A NEW SHELL FOR THE DEVELOPMENT OF ALARM PATTERN RECOGNITION EXPERT SYSTEMS

By

Juan Arellano, Yalu Galicia, Edgar C. Ramirez

Instituto de Investigaciones Electricas
Apartado Postal 475
Cuernavaca, Morelos
62000 Mexico

ABSTRACT

This paper presents the first version of GENESIS, an expert system shell suitable for the development of alarm pattern recognition expert systems (APRES). GENESIS includes a series of algorithms and procedures especially designed for a rapid and systematic construction of APRES. The inputs required by GENESIS are the fault trees of the system under analysis, the probability of occurrence of each fault in the trees, and the set of symptoms (alarms and measurements) associated to the occurrence of each individual fault. With this information, GENESIS generates a set of production rules which relates faults and symptoms. The shell uses these rules and the probability of occurence of each of the faults in order to generate optimal alarm pattern recognition strategies (algorithm of the inference engine). A strategy helps the operator to recognize which alarm pattern is occurring without having to search the entire set of patterns. All the alarm pattern recognition strategies are generated off-line, as a consequence, the response of the system will be very fast. This feature makes GENESIS a powerful tool for the development of real-time APRES.

INTRODUCTION

The successful development of expert systems for such problems as medical diagnosis, interpretation of oil well data and configuration of computer systems has fostered wide interest in a broad range of applications. In particular, Moore [1] has described several areas where expert systems technology might be utilized for improved power plant operation. One of the proposed areas is the analysis of significant events, such as plant upsets. In a modern distributed process control system there may be hundreds of alarms and measurements presented to an operator. Unfortunately, most of these alarms and measurements are related to single plant parameters while the plant is operated according to mass and energy balances that typically include several variables (this implies that fault conditions will be indicated by patterns of alarms and other indicators). Thus, if in the first minute of a major upset, the operator is faced with over a hundred alarms, how would he or she know which alarms should be considered first, which would be the most critical and which would be the basic causes of the problem ?. In this case an expert system would interpret alarms, identifying underlying process problems, distinguishing causes from effects, and would advise the operator accordingly.

In this paper GENESIS, a new shell for the development of alarm pattern recognition expert systems (APRES) is presented. GENESIS provides a formal and consistent environment for the development of APRES. It includes a series of procedures and algorithms especially designed for a rapid and systematic construction of alarm pattern recognition expert systems.

The inputs required by GENESIS are the fault trees (logic models which contain information on the different combinations of faults that can produce undesired events) [2] of the system under analysis, the probability of occurrence of each fault in the trees, and the set of symptoms (alarms and measurements) associated to the occurrence of each individual fault. With this information, GENESIS obtains the minimal combinations of faults (minimal cut sets [2]) such that if any of these combinations occur, the undesired event occurs. These minimal cut sets are used by GENESIS, along with the set of symptoms, to generate a set of production rules which relates faults and symptoms. The shell uses these rules and the probability of occurence of each of the faults in order to generate optimal alarm pattern recognition strategies (algorithm of the inference engine). A

strategy helps the operator to recognize which alarm pattern is occurring without having to search the entire set of patterns. On the contrary, the inference algorithm ensures that, in general, only a few patterns will have to be checked before an effective diagnosis can be made to determine the causes of the undesired event.

The main features of GENESIS are:

1. Knowledge acquisition from the human expert domain is performed in a very rapid and systematic way. This feature reduces the time required to develop a particular application and expedites maintenance of the knowledge base.

2. It incorporates not only empirical observations of human experts, but also knowledge about the structure and function of the device under study.

3. Alarm pattern recognition strategies are generated off-line; this approach eliminates the need for a substantial part of the on-line search required to solve classification problems using traditional expert systems development techniques. This feature makes this shell a powerful tool for the development of real-time APRES.

4. It was developed on a PC using the "C" programming language.

5. It enables users to easily implement APRES.

The organization of this paper is the following: Section 2 presents the architecture of GENESIS and describes the main modules contained in it; section 3 illustrates the use of the shell by means of an example taken from literature; finally, the most relevant conclusions are discussed in section 4.

ARCHITECTURE OF GENESIS

Figure 1 shows the components that conform the shell. A brief description of both each module of GENESIS and the steps that a user must follow to develop an APRES using the approach proposed here is presented next.

