis et al. 1982].

This approach requires no information about how a system fails. Use of resolution in predicate calculus logic has shown the possibility of generating knowledge directly useful for diagnosis of a nuclear power plant without requiring causality relationship [Yamada and Motoda 1983] as was done in [Genesereth 1982]. Although this approach simplified the task of building diagnostic systems, inefficiency lies in its implementation of knowledge representation and a general inference procedure.

These two approaches are characterized in that the former uses knowledge of anomalous situations, whereas the latter uses knowledge of normal situations. The former is more direct and hence is more efficient, but it requires all anomalous situations be covered for completeness. The latter is not as efficient as the former because it is indirect. However,as it is much easier to describe how the system should work if functioning normally, the latter is more powerful.

This paper introduces an attempt to build an expert system that can use the knowledge of both types for diagnosis of a nuclear power plant. For the proposed system to be applicable for on-line use, knowledge is represented in a frame structure into which fast inference capability is built.

Section 2 describes a general diagnostic procedure. Section 3 explains knowledge representation schemes and section 4 inference capability. Section 5 summarizes results in applications to a boiling water reactor (BWR).

2 Diagnostic Procedure

Plant diagnosis is regarded as the coupled procedures of problem diagnosis and guidance in corrective actions. The problem is then to identify the faulty component(s) from the given symptom(s) indicating an anomaly and to make necessary recommendations for operator actions based on the diagnosis results.

2.1 Identification of Faulty Component(s)

The diagnosis starts when a symptom does not match that expected. The general procedure of diagnosis consists of three steps: suspect pick up, suspect discrimination and test generation. The last step is necessary only when the second one fails to identify the cause. First an attempt is made to use event-oriented knowledge. If the cause is identified, the system turns to the guidance phase. Otherwise, it looks for function-oriented knowledge to further discriminate between the suspects.

Event-oriented knowledge is given as production rules. These are divided into two categories depending on the form of their conditional part: (1) those that have only a logical expression of symptoms (observed data) and (2) those that have at least one intermediate hypothesis (unobserved data). The rules in the first categories are designated check rules. These are grouped into different sets each pertinent to some symptom or component.

One of the check rules in some set is fired first. There is a special set whose rules have to be referred to regardless of the firing condition. Forward chaining is started from all of the check rules in the fired and the special sets. Rules in the second category are used in the succeeding chainings. Simulators are executed when necessary during the course of chaining.

Function-oriented knowledge consists of structure, behavior and status of the plant. Since this knowledge is not directly useful for diagnosis, deduction by formal reasoning is necessary to generate knowledge applicable to diagnosis.

Resolution is used for each of the three steps in the diagnosis. Starting resolution with the component where an anomaly is first detected, a clause consisting of a set of components, a set of sensors and their functional relationships with variables remaining unsubstituted is obtained. It is not necessary to go back to a state in which all the components were normal because the input and output relationships of a normal component is consistent regardless of the value of its input(s). It is possible to use the observed data at a state whose effects propagate through all the components at least once before being detected by some sensor. Execution of the simulator using these observed data shows some discrepancy between the observed and expected value of the symptom. This resolves into a set of suspect candidates.

Next, resolution is used in a similar manner to pick up a minimum set of observable data sufficient to check the operational status of a specified component in the suspects. Again by simulation the observed and expected values can be compared. If there is some discrepancy, the components in the obtained clause are the discriminated suspects. The order of components from which to start resolution is guided by some heuristic knowledge such as redundancy, a priori failure probability, etc. because this provides information about how important a component is in the system or about how likely a component is to fail.

If there is more than one suspect left in the discriminated set(s), an attempt is made to generate tests for further discrimination under the single fault assumption in each set. Changing the selector switch of redundant components or open/close manipulation of some valves are such tests. At this point teleological knowledge such as switchable, achievable etc. is used, in conjunction with the status information such as redundancy, or-state etc., to select the component(s) from which to start resolution in the set. For each of the selected component, resolution results in a clause that contains a set of components, at least one of which is different from those in the discriminated suspects obtained in the previous step. The generated tests are then evaluated to see if they are successful or not.

