A Plant Diagnosis Method Based on the Knowledge of System Description

NAOYUKI YAMADA* and HIROSHI MOTODA*

A method is proposed for automatic diagnosis of a dynamic system based on a knowledge engineering technique. Basically, the method uses only knowledge about a system decription and does not require any knowledge concerning failure causality. Inference is made in each of the four diagnostic steps: i) expectation value computation, ii) suspects computation, iii) suspects discrimination using observable data and iv) suspects discrimination by test generation. One of three inference procedures is used where appropriate in selection: forward chaining, backward chaining and resolution. This method can diagnose in principle all kinds of failures that are logically diagnosable if the system description is appropriate. The capability of the method is demonstrated by an application example to a nuclear reactor feed-water system.

1. Introduction

A compelx power plant 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.

A well-established approach for identifying anomaly causes in a dynamically changing system is to use the pre-analyzed scenario of event propagation. A typical example is the Disturbance Analysis System (DAS) based on the Cause-Consequence Tree (CCT) [1]. DAS has useful pre-alarming and diagnosis capabilities that can cover a variety of foreseen circumstances. Its efficiency comes from its use of a set of explicitly enumerated faults, but biulding a CCT that covers almost all possible faults is a complicated and difficult task.

Several groups [2][3][4][5][6][7] have proposed a new approach which utilizes the knowledge engineering technique. A main feature of this approach is its capability of tracing a logical chain of events, which is constructed by a set of rules incorporated in a knowledge base. Application to plant diagnosis offers the following advantages:

1) Effective integration of information

Complex phenomena that propagate through various plant components can be represented in terms of logical event chains.

2) Easy system management

The diagnostic ability can be easily improved by modification of the knowledge base.

3) Excellent man-machine interface

It can explain its line of reasoning in reaching a conclusion.

4) Use of heuristic knowledge

Know-how that is heuristic and has been accumulated through the experiences of experts can be utilized in problem solving.

The reasoning process, however, is one of searching the state of classifications, based on knowledge of the causal association between observable symptoms and possible causes. In this regard, the approach requires an explicit expression of causality: cause and result relationship. What is not expressed in this knowledge is out of the scope of the diagnosis.

A second new approach that has been put forward is to use knowledge about a system description, i. e. intended structure and expected behavior. Application to computer hardware diagnosis shows that the algorithm can work directly from information about a system description without requiring causality relationship [8][9][10].

These two approaches are characterized in that the former uses kowledge of anomalous situations, whereas the latter uses knowledge of normal situations. The former is more direct and hence more efficient but, as mentioned above, all anomalous situations must be covered for the diagnostic capability to be complete. 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 extend the second approach which, so far, has been limited to computer hardware diagnosis, to a diagnostic problem in a dynamic system that has feedback loops. The algorithm uses a general inference procedure to i) compute expectation value, ii) compute suspects, iii) discriminate between the suspects using observable data and iv) further discriminate between them by generating tests. The advantages listed above for the first approach also apply to this approach.

Section 2 defines a problem to be solved, and section 3 presents some examples of system description. Section 4 describes the diagnostic method. Section 5 sum-

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

marises application results for a power generating plant.

2. Diagnosis Problem

The problem to be solved can be stated simply as follows: [Given a symptom indicating an anomaly through an observable signal from some detector at time, identify the faulty component(s) that caused the symptom.]

The main differences between this diagnosis and that of computer hardware are:

1) The system is dynamic, i. e. the observable signals are time dependent.
2) The system forms feedback loops, i. e. input from some components is affected by its own output as well as that of other components, with some time lag.
3) Many of the important signals are observable.
4) Some of the important components are redundant.

The first two make the problem more difficult than that for computer hardware diagnosis. The time dependency of the system must be appropriately modeled. Feedback nature requires elaborate control for inference to work correctly. The last two make the problem easier. However, there are many variables that are not sensored and prediction by model is required for these variables. Redundancy comes from a safety requirement. This information can be used to guide inference control.