Knowledge Acquisition

Knowledge for a particular APRES, developed using

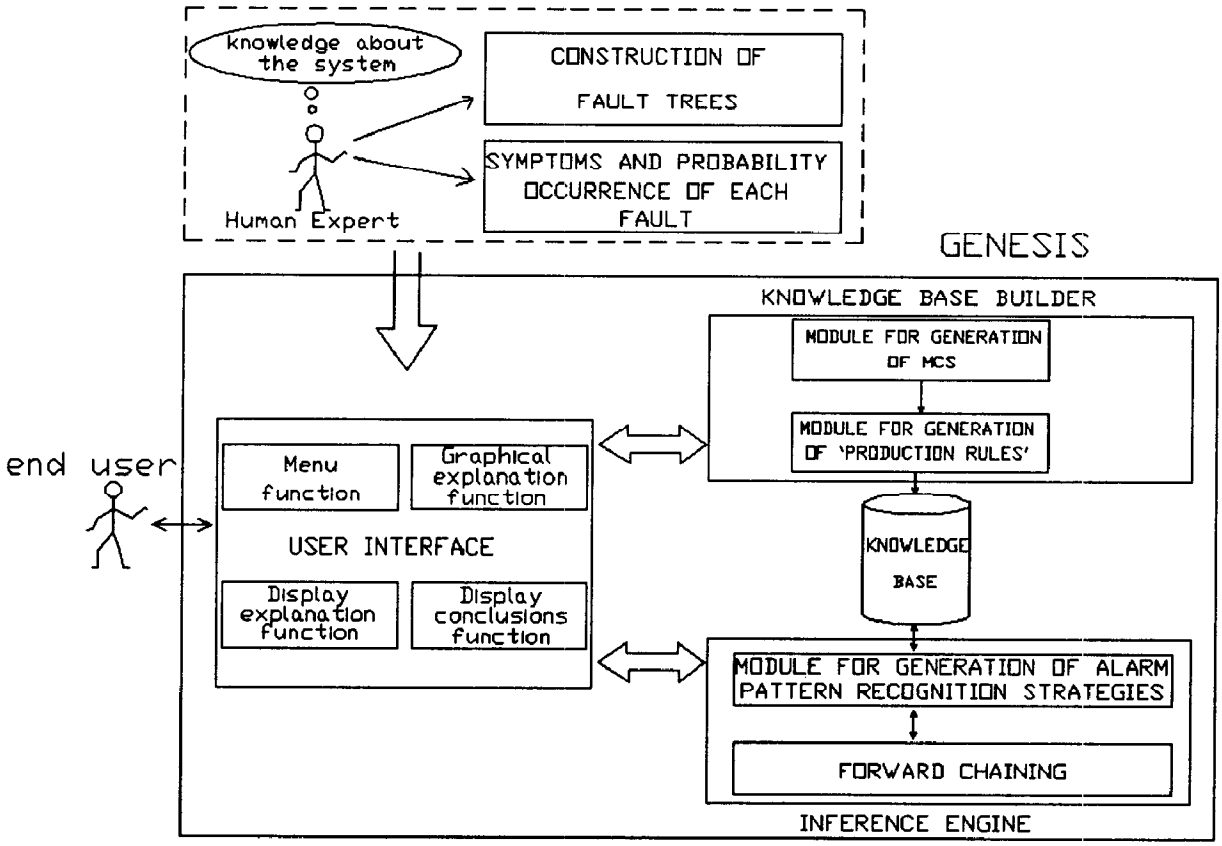

Figure 1. Architecture of GENESIS.

GENESIS, is acquired in a systematic way by the construction (performed by human experts) of logic models called fault trees. A fault tree is a logic model that represents the different ways in which faults (component faults, human errors, etc.) are combined to produce an undesired event. The relationship between the different causes is established by means of logic gates (mainly AND, OR and NOT gates). Construction of fault trees can be accomplished by following the rules proposed in reference 2. For this purpose GENESIS does not provide a program (dotted line module); it only provides guidelines and procedures.

Knowledge Base Builder

Once the fault trees for a given system are developed, they will contain information on the different combinations of primary events that can produce the undesired event. This information can be retrieved from these trees by obtaining their Minimal Cut Sets (MCS). The minimal cut sets of a fault tree are the minimal combinations of faults (primary events) such that if any of these combinations occur, the top undesired event occurs. The order of each MCS is determined by the number of faults contained in it.

Once the MCS have been obtained for each of the fault trees, we will have an equivalent tree for each top udesired event. These equivalent trees are largely reduced compared with the original trees. In general, any fault tree can be reduced (via the MCS) to an equivalent tree with the structure shown in figure 2. In the reduced tree, the top event can occur due to any of the MCS under the OR gate. Each MCS is represented by an AND gate since, by definition, all the faults in a MCS must occur to cause the undesired event. This simplification process has been computerized and it can be accomplished using any of the existing codes for such a process[2]. GENESIS has a module for generation of MCS; this module includes a "top-down" and a "botton-up" algorithm in order to deal efficiently with a wide range of possible sizes and structures of fault trees.

Once the two-level trees have been obtained, they can be extended to include, under each MCS, the pattern of symptoms (alarms and measurements) associated with each primary event. Each extended tree, called a symptom tree, represents those patterns of alarms and measurements that could occur when the top undesired event occurs.

