CONF-890782--1

DE89 012507

# DESIGN AND OPERATION OF THE COMMUNICATION ALARM PROCESSOR EXPERT SYSTEM

by

S. L. Purucker
B. E. Tonn
R. T. Goeltz
T. P. Wiggen
Oak Ridge National Laboratory*
Oak Ridge, Tennessee

K. M. Hemmelman
S. F. Borys
R. D. Rasmussen
Bonneville Power Administration
Vancouver, Washington



*ORNL is operated by Martin Marietta Energy Systems, Inc., under Contract No. DE-AC05-84OR21400 with the U.S. Department of Energy.

**Design and Operation of the Communication Alarm Processor Expert System**


Oak Ridge National Laboratory
S.L. Purucker
B.E. Tonn
R.T. Goeltz
T.P. Wiggen

Bonneville Power Administration
K.M. Hemmelman
S.F. Borys
R.D. Rasmussen



## ABSTRACT

This paper discusses the expert system design, verification testing, installation, and initial operating experiences of the Communications Alarm Processor (CAP), a prototype expert system developed for Bonneville Power Administration (BPA) by Oak Ridge National Laboratory. The system is designed to assist operators by receiving and diagnosing alarms from Bonneville's Microwave Communications System. The microwave system transmits data from power facilities in four Pacific Northwestern states to Dittmer, Bonneville's central operations control center. The prototype is limited to one of seven branches of the communications network and a subset of alarm systems and alarm types.

CAP receives real-time data, diagnoses operational problems, archives alarm information, and supports analysis aimed at improving equipment maintenance. The expert system operates in an advisory capacity so diagnoses are presented to operators for their review and concurrence before being archived. CAP employs a backward chaining approach for diagnosing alarms. The system, which was delivered in January 1989, resides on a VAX 3200 workstation under the VMS operating system and utilizes C code to retrieve and process alarm data. An expert system shell, Nexpert Object, was used to develop the expert system. The ultimate goal is to develop expert systems that will enhance power system operation and maintenance.


## 1.0 INTRODUCTION

Developers and users of energy management systems are beginning to explore ways in which expert system technology can provide assistance in managing power system operations data. Particular interest has focused on improving the management of alarm data which flow into power system control centers. This is an important problem because operators may not be able to assimilate all the data that the technology collects and delivers. Additional operator support may be required due to the increasing complexity of the power system and the need to respond quickly during crisis situations. Expert systems can be used to assist operators to prioritize and reduce alarms, diagnose and monitor alarm events, provide diagnostic consultation, optimize displays, help train operators, optimize maintenance, anticipate contingencies, analyze output from conventional power system codes, keep track of interconnection agreements, and suggest or implement control actions. This paper describes the prototype expert system developed by Oak Ridge National Laboratory for Bonneville Power Administration for processing alarms from Bonneville's microwave communication system.

The organization of this paper is as follows. Bonneville's power system operation and microwave communication system is introduced. Next, the architecture of the expert system is described. Then the design of the expert system is discussed, where particular attention is devoted to the hierarchial inferencing process, knowledge representation, and application of confidence factors. Two other sections provide testing and installation experiences. The last section, validation, discusses how CAP will be fine tuned to continue capturing the operator's knowledge and experience.

## 2.0 BPA MICROWAVE SYSTEM

The Bonneville Power Administration (BPA) owns and operates a 300,000 square mile power transmission network spread over four states in the Northwest. Bonneville's transmission system of almost 13,000 circuit miles of high voltage transmission lines is interconnected with 14 regional utilities at more than 150 points. Reliable operation and control of this large, complex power system requires extensive use of automation at substations and control centers. Advances in automation are necessary to keep abreast with increasing power system complexities due to system growth, reduced operating margins, complicated operating and control agreements, environmental constraints, and economic operation considerations.

To facilitate management of the power transmission system, Bonneville operates a region-wide microwave communications system for protective relaying, load and generator shedding, generator stability support, telemetering of critical quantities, supervisory control and data acquisition systems, and automatic generation control. The microwave system consists of seven major networks with 141 microwave stations -- 80 mountain-top repeaters and 61 substations (Fig. 1). Each microwave network consists of a main backbone with spurs to substations. To improve microwave system availability and reduce operating costs, Bonneville developed several automatic monitoring systems that measure performance and generate alarms to the Dittmer Control Center. Two of these monitoring systems, the Microwave Monitor (MWM) and the Badger system supply data to the CAP expert system.