2.2 Guidance for Corrective Actions

Once the faulty component(s) has been identified, the guidance for corrective actions is straightforward. Rules for guidance are also grouped into sets similar to the check rules in the diagnostic phase.

At each time step of simulation, rules in the special set are searched to categorize the plant behavior. This activates the relevant set to focus on the rules to select. The rules for guidance can be distinguished from those for diagnosis by context change. If there are uncertainties in the identified causes, corrective actions are suggested for each possible cause.

3 Knowledge Representation

The nature of plant diagnoses has the following prerequisites:

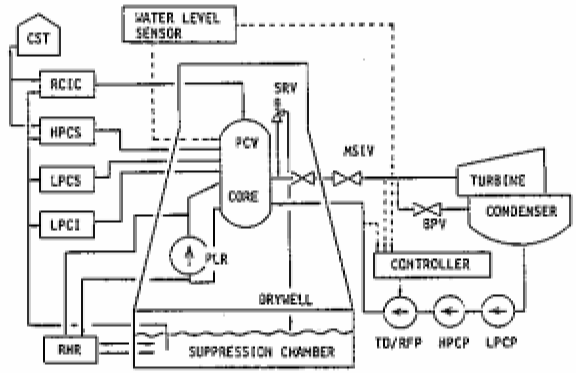
1) Not only static data (e.g. operational status of each component), but time varying process data should be handled appropriately.

2) Inference should be fast enough for the overall diagnosis to be of practical use. As is often the case, real time inference is needed.

3) A numerical simulation program, coded by Fortran or some other language, should be utilized in the inference process.

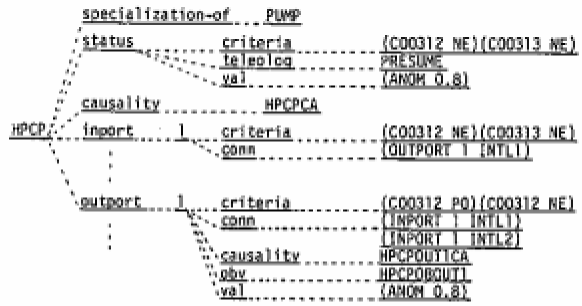
In order to meet these needs, function-oriented knowledge (functional hierarchy, connectivity and behavior) is represented in frame structures corresponding to the real plant schematics. Event-oriented knowledge is embedded around these frames. Along this line, the knowledge base is classified into the following four frames, the component, causality, criteriality, and simulator frames, and production rules.

In the following explanation, examples are taken from a BWR nuclear power plant for which feasibility of this approach has been studied. Figure 1 shows schematically the main components of a BWR. The knowledge dealing with the high pressure condensate pump (HPCP) is given in Fig. 2 by a causality description, and the knowledge dealing with the controller (ATEC, cf. Fig. 8) in Fig. 3 by a function description. In these frames, each symbol and list linked with a dotted line from left to right denote frame name, slot name, facet name, and value pairs or their modifications.

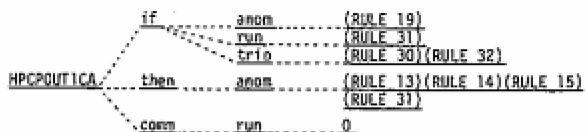


CST	Condensate Storage Tank	PLR	Primary Loop Recirculation System
RCIC	Reactor Core Isolation Cooling System	SRV	Safety Releaf Valve
HPCS	High Pressure Core Spray System	MSIV	Main Steam Isolation Valve
LPCS	Low Pressure Core Spray System	BPV	Bypass Valve
LPCI	Low Pressure Coolant Injection System	TD/RFP	Turbine Driven Reactor Feed Pump
RHR	Residual Heat Removal System	HPCP	High Pressure Condensate Pump
PCV	Primary Containment Vessel	LPCP	Low Pressure Condensate Pump