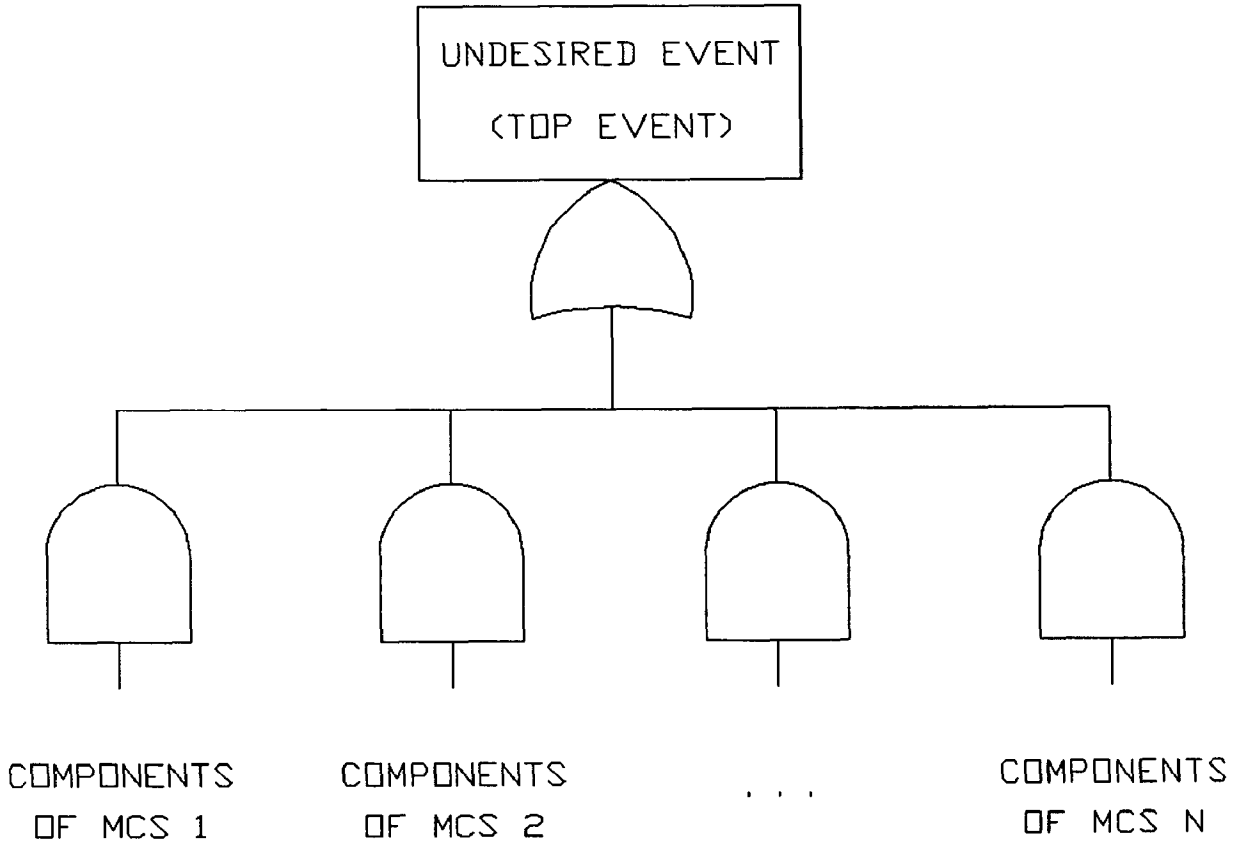

Figure 2. General two-level structure of the reduced tree.

When constructing the extended trees (symptom trees) a second minimization step could be necessary. As an example, let us consider the two-level extended tree shown in figure 3. The minimal cut set "MCS-1" occurs when the primary events (faults) "PE-1" and "PE-2" occur; in turn, when "PE-1" occurs alarms "A1" and "A2" are turned on, and when "PE-2" occurs alarms "A2" and "A3" are turned on. Thus, it is obvious that when MCS-1 occurs the alarm pattern that occurs is "A1", "A2" and "A3" ("A2" and "A2" = "A2").

The two-level tree in terms of faults and symptoms provides us with a set of rules (rule-based scheme) which relates faults and symptoms. Examples of such rules are:

IF T102-HI AND
F201-LO AND
P203-LO

THEN "SEA WATER PUMP HAS FAILED"

and

IF A1 AND
A2 AND
A3

THEN "PE-1" AND "PE-2" HAVE OCCURRED

This set of rules constitutes the knowledge base of an alarm pattern recognition expert system. GENESIS includes a module for automated production rules generation. Again, this task is performed using the two-level extended trees and a Boolean minimization process.

Inference Engine

(a) Generation of alarm pattern recognition strategies

The obtained rules represent the possible combinations of alarms and measurements that could be present when the undesired event occurs. Beside, each rule (pattern) is related to the faults of components that can produce a given symptom. Thus, the knowledge base can be used to recognise this pattern and so aid the operator in fault diagnosis.