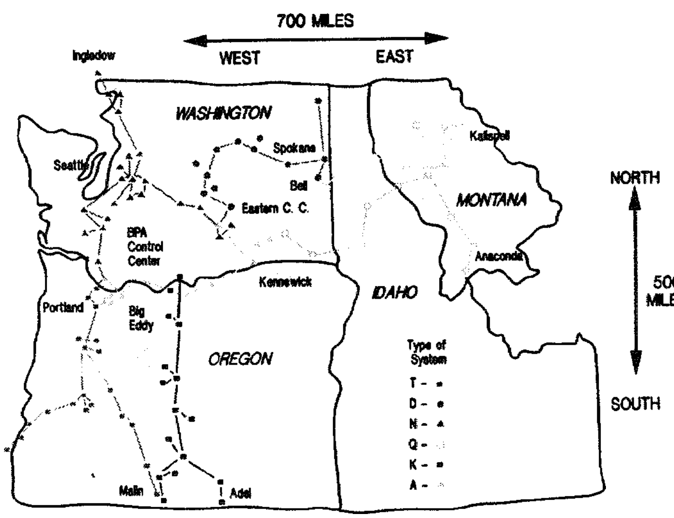


Fig. 1 BPA Microwave Communication System

## 3.0 CAP SYSTEM ARCHITECTURE

CAP's software architecture is shown in Fig. 2. It consists of three basic functions: processing real time alarm data, contributing the expert system's diagnosis, and interjecting the operator's expertise and judgement. Real-time alarm data from two independent alarm systems are read, compressed, filtered and buffered until the expert system is ready for new data. The expert system reads in new alarms, gets rid of old ones, then diagnoses an alarm using current alarm data. The expert system concludes its diagnostic action by writing the results to disk. The operator interface reads the diagnostic file and presents the diagnosis to the operator. The operator interface gives the operator the capability to log observations, examine alarm data, archive conclusions, and examine past diagnoses. Thus CAP supports two complimentary diagnostic sources: the expert system, which provides well-formulated diagnostic expertise and the operator, who retains control and can add his expertise when the expert system falls short.

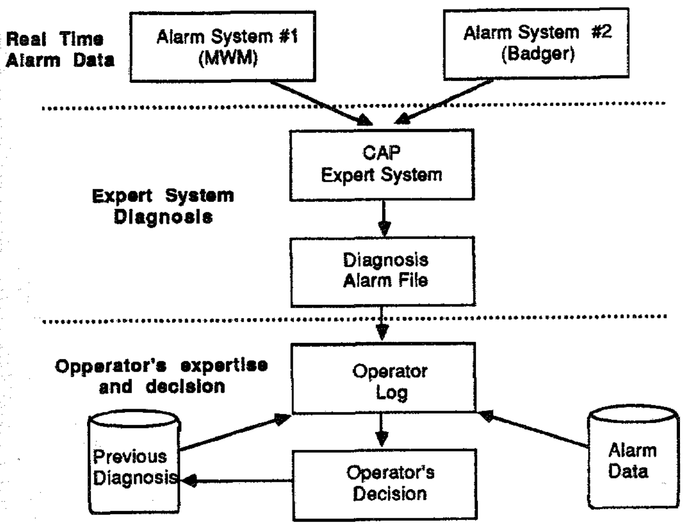


Fig. 2 CAP System Architecture

A feasibility study identified the following requirements:

(1) A powerful multi-tasking operating system;
(2) A workstation environment for building CAP;
(3) An expert system shell, based on C;
(4) A database/statistical analysis software for some CAP functions;
(5) Use of C computer language for other tasks.

The feasibility study also defined the scope of CAP. It recommended that CAP be kept to manageable dimensions by reducing input data and the expert system's responsibility. Data for the expert system comes from two real-time systems, the microwave monitor (MWM) and the Badger. Also, the prototype models only one of the seven microwave networks, the N system which is the most complex (see Fig. 3). With respect to alarms, the prototype diagnoses all MWM outage and performance alarms (two of the five alarm types). Configured in this manner CAP should diagnose 90% of the microwave alarms on the N system.

The microwave application also allows relaxation of two key operational parameters for the expert system, response time and control actions. The response time requirement is 30 seconds. Secondly, CAP acts as an advisor; no control actions are required. Thus, CAP was easier to develop than a system to process real-time transmission system alarms and is easier to integrate into Bonneville's organization because CAP does not initiate control actions.