Fig. 1 Main components of BWR plant system



COMPONENT FRAME



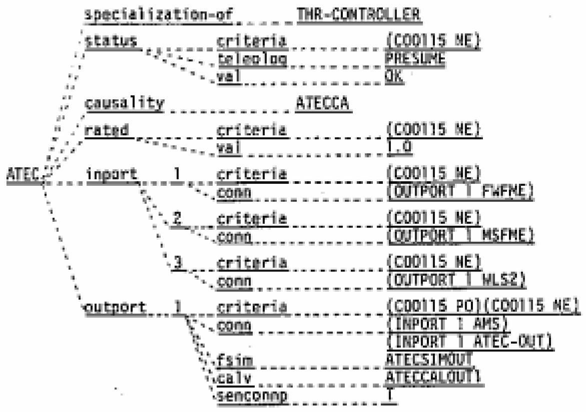
CAUSALITY FRAME

RULE 31

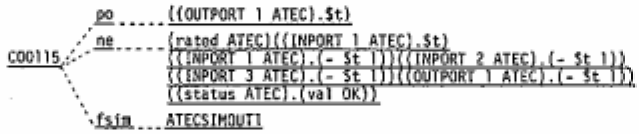
```
(if (cont $t 3.0
        (and (true (value (outport 1 HPCP) run) $t)
             (true (value (outport 1 LPCP) trip) $t)))
    (value (outport 1 HPCP) anom))
(CF 1.0)
```

CAUSALITY RULE

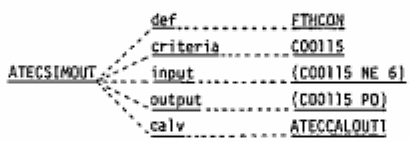
Fig. 2 Examples of frames related to the component HPCP (causality description)



COMPONENT FRAME



CRITERIALITY FRAME



SIMULATOR FRAME

Fig. 3 Examples of frames related to the component ATEC (function description)

(1) Component frame

This frame describes component properties within each functionally hierarchical level. All the component frames have at least four slot names , i.e. specialization-of, status, inport and outport to specify hierarchical structure, operational status, and inport and outport attributes. Static data are stored directly as facet values, and time varying numerical data obtained from both observation and simulation are stored in vectors with error bounds, the pointers to which are stored as facet values. Pointers to criteriality and simulator frames are also stored as values, while pointers to causality frames are stored as slot and/or facet values. In order to control the inference procedure, teleological knowledge is implemented by using a teleology facet.

(2) Causality frame

This frame structure is introduced to realize fast inference capability as a production system. Names of the related causality rules (event-oriented knowledge) are stored as the values of if and then slots for each possible status of the corresponding component or port (e.g. anom, run , trip in Fig. 2, $ for unknown state).

Rules in the if slot have at least one term for this component or port as their conditional part. Rules in the then slot conclude something about this component or port. The conn slot relates the symbols to numerical values (e.g. process data).

(3) Criteriality frame

This frame structure is introduced to realize general inference capability (formal reasoning) upon frame structured knowledge. It specifies inter- relationships among properties of each component frame. These are mainly concerned with inport and outport properties, describing two types of behavior rules: forward and backward rules in which the former relates input(s) to output and the latter output to input(s). Here, criteriality means that the success or failure of satisfying this relationship directly determines the operational status of the corresponding component.

Each inter-relationship is stored over the two slots: the po and ne slots (cf. Fig. 3), each of which contains positive and negative terms respectively when expressed in a conjunctive normal form. It further stores a pointer to the simulator frame if simulation is involved in that relationship. If a time difference exists between properties, it is expressed by using a variable $t assuming a constant time step in each hierarchical level (e.g. $t, (- $t 1) in Fig. 3).

(4) Simulator frame

It describes the name of the function used for simulation. Inputs to this function and its output are specified by referring to the corresponding criteriality frame.

(5) Production rules

There are two types of rules, one for expressing the causality and the other for expressing the guidance, both being event-oriented knowledge. Each rule is assigned a certainty factor. There are several predefined predicates. These include and, or, true, value, last, before, cont, etc. For example, (true A B) means that A is true in situation B; (value A B) means that the value of A is B; and (before A B) means that event A occurs before event B.