It is evident that, by analyzing each one of the rules (alarm patterns), the operator could find the primary events (faults) responsible for the top event occurrence. This procedure would be very inefficient when the number of rules increases. GENESIS includes an algorithm for the generation of optimal alarm pattern recognition strategies. A strategy helps the operator to recognize which alarm pattern is occurring without having to search the entire set of patterns. On the contrary, the algorithm ensures that, in general, only a few patterns

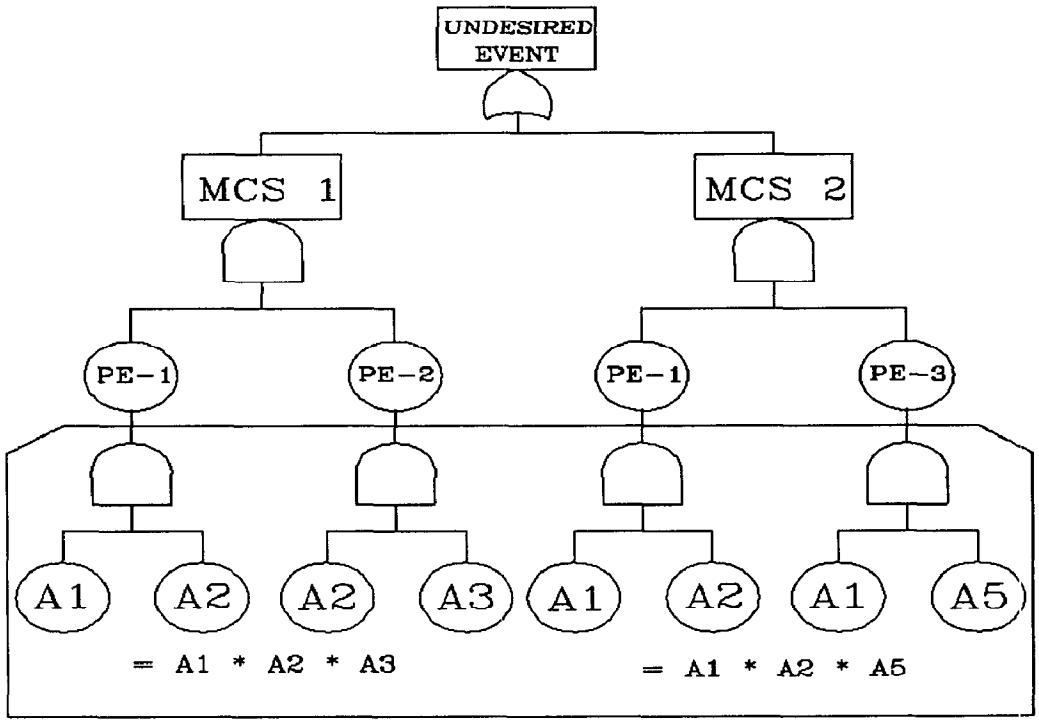

Figure 3. A sample two-level extended tree.

will have to be checked before an effective diagnosis can be made to determine the causes of the undesired event.

The inputs required by the algorithm are the rules and the probability of occurrence of each of the primary events in order to generate an alarm pattern recognition strategy. Once this information has been provided, the algorithm proceeds as follows:

Step 1: Calculate the Fussell-Vesely (F-V) probabilistic importance function for each alarm and measurement in the symptom tree [4]. The probabilistic importance function depends on both the probability of occurrence of the fault (probability of occurrence of the pattern of symptoms associated with the fault) and the number of rules in which an alarm is included.

Step 2: Sort the alarms and measurements by descending value of their F-V importance and combine this list with the table of MCS to generate the alarm pattern recognition strategy following the procedure outlined next.

Given that the undesired event (top event) is present, the general idea in our procedure is that if we check an alarm (or measurement) and this alarm has not turned on, then we know for sure that none of the patterns (rules) to which this alarm belongs has occurred, and, as a consequence, these patterns can be eliminated from the search space. The final search strategy for each undesired event will consist of an initial table and a set of sub-tables. The initial table consists of a group of alarms ordered according to their importance (descending values) accompained by the recommended path that has to be followed in case that alarm has occurred. If the alarm is the only element of a pattern (rule), then by itself can tell us what component(s) has failed. If the alarm is linked to anothers the search strategy will direct us to a sub-table. This sub-table contain some of the alarms that form patterns with the refering alarm in the initial table. Thus the alarm pattern recognition strategy constructed in this way will be very efficient since only the patterns that contains alarms that have occurred need to be considered during the search procedure. A detailed description of the algorithm can be found in reference 6.

(b) Forward chaining mechanism

Once constructed, alarm pattern recognition strategies can be easily tracked by a forward chaining mechanism in order to make an effective diagnosis to determine the causes of the undesired event.