3. System Description

The method described here requires data giving a full description of the plant to be analyzed. The degree of sophistication of the behavior description is determined by its capability of distinguishing a normal state from an abnormal one. The knowledge representation used for system description is based on MRS [11]. Its syntax is the same as that of predicate calculus. In the following description, "if", "and" and "not" are logical symbols and some predicate symbols, function symbols and relation symbols are introduced to describe a dynamic system.

3.1 Structure Description

The system structure is specified by describing components, interconnetions and states.

i) Component description

Each component is designated by an atomic name and its type is specified by using a function symbol "type". The following assertion example declares that components A, B, and C are a sensor, selector and pump, respectively.

```
(type A Sensor)
(type B Selector)
(type C Pump)
```

ii) Connectivity description

Each component has zero or more input and output ports. These ports are designated using function symbols input and output. The connectivity relationship between components is specified by using a relation symbol "conn". The following example declares that the first output of pump C is connected to the second input of sensor A.

```
(conn (output 1 C) (input 2 A))
```

iii) State description

Some components need state information. This information refers to on/off state, observability, redundancy, switching condition, etc. and is represented by introducing appropriate predicate symbols and relation symbols. The following assertions declare that the state of the first input of sensor A is on, the first output of the selector B is observable, pumps C and D are redundant, and the first input of selector B is switchable, where symbols starting with a "$" represent variables.

```
(value (input 1 A) on)
(observable (value (output 1 B) $x))
(redundant C D)
(switchable (value (input 1 B) $x))
```

3.2 Behavior Description

The system behavior is specified by describing the relationship between input(s) and output(s) of each component in terms of rules. Behavior of a component in a dynamic system is usually described by a differential equation, which is made discrete by a set of arithmetic expressions.

i) Dynamics description

The behavior rules relating input(s) to output(s) are denoted as forward rules and those relating output(s) to input(s), backward rules. Simulation of the system behavior requires use of only the forward rules, but inference for diagnosis requires both.

a) Forward rules

Two examples are given. The first describes sensor behavior and the second, controller behavior. OK means that a component is not faulty. A predicate symbol "true" is introduced to represent time varying data. The statement (true A B) means that A is true in situation B.

The first rule states that the sensor is a two-input, one-output device: one input being an on/off switch and the other sensing a quantity *mi*, and that if the sensor is on and functioning normally, the output *mo* is *mi*/*mr*, where *mr* is a scale factor. This is true for every *t*.

```
(if (type $x Sensor)
  (if (and (OK $x)
           (value (input 1 $x) on)
           (true (value (input 2 $x) $mi) $t)
           (value (rated $x) $mr)
           (= $mo (/ $mi $mr)))
    (true (value (output 1 $x) $mo) $t)))
```

The second rule is more complicated. The output of the controller *mo* at time *t* is computed by a function f that requires 6 variables, one of which is the output *mi*

itself at the previous time step *s*.

```
(if (type $x Controller)
  (if (and (OK $x)
           (true (value (input 2 $x) $ll) $s)
           (true (value (input 3 $x) $wfl) $s)
           (true (value (input 4 $x) $wml) $s)
           (true (value (output 1 $x) $ml) $s)
           (value (input 1 $x) $ld)
           (= $t (+ $s 1))
           (true (value (input 2 $x) $lo) $t)
           (= $mo (f $ml $ll $lo $ld $wfl $wml)))
      (true (value (output 1 $x) $mo) $t)))
```

b) Backward rules

The following rule corresponds to the first example of the forward rule given above.

```
(if (type $x Sensor)
(if (and (OK $x)
         (true (value (output 1 $x) $mo) $t)
         (value (input 1 $x) on)
         (value (rated $x) $mr)
         (= $mi (* $mo $mr)))
    (true (value (input 2 $x) $mi) $t)))
```

ii) Connectivity description

A set of rules is needed that handles the connectivity relationship. These rules state that if two ports are connected, they always have the same value.