The rule for causality shown in Fig. 2 means that if the state HPCP-run and LPCP-trip continues for 3 seconds, it is definite that the HPCP (more precisely the value of the outport 1 of the HPCP) is anomalous. Likewise, the rule for guidance in Fig. 4 means that if the water level of the CST is below its limit and the source of the RCIC is the CST, then its source must be switched to the suppression chamber (SC).

```
RULE 101
        (if (and (last $t)
                 (true (value (outport 1 CST) $v) $t)
                 ( ≤ $v Limit)
                 (true (value (inport 1 RCIC) CST) $t))
             (true (value (inport 1 RCIC) SC) $t))
        (CF 1.0)
```

Fig. 4 Example of rule used for guidance of corrective actions

To reduce the burden of preparing the knowledge in frame structures, function-oriented knowledge is given by predicate calculus. Examples corresponding to Fig. 3 are shown in Fig. 5. Event-oriented knowledge is also given in a similar manner. These two kinds of knowledge are automatically converted to the above frame representation.

The inference program is written in Lisp and the simulators and the process I/O control programs, in Fortran. The fast data transfer among these programs is realized by enhancing the memory management capability of Lisp to control the data in the global memory.

```
(if (and (value (status ATEC) OK)
         (value (rated ATEC) $r)
         (true (value (inport 1 ATEC) $in1)(- $t 1))
         (true (value (inport 2 ATEC) $in2)(- $t 1))
         (true (value (inport 3 ATEC) $in3)(- $t 1))
         (true (value (outport 1 ATEC) $in4)(- $t 1))
         (= $out (FTHCON $r $in1 $in2 $in3 $in4)))
    (true (value (outport 1 ATEC) $out) $t))

(conn (OUTPORT 1 FWFME)(INPORT 1 ATEC))
(conn (OUTPORT 1 ATEC)(INPORT 1 AMS))

(value (status FWFME) OK)
(value (status ATEC)  OK)
(value (rated ATEC)  1.0)

(PRESUME (status ATEC))
(OBSERVABLE (OUTPORT 1 FWFME))
```

Fig. 5 Knowledge expressed in predicate calculus (user input)

4 Inference Capability

Event-oriented knowledge is used in both diagnosis and guidance and the function-oriented knowledge in diagnosis only. The former requires forward chaining capability, and the latter, resolution capability. The inference capability of the system, however, covers more to enhance the generality.

4.1 Inference Capability as a Production System

(1) Forward chaining

The main flow of this procedure is first to get a causality frame attached to a specified component or port. It then picks up rules in the if slot that have the same facet name as the value of the data (e.g. run in Fig. 2) and the $ facet name, and put them in agenda list together with other rules in the special set if any.

Interpreting the predicate of terms in the condition part of these rules, it gets the causality frames, corresponding to the terms in the concluding part of the satisfied rules and recursively calls itself until there is no change in the certainty factor of the conclusion.

(2) Backward chaining

The main flow of this procedure is basically the reverse of the above. It looks for the rules in the then slot of the causality frames for backward chaining.

4.2 Inference Capability as a Formal Reasoning System

(1) Forward chaining

The main flow of this procedure is first to get the criteriality frame attached to the specified component property, second to check all the component properties in the ne slot and then to get the property specified in the po slot of that frame, the value of which is then transferred to the other properties through the connectivity relationships. Simulation programs are automatically invoked and executed when all input values are gathered. A control necessary for this procedure is simply not to conclude the property already inferred.

(2) Backward chaining

The main flow of this procedure is basically the reverse of the above. It first gets the criteriality frame that concludes the specified component property in the po slot and checks whether the component properties in the ne slot of that frame have available data or not. These steps are recursively executed through the connectivity relationships until all available data that lead to those properties have been found.

(3) Resolution

The starting point of resolution is either a component frame or a port value. In the former case, it picks up one of the criteriality frames attached to the component and follows the same procedure as the latter.

Ordinary, there exist many feedback loops in the system description of an actual plant and besides, inter-component feedback loops are formed when dynamic behaviors of each component are discretized. Therefore, it is definitely necessary to implement some kind of a best first search mechanism in order to achieve an efficient resolution procedure.