It is important to realize that all the strategies are generated off-line. As a consequence, the response of the system will be very fast. This feature makes GENESIS a powerful tool for the development of real-time APRES.

User Interface

This module provides a friendly environment for communication with the user (human expert or end user). Its main functions are to generate menus, to display diagnostics, and to display graphical and written explanations.

AN APPLICATION EXAMPLE

The Sample System

Let us assume that it is necessary to construct an APRES for the Utility Cooling Water System (UCWS) shown in figure 4 [3]. The system is designed to cool various loads spread over a wide area some of which do not run continuously. A wide range of operating conditions can therefore occur. The Cooling Water System is a closed-loop system that is cooled in turn by a Sea Water System. Automatic flow and temperature control systems are used to stabilize system performances as shown in the figure. The system normally runs unattended, with operator intervention only when faults occur. Operators are therefore not as familiar with the instrument as they would be on a system requiring constant attention.

Construction of the Fault Tree

Figure 5 shows the resultant fault tree for the Cooling Water System. The proposed undesired top event is "Loss of function of the UCWS" (UCWS fails to deliver a particular flow of coolant at a particular temperature). The faults included in figure 5 are chosen for the purpose of illustration and are not claimed to be exhaustive. In this case, the top event occurs when "UCWS flow faults" occurs, or "UCWS temperature faults" occurs. In the same way, the intermediate event "UCWS flow faults" occurs when one or more of its input events occur, and so on. The fault tree is complete when all the terminal branches contain circles (primary events).

Building the Knowledge Base

The fault tree was input to GENESIS in order to obtain the minimal cut sets; these MCS are shown in Table 1. It can be seen that, this very simple tree, contains only minimal cut sets of order one. For example, "Heat Exchanger Fouled" is a minimal cut set of order one since its occurrence leads to the undesired event.

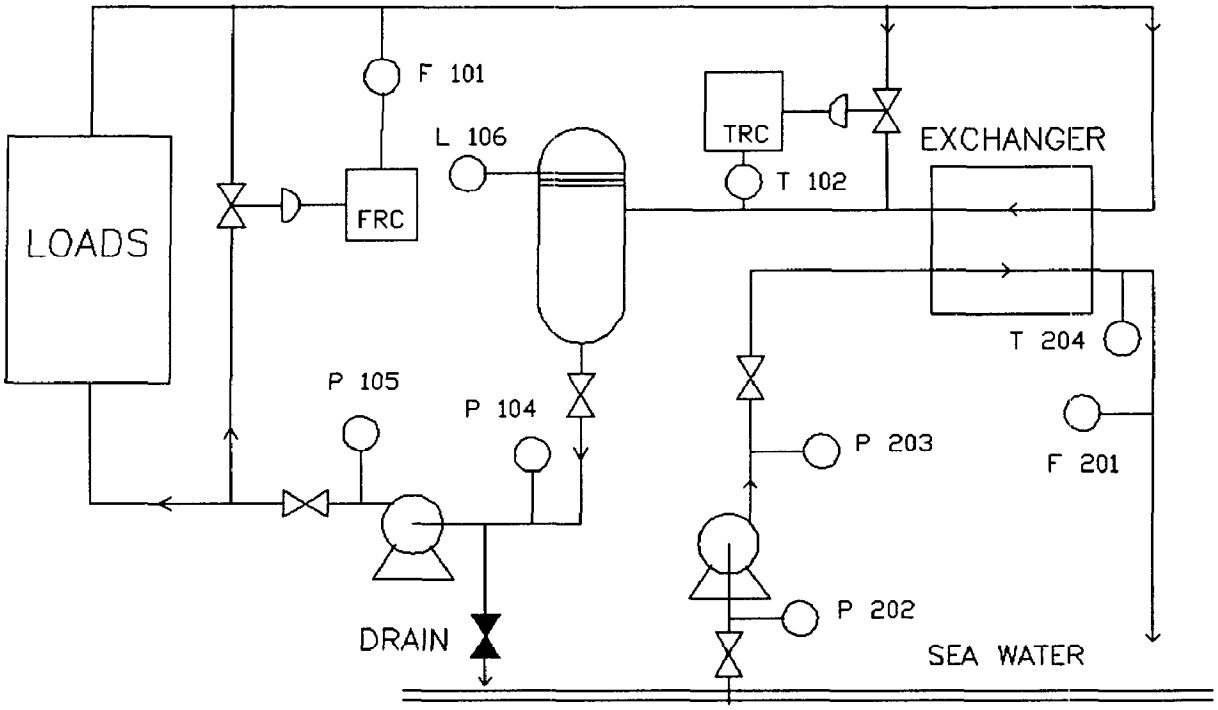

Figure 4. Utility cooling water system.