a) Forward connectivity rule

```
(if (and (conn $x $y)
         (true (value $x $x) $t))
    (true (value $y $z) $t))
```

b) Backward connectivity rule

```
(if (and (conn $x $y)
         (true (value $y $z) $t))
    (true (value $x $z) $t))
```

4. Diagnostic Method

An anomaly is detected by observing that some sensor output gose beyond its allowable range. The real cause may lie in a component that is not directly related to the sensor where the anomaly is first detected. By the time of detection, the anomaly may have propagated through various components and affected many sensor outputs although they were still within their allowable ranges. The diagnosis consists of the four steps described in detail in the following subsections. In each step, diagnosis is realized by one or a combination of the three inference procedures: forward chaining, backward chaining and resolution. The resolution procedure is necessary to fully mechanize the diagnosis made by the mixed use of the forward and backward behavior rules. Linear input strategy is adopted because of its ease to implement a control mechanism.

4.1 Computation of Expectation Value

Diagnosis starts when a symptom does not match what is expected. It is, therefore, necessary to estimate the expected value of the sensor where an anomaly is detectd. To do this, plant dynamics has to be simulated starting from some initial state. It is not necessary to return to a state in which all components were normal because the input and output relationship of a normal component is consistent regardless of the value of its input(s). It is sufficient, in a dynamic system having feedback loops, to go back at least to the time t-Δ and use a set of consistent observable data, where t is the time of anomaly detection and Δ is the maximum difference in time between each observable output in solving the dynamic system having feedback loops by discrete time periods. In other words, Δ is the minimum time needed for an erroneous signal to propagate through all the components at least once before it is detected by some sensor. Inference is made in two steps.

i) Inference to obtain unobservable data at time t-Δ using the observable data at time t-Δ

Forward chaining is applied starting from the observable data by using the structural knowledge backward behavior rules, and backward connectivity rule assuming all components are normal. Inference control is a) not to conclude the once instantiated data by previous inference and b) not to go back further than the time t-Δ. The first one is necessary to get a set of consistent data if there are more observable data than minimum necessary to start simulation.

ii) Inference to obtain the expectation value of the anomaly detecting sensor at time t using the unobservable data obtained in step i)

Forward chaining is applied starting from the estimated unobservable data using structure data, forward behavior rules and forward connectivity rule. Inference control is to stop when the expectation value is obtained.

4.2 Computation of Suspects

Using the fact that the symptom is not the expected observation, all components that can logically be responsible for it are picked as supects. Resolution is applied starting from the expectation violation at time t until reaching the estimated unobservable data at t-Δ using the structure data, forward behavior rules, and forward connectivity rule. All rules are converted to conjunctive normal from. The control mechanism employed to pick the suspects is to designate in advance the statements for which further inference is not required. These statements are an OK statement for each component.

4.3 Discrimination of the Suspects Using Observable Data

It is possible to discriminate between the suspects obtained in the previous step by checking the consistency of the available data. Here, consistency means that the observed output(s) can be expected from the observed input(s) using knowledge about a system description. The knowledge required in this step is the structure data, forward and backward behavior rules, and for-

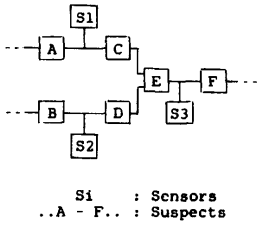


Fig. 1 An example of suspect discrimination by observable data.