The main flow of the resolution procedure is controlled by the histories of resolution of each term. The number of resolvent terms does not decrease when connectivity and criteriality are used. Which to use next depends on how the current state has been established. One of the control rules for selection is shown below.

```
If the term to be resolved is
   a negative one,
   it is outport property, and
   it was resolved by using
   criteriality,
then select the criteriality to find
   a resolver.
```

Six other similar rules are implemented for efficient control. If the resolution procedure terminates normally, the result is a clause consisting of terms each of which has a specified teleological knowledge in the corresponding teleology facet.

The purpose of this procedure is to generate knowledge useful for diagnosis. In plant diagnosis, assembling the necessary conditions to calculate the specified component property, picking up a minimum group of observation data to determine the operational status of a specified component and generating a test form, for example, are achieved by this method.

It is also made possible to specify not to execute the simulation program when it is encountered, by which a kind of symbolic simulation is performed.

This frame representation makes the best use of structural information. It helps reduing the unnecessary unifications which would have been required if the knowledge was represented in predicate calculus as in Fig. 5. Efficiency improvement is worth noting. The second example in Sec. 5 ran by the order of two as fast as in the case of the straightforward implementation as in [Yamada and Motoda 1983]. Execution time increases almost linearly with the number of components because of the connectivity nature of a real plant.

4.3 Capability of Uncertainty Treatment

The uncertainties considered in this system are the following two:

1) errors introduced in observed process data and simulation results; and

2) uncertainty inherent to event-oriented knowledge.

Interpretation of equality of the numerical values allows some allowable bounds. Interpretation of inequality assumes a normal distribution of the value with its specified error equal to the standard deviation and includes a statistical evaluation. Uncertainty is propagated by using the evaluation rules for combination, conjunction, disjunction, and implication on the basis of either fuzzy logic or EMYCIN logic by option.

5 Application Study

The foregoing approach was implemented in VAX11 with Franzlisp to explore its diagnostic capability. Figure 6 shows the organization of the experimental system. To best evaluate the approach two examples are shown here in which knowledge of each type was used separately.

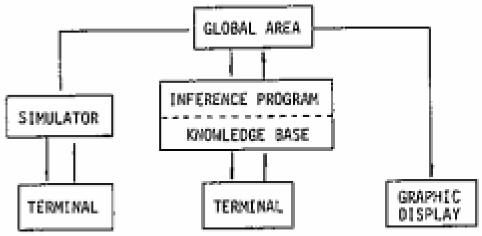


Fig. 6 Organization of experimental system

The first example was a diagnosis based on event-oriented knowledge. The assumed causes were LPCP trouble (primary event) and HPCP startup failure (succeeding event). The results are shown in Fig. 7. Both of the assumed causes were identified correctly together with the appropriate corrective actions to guide the reactor to cold shutdown.

The second example was a diagnosis based on function-oriented knowledge. The assumed cause was the failure of water level sensor (S-W-A) and the anomaly was first detected by the flow sensor (S-TDFLOW-A) at the TD/RFP output, by which time the anomaly had already propagated through many components. The results are shown in Fig. 8. The diagnostic process could not focus on the cause uniquely by using the observed data alone. It , therefore, generated a test indicating the switching of the selector WLS2 from the S-W-A to the S-W-B. The test was successful and the cause was identified correctly as the S-W-A. In this case the switching itself was the right corrective action.

Although the models used here were simple ones, the method worked as expected and its capability for process computer applications was demonstrated.

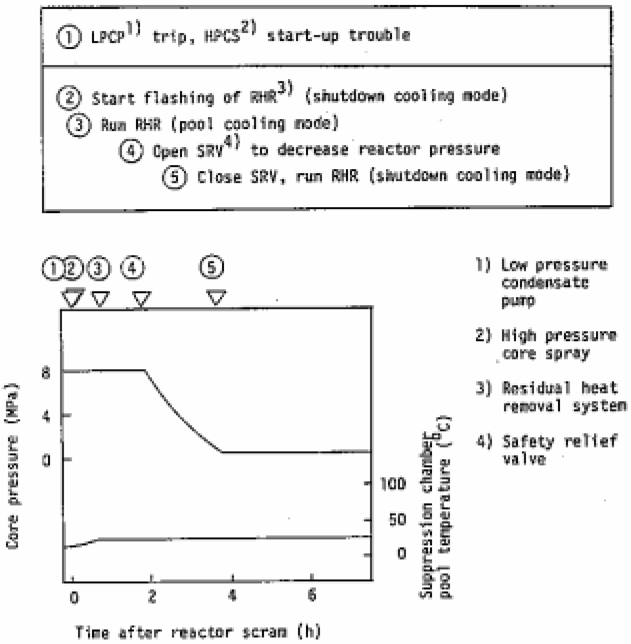


Fig. 7 Results of diagnosis for the failure of LPCP and HPCP (event-oriented knowledge used)