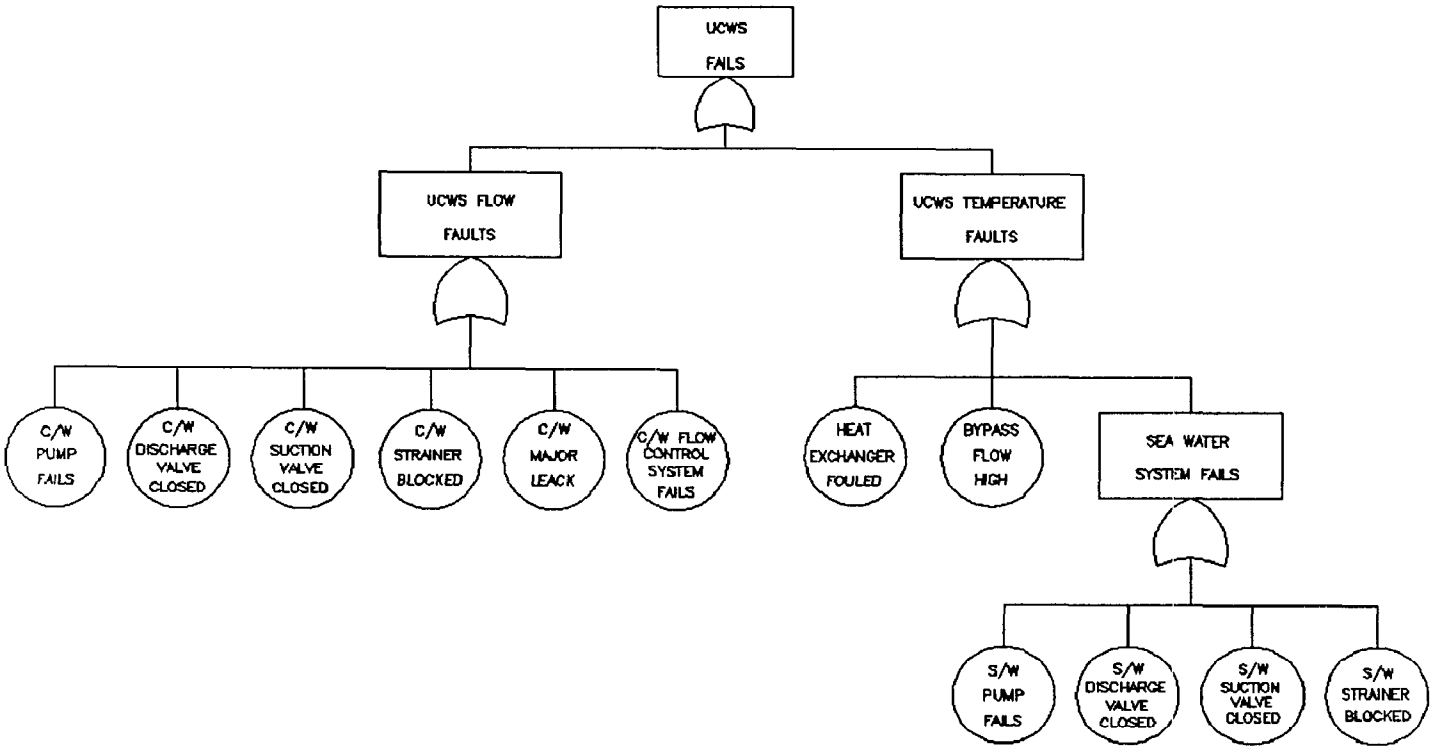

Figure 5. Fault tree for the UCWS.

Once the MCS were calculated, GENESIS used them, along with the pattern of symptoms associated with each primary event (fault) to construct the production rules (knowledge base). Figure 6 shows the extended tree for the UCWS. Due to the original tree contains only single failures under an OR gate (see figure 5) the extended tree is in its MCS representation already. However this is not a common situation; fault trees of complex systems usually have a large and complex structure, thus requiring reduction to their two-level structure.
Figure 6 also shows the set of resultant production rules.

Generation of the alarm pattern recognition strategy

Figure 7 shows the final search strategy generated by GENESIS for the system under analysis as a rule structure diagram (classification scheme). As we can see, if the top event occurs and the alarm pattern being present is "F101-NORMAL", "F201-NORMAL", "T102-HI" and "T204-LO" the search will be directed through "T102-HI", "F201-LO", "F101-NORMAL", "T204-LO" and "F201-NORMAL". Once this pattern has been identified the expert system can infer that the heat exchanger is fouled.

It is very important to realize that GENESIS recognized categories of similar faults in an automated way. The main fault categories (initial table) found by GENESIS are: cooling water temperature faults (T102-HI), cooling water flow faults (F101-LO), cooling water pressure faults (P105-LO), and faults associated with the tank located in the cooling water loop (L-106 FALLING). Each of these categories (alarms) is accompained by the recommended search path that has to be followed in case that alarm has occurred. The final search strategy agrees with the one reported in reference [3].