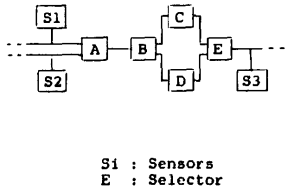


Fig. 2 An example of suspect discrimination by test generation.

ward and backward connectivity rules. Inference is made in two steps.

i) Identification of which observable data to use

This step finds a set of observable data that are required to identify the anomaly of one more components. Symbolic simulation is performed by resolution starting from the forward behavior rule of any one of the suspect candidates obtained in 4.2. Backward chaining can be applied using heuristic knowledge to select a component from which to start inference for greater efficiency. Because the feedback nature of the system necessitates mixed use of both forward and backward rules, inference should be controlled to avoid getting into an infinite loop. This can be achieved by preparing a) repetition checks to rule out an infinite loop in which the process of proving a statement involved its own proof and b) tautology checks to rule out an infinite loop between input and output within a component. In addition to these controls, the OK and Observable Statements are designated as statements for which further inference is not required.

In Fig. 1, for example, starting the resolution from the rule of component E, the outputs of sensors S1, S2 and S3 are picked up as the data that can be used to identify an anomaly in the components C, D and E. The sensors can also be the suspects in this case. This process is repeated until none of the suspects can be exonerated by the consistency check with the observed data. In the Appendix, the detailed inference process of this simple example is described in part along with the system description used.

ii) Evaluation of observable data

Numeric simulation is performed for each set to check whether observable data are consistent with each other. If they are inconsistent, at least one of the components for that set is faulty.

4.4 Discrimination of the Suspects by Test Generation

It is possible to further discriminate between suspects by placing single fault and non-intermittency assumptions for each set that has been selected in 4.3 if a meaningful test can be generated and if it is successful. Use of the redundant component or valve open/close can be realized in plant diagnosis. In order to infer the test form logically, knowledge about the ability to modify the system or components is needed. This knowledge, supplied as state description in 3.1 is used in selecting the component to start inference. Inference is made in two steps.

i) Generation of tests

The knowledge required and the inference procedure used in this step are the same as in 4.3. Fig. 2 is an example of a set of suspects for which a test is possible. Inconsistency is observed among the data of S1, S2 and S3. The components C and D are redundant and the component E is a selector that determines whether to use C or D. Assume that component C is selected. The suspects at this stage are A, B, C and E. If there is still inconsistency among the data when the selector E is switched from C to D, component C is exonerated from among the suspects, otherwise, component C is faulty.

The test generation in this example, therefore, is to derive a statement of the form:

```
(if (and (OK S1) (OK S2) (OK S3)
         (OK A)(OK B) (OK D) (OK E)
         (true (value (output 1 S1) $s1) $t)
         (true (value (output 1 S2) $s2) $t)
         (value (setting E) D)
         (= $a (fa $s1 $s2))
         (= $b (fb $a))
         (= $d (fd $b))
         (= $e $d)
         (= $s3 (/ $e $er)))
    (true (value (output 1 S3) $s3) $t)),
```

where no time lag is assumed between input(s) and output(s) for simplicity. This statement says that if sensors S1, S2 and S3 and components A, B, D and E are working, and if selector E is switched to D, the output of sensor S3 is expected to be a value calculated by the set of functions indicated in the rule using the outputs of sensors S1 and S2. The resolution starts from the redundant compoment D which is selected by the knowledge that C and D are redundant, selector E is switchable and now C is selected. It is continued until the above test form is obtained. A similar test can be generated for valve control.

ii) Evaluation of tests

Numeric simulation is performed for the derived test, and the simulated results are evaluated against the observable data.

Single fault assumption is used for each of the discriminated sets of the suspects at this final stage to simplify the diagnosis. In a situation where this assumption is not accepted, it becomes extremely difficult to

discriminate between suspects unless some heuristic knowledge is employed. Some of the cases cannot be logically diagnosed without such knowledge.