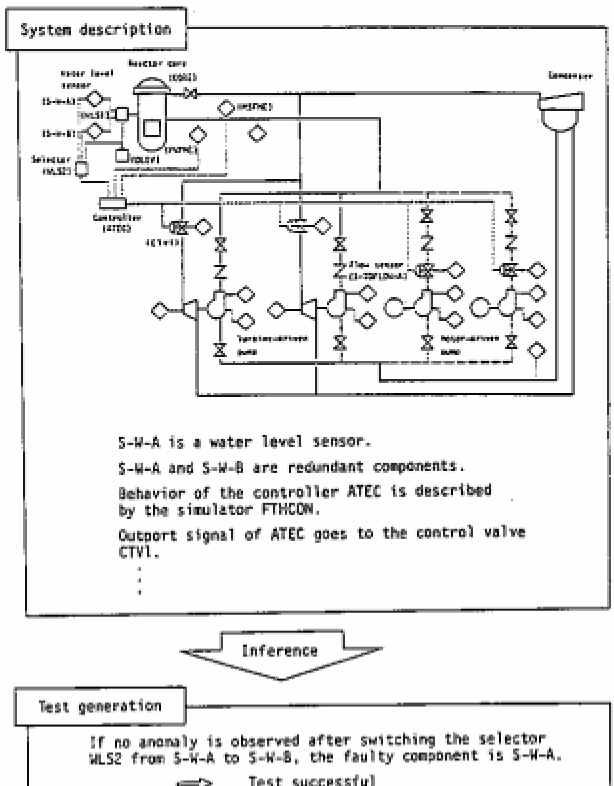


Fig. 8 Results of diagnosis for the failure of S-W-A (function-oriented knowledge used)

6 Conclusion

A knowledge based system pertaining to plant diagnosis was presented. The system used both event-oriented and function-oriented knowledge. These two types of knowledge were represented by mutually nested frames and production rules. This implementation scheme significantly reduced the search process.

Forward chaining mechanism was used to provide both diagnosis and guidance based on event-oriented knowledge, whereas the resolution mechanism was used to make a diagnosis based on function-oriented knowledge. The feedback nature of the plant required the best search mechanism that uses histories in the resolution procedure.

Simulation applications to a BWR plant diagnosis has demonstrated the diagnostic capability of this system.

References

1) Bennett, J. S. and Hollander C. R., DART: An Expert System for Computer Fault Diagnosis, Proc. of IJCAI, 843 (1981)

2) Davis R., Shrobe H., Hamscher W., Wieckert K., Shirly M. and Polit S., Diagnosis Based on Description of Structure and Function, Proc. of AAAI, 137 (1982)

3) Genesereth M. R., Diagnosis Using Hierarchical Design Models, Proc. of AAAI, 278 (1982)

4) McDermott D. and Brooks R., ARBY: Diagnosis with Shallow Causal Models, Proc. of AAAI, 370 (1982)

5) Shubin H. and Ulrich J. W., IDT: An Intelligent Diagnostic Tool, Proc. of AAAI, 290 (1982)

6) Wells A. H. and Underwood W. E., Knowledge Structures for a Nuclear Power plant Consultant, Trans. Am. Nucl. Soc., 41, 41 (1982)

7) Yoshida k., Kiguchi T., Motoda H. and Kobayashi S., Knowledge-Based approach to Plant Diagnosis, Trans. Am. Nucl. Soc., 44, 174 (1983)

8) Yamada N. and Motoda H., A Diagnosis Method of Dynamic System using the Knowledge on System Description, Proc. of IJCAI, 225 (1983)