TABLE 1

MINIMAL CUT SETS FOR THE SAMPLE SYSTEM

MCS	ORDER	DESCRIPTION
1	1	Cooling water pump fails
2	1	Cooling water discharge valve closed
3	1	Cooling water suction valve closed
4	1	Cooling water strainer blocked
5	1	Cooling water major leak
6	1	Cooling water flow control system fails
7	1	Bypass flow high
8	1	Heat exchanger fouled
9	1	Sea water pump fails
10	1	Sea water discharge valve closed
11	1	Sea water suction valve closed
12	1	Sea water strainer blocked

CONCLUSIONS

A new shell (GENESIS) for the development of alarm pattern recognition expert systems has been presented. GENESIS provides a formal and consistent environment for the development of APRES. It includes a series of procedures and algorithms especially designed for a rapid and systematic construction of alarm pattern recognition expert systems.

The inputs required by GENESIS are the fault trees of the system under analysis, the probability of occurrence of each fault in the trees, and the set of symptoms (alarms and measurements) associated to the occurrence of each individual fault. With this information, GENESIS obtains the minimal combinations of faults (minimal cut sets) such that if any of these combinations occur, the undesired event occurs. These minimal cut sets are used by GENESIS, along with the set of symptoms, to generate a set of production rules which relates faults and symptoms. The shell uses these rules and the probability of occurence of each of the faults in order to generate optimal alarm pattern recognition strategies (algorithm of the inference engine). A strategy helps the operator to recognize which alarm pattern is occurring without having to search the entire set of patterns.

Initial tests of the shell presented here show that it can be successfully used as a tool for the development of alarm pattern recognition expert systems. At present, an APRES for the condensate system of a nuclear power plant is currently under development using GENESIS.

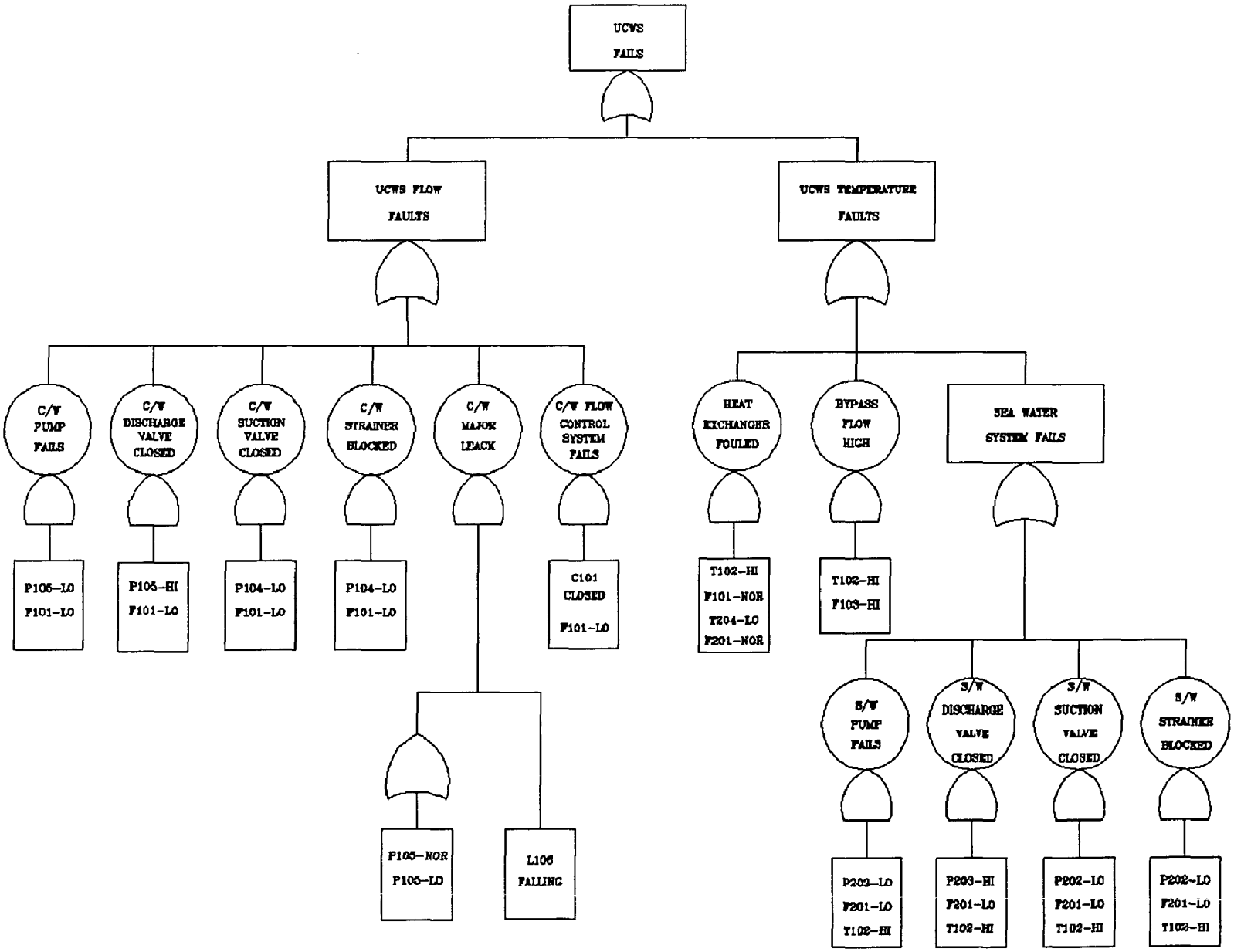

Figure 6. Extended fault tree for the UCWS.