5. Application to Diagnosis in a Nuclear Reactor Power Plant

The above method is applied to a simplified model of the feed-water system in a boiling water reactor shown in Fig. 3. The system is composed of 29 components. Steam leaving the core is condensed to water and returned to the core by the feed-water pump. A small fraction of the steam is used to drive a turbine-driven feed-water pump. The power level is controlled by the recirculation flow rate. The water level is kept constant by the controller which uses signals from a water level sensor, feed-water flow meter, and main steam flow meter. The condenser is assumed to serve as a source and sink for water and steam. The system dynamics are, thus, determined by those of core, controller and pumps. The water level sensor 1 (S3) and 2 (S4), and the turbine-driven (J) and motor-driven (L) pumps are redundant components. Under normal operating conditions, S3 and J are used. When S3 is used, S4 is not observable.

Simulation was performed on a DEC 2060 using TSS. All the functions were written in MACLISP. The Δ defined in 4.1 to satisfy the propagation requirement was set at 0.2 sec., which is twice the time step of solving the differential equations in this application.

The following hypothetical situation is assumed. Component S3 failed. The anomaly was first detected by the alarm signal of S9 at the feed-water pump outlet during a load following operation in which the plant was not in a steady state. By the time of detection, the anomaly had already propagated through various components and affected many sensor outputs although they were still within their allowable ranges, except for S9.

After computing the expectation value of S9 using the past observable data (step 1), the suspects computation was started to return the following components (step 2):

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, Q, T, S1, S2, S3, S8, S9

In this example, almost all components in the system could be suspects. Use of the observable data narrowed the suspects to the following seven (step 3):

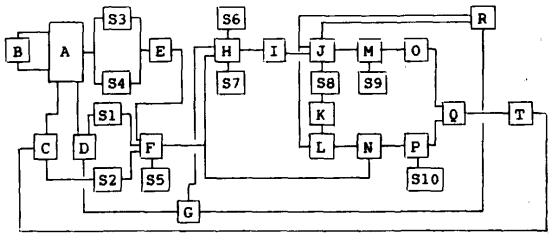
A, B, C, D, S1, S2, S3

A test was then generated with the knowledge that S3 and S4 were redundant and S3 was in use:

[Switch selector E to S4 from S3. If the data S1, S2 and S4 are consistent, S3 is faulty. Otherwise, the fault must lie in either A, B, C, D, S1, S2 or S3.]

In this case, the data were consistent by the assumption, and thus, the test was successful. The faulty component was concluded to be the water level sensor 1.

All of the above diagnostic steps were automated. To improve their efficiency, heuristic knowledge was also employed and used together with knowledge about system description. An example is that it is worthwhile to start resolution from the redundant component in discriminating between the suspects using the observable data.



Core	: A	Condenser	: R
Feed-water controller	: F	Pipe	: C, D, G, I, M, P, Q
Water level selector	: E	Sensor	: S1 - S10
Recirculation pump	: B	Water level sensor 1	: S3
Feed-water pump		Water level sensor 2	: S4
Turbine-driven (main)	: J	Pump outlet flow meter	: S9
Motor-driven (aug)	: L	Interlock	: K
Valve	: H, N, O, T		

Fig. 3 Simplified diagram of a BWR feed-water system.

6. Conclusion

A method to diagnose a dynamic system with feedback structure was proposed. The method requires no fault model or knowledge concerning failure causality. The diagnosis is based mainly on linear input resolution and all the inference steps are automated.

Application to a BWR feed-water system demonstrated its diagnostic capability although the model was much simplified and the assumed anomaly was hypothetical.

Experience with examples recommended use of knowledge about system description in combination with heuristic knowledge for better efficiency. Employment of frame type representation of the system coupled with criteria inference capability would further improve the efficiency. It is important to distinguish logical inference from simulation.

Acknowledgements

This paper is based on work performed at Teknowledge Inc. The authors are grateful to Profs. M. R. Genesereth and E. H. Shortliffe of Stanford University for consultation. Thanks are also extended to Drs. T. Kiguchi and S. Kobayashi for critical discussions, and to Drs. S. Yamada and K. Taniguchi for support of this work.

References

1. MEIJER, C. H. and FROGNER B. On-line Power Plant Alarm and Disturbance Analysis System, EPRI, NP-1379 (Apr. 1980).
2. WELLS, A. H. and UNDERWOOD, W. E. Knowledge Structures for a Nuclear Power Plant Consultant, *Trans. Am. Nucl. Soc.* **41**, 41 (1982).
3. NELSON, W. R. REACTOR: An Expert System for Diagnosis and Treatment of Nuclear Reactor Accidents, *Proc. of the National Conference on Artificial Intelligence*, 296 (1982).
4. UNDERWOOD, W. E. A CSA Model-Based Nuclear Power Plant Consultant, *Proc. of the National Conference on Artificial*

```
FACTS

Connectivity
  (conn (output 1 A) (input 2 S1))
  (conn (output 1 A) (input 1 C))
  (conn (output 1 C) (input 1 E))
  (conn (output 1 B) (input 2 S2))
  (conn (output 1 B) (input 1 D))
  (conn (output 1 D) (input 2 E))
  (conn (output 1 E) (input 2 S3)))

Status
  (value (rated Si) ri)   (i = 1 - 3)
  (value (input 1 Si) on)   (i = 1 - 3)


RULES (CNF form)

F-rule
     (true (value (output 1 Si) $mo) $t))
   V (ok Si)
   V (value (rated Si) $mr)
   V (value (input 1 Si) on)
   V (true (value (input 2 Si) $mi) $t)
   V (= $mo (/ $mi $mr))                    (i = 1 - 3)

     (true (value (output 1 C) $mo) $t))
   V (ok C)
   V (true (value (input 1 C) $in1) $t)
   V (true (value (input 1 C) $in2) (- $t 1))
   V (= $mo (funC $in1 $in2))

     (true (value (output 1 D) $mo) $t))
   V (ok D)
   V (true (value (input 1 D) $in) $t)
   V (= $mo (funD $in))

     (true (value (output 1 E) $mo) $t))
   V (ok E)
   V (true (value (input 1 E) $in1) $t)
   V (true (value (input 2 E) $in2) $t)
   V (= $mo (funE $in1 $in2))

B-rule
     (true (value (input 1 Si) $m) $t))
   V (ok Si)
   V (value (rated Si) $mr)
   V (value (input 1 Si) on)
   V (true (value (output 1 Si) $mo) $t)
   V (= $m (* $mr $mo))                     (i = 1 - 3)

F-connectivity
     (true (value $y $z) $t)
   V (conn $x $y)
   V (true (value $x $z) $t)

B-connectivity
     (true (value $x $z) $t)
   V (conn $x $y)
   V (true (value $y $z) $t)


TERMINATION CONDITION

  (ok $x)
  (= $x $y)
  (observable (true (value (output 1 Si) $x) $t))   (i = 1 - 3)
```