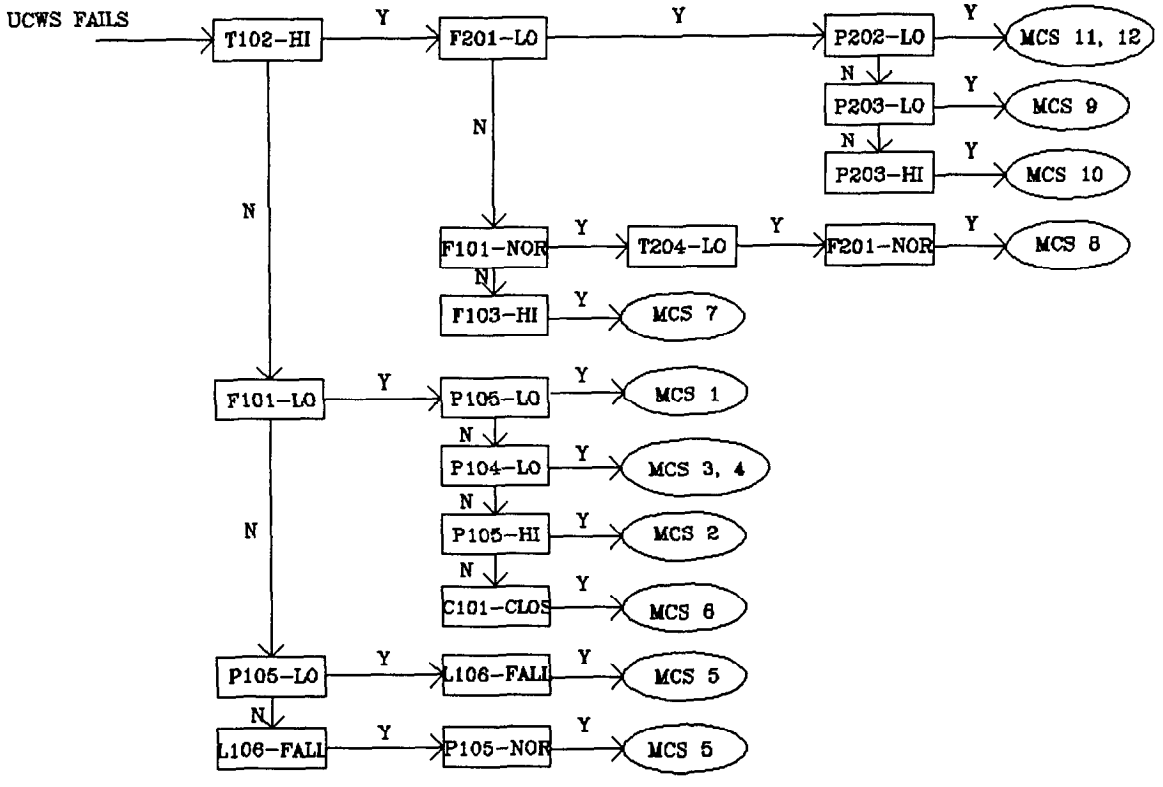

Figure 7. Alarm pattern recognition strategy constructed by GENESIS.

REFERENCES

1. R.L. Moore, "Artificial Intelligence Applications in the Power Industry: A Tutorial", Proceeding ISA-85, pp. 81-85, 1985.

2. W.E. Vesely, et.al., Fault Tree Handbook, U.S. Nuclear Regulatory Commission NUREG-0492, 1981.

3. P.K. Andow, "Improvement of Operator Reliability Using Expert Systems", Reliability-85, pp. 4B/4/4-4b/4/10, 1985.

4. J. Olmos, L. Wolf, "A Modular Approach to Fault Tree and Reliability Analysis", MIT Report MITNE-209, 1977.

5. USNRC, PRA Procedures Guide, NUREG/CR-2300, Vol. 1 and 2, 1983.

6. J. Arellano, "A New Approach for the Development of Alarm Pattern Recognition Expert Systems", UNIPEDE/CORECH-IERE 2nd. Workshop on Expert Systems, Milan, Italy, April 1989.

7. H.E. Lambert, "Fault Trees for Decision Making in Systems Analysis", UCRL-51829, 1975.