Fig. A1 Facts, rules and termination conditions required in suspect discrimination of Fig. 1.

```
RESOLUTION PROCESS

     (true (value (output 1 E) $mo) 7)
   V (ok E)
   V (true (value (input 1 E) $in1) 7)
   V (true (value (input 2 E) $in2) 7)
   V (= $mo (funE $in1 $in2))

     (conn (output 1 E) $y))                    (by F-connectivity)
   V (true (value $y $mo) 7)
   V (ok E)
   V (true (value (input 1 E) $in1) 7)
   V (true (value (input 2 E) $in2) 7)
   V (= $mo (funE $in1 $in2))

     (true (value (input 2 S3) $mo) 7)          (by fact)
   V (ok E)
   V (true (value (input 1 E) $in1) 7)
   V (true (value (input 2 E) $in2) 7)
   V (= $mo (funE $in1 $in2))

     (true (value (output 1 S3) $001) 7)        (by F-rule)
   V (ok S3)
   V (value (rated S3) $mr)
   V (value (input 1 S3) on)
   V (= $001 (/ $mo $mr))
   V (ok E)
   V (true (value (input 1 E) $in1) 7)
   V (true (value (input 2 E) $in2) 7)
   V (= $mo (funE $in1 $in2))

     (true (value (input 1 E) $in1) 7)          (by fact)
   V (true (value (input 2 E) $in2) 7)
   V (= $mo (funE $in1 $in2))
   V (true (value (output 1 S3) $001) 7)
   V (ok S3)
   V (= $001 (/ $mo r3))
   V (ok E)

     (conn $002 (input 1 E))                    (by F-connectivity)
   V (true (value $002 $in1) 7)
   V (true (value (input 2 E) $in2) 7)
   V (true (value (output 1 S3) $001) 7)
   V (ok S3)
   V (= $001 (/ $mo r3))
   V (ok E)
   V (= $mo (funE $in1 $in2))

     (true (value (output 1 C) $in1) 7)         (by fact)
   V (true (value (input 2 E) $in2) 7)
   V (true (value (output 1 S3) $001) 7)
   V (ok S3)
   V (= $001 (/ $mo r3))
   V (ok E)
   V (= $mo (funE $in1 $in1))

     . . . . .


     (true (value (output 1 S3) $001) 7)
   V (true (value (output 1 S1) $004) 7)
   V (true (value (output 1 S1) $005) 6)
   V (true (value (output 1 S2) $007) 7)
   V (= $002 (* r1 $004))
   V (= $003 (* r1 $005))
   V (= $006 (* r2 $007))
   V (= $in2 (funD $006))
   V (= $in1 (funC $002 $003))
   V (= $mo (funE $in1 $in2))
   V ($001 (/ $mo r3))
   V (ok S1)
   V (ok S2)
   V (ok S3)
   V (ok C)
   V (ok D)
   V (ok E)
```

Fig. A2 Resolution process for suspect discrimination of Fig. 1.

Intelligence, 302 (1982).

5. CHANDRASEKARAN, B., SHARMA, D. D. and MILLER, D. W. The Application of Knowledge-Based Systems to Reactor Operations, *Trans. Am. Nucl. Soc.* **43**, 241 (1982).

6. GUILLON, G., PARCY, J. and BERLIN, C. Application de L'intelligence Artificielle a la Detection et an Diagnostic des Defauts du Coeur dans les Reacteurs a Neutrons Rapides, International Symposium on Nuclear Power Plant Control and Instrumentation, IAEA-SM-265/54 (Oct. 1982).

7. YOSHIDA, K., KIGUCHI, T., MOTODA, H. and KOBAYASHI, S. Knowledge-Based Approach to Plant Diagnosis, to appear in *Trans. Am. Nucl. Soc.* (Jun. 1983).

8. GENESERETH, M. R. The Use of Hierarchical Design Models in the Automated Diagnosis of Computer Systems, HPP-81-20, Stanford University Heuristic Programming Project (Dec. 1981).

9. GENESERETH, M. R. Diagnosis Using Hierarchical Design Models, *Proc. of the National Conference on Artifical Intelligence*, 278 (1982).

10. DAVIS, R., SHROBE, H., HAMSCHER, W., WIECKERT, K., SHIRLEY, M. and POLIT, S. Diagnosis Based on Description of Structure and Function, *Proc. of the National Conference on Artificial Intelligence*, 137 (1982).

11. GENESERETH, M. R., GREINER, R. and SMITH, D. E. MRS Manual, HPP-81-6, Stanford University Heuristic Programming Project (Dec. 1981).

Appendix

The resolution process of Fig. 1 is shown in detail together with facts, rules and termination conditions for illustrative purpose. The last statement in Fig. A2 relates the observable outputs of the sensors S1 and S2 to the observable output of the sensor S3 when the components S1, S2, S3, C, D and E are functioning normally. Functions are not evaluated in this resolution process. This process is called symbolic simulation in this paper. The resolution process in other steps is in principle the same.

(Received July 13, 1983; revised February 6, 1984)