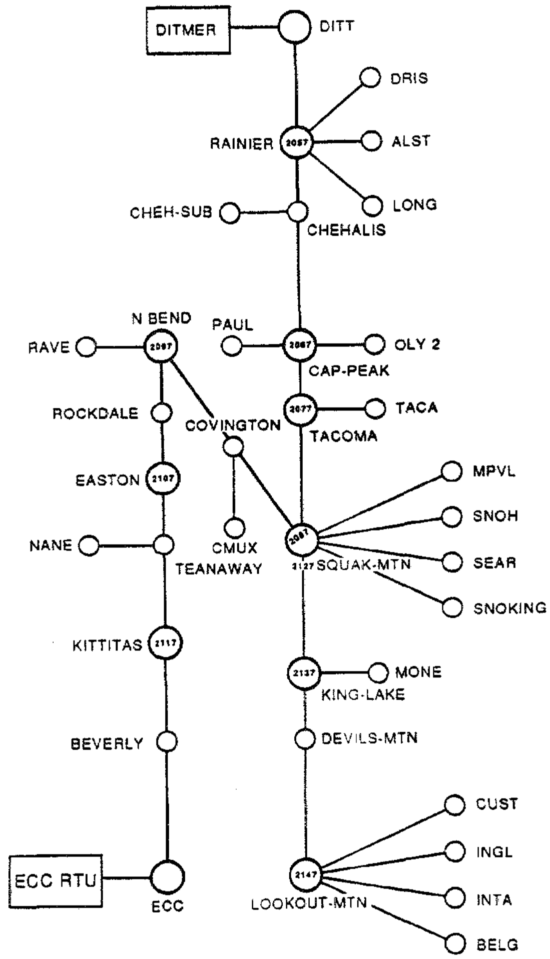


Fig. 3 N System

The feasibility study also identified a number of difficult research issues associated with prototype development. Among these issues are continuous operation, handling asynchronous real-time input data, accommodating alarm bursts, processing incomplete or uncertain data, impact of constant addition and deletion of alarm data, representing interrelated topology/alarm knowledge, acquiring and modeling the expert's diagnostic knowledge, expandability, maintainability, including operator judgement, integrating procedural software, and incorporating temporal logic. The project team believes these issues have been successfully addressed. CAP was installed in January 1989 and is currently being evaluated by BPA.

## 4.0 DESCRIPTION OF THE EXPERT SYSTEM MODULE

The expert system must be able to quickly process incoming alarms and provide understandable and timely diagnoses while addressing the issues identified in Section 3. It must also be able to utilize knowledge about microwave system alarm patterns and topology. The expert system satisfies these criteria by systematically processing incoming alarms, identifying highest priority alarms for diagnosis, focusing attention on areas or stations in the network, and generating all potential diagnoses with confidence factors. Figure 4 shows the four inferencing stages.

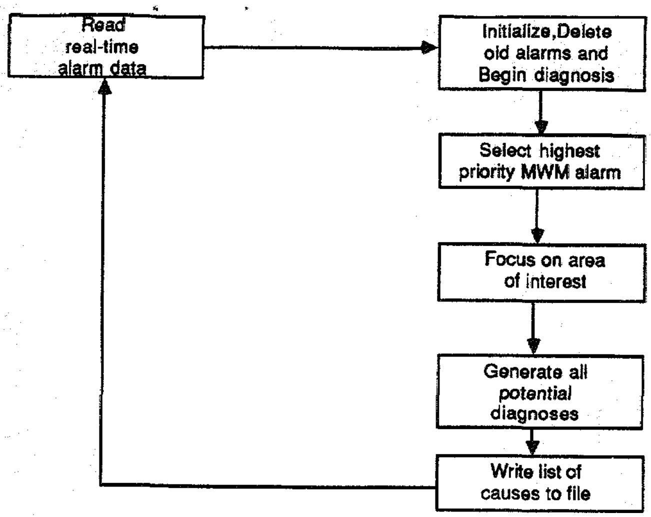


Fig. 4 CAP's four Inferencing Stages

CAP can be described from either the inferencing or knowledge representation viewpoint. We chose to emphasize inferencing because inferencing more naturally leads to discussions of representation. Inferencing refers to the manner in which the expert system operates. A description focusing on this viewpoint concentrates on explaining each inferencing step. The second viewpoint concentrates more on how knowledge and topology are represented in the expert system. As a result, representation issues are addressed as they arise during the inferencing descriptions. The four inferencing stages of the expert system are described below.

Initialize and Begin Diagnosis

The expert system's first inferencing stage determines which alarms within the expert system knowledge base need to be deleted. Alarms are deleted from the expert system fifteen minutes after they are cleared by field hardware. Next, the expert system reads in new MWM and Badger alarm data.

Each MWM alarm is represented as an object. MWM objects (alarms) are members of the mwm_alarms class. Each new alarm inherits values from its class's properties. In this case, the status of the diagnosis (diag) is initialized as "new_alarm". Figure 5 illustrates this concept. Each MWM alarm is described by nine pieces of information which are referred to as properties or slots. The first property is the name of the alarm (e.g., Noise outage, Backbone Level performance). The second property describes whether the alarm is new or has been diagnosed. Third, each alarm receives a unique id. Next, the source of the alarm is captured in the receiving station/sending station slots as well as its reporting field hardware. Seventh, the microwave network is recorded, which is always the N system in this prototype. Lastly, the time_in and the time_out of the alarm is recorded, most often the time_out slot is empty when the expert system begins its diagnosis.

The expert system also reads in badger alarms. These alarms are used as diagnostic evidence. All badger alarms are members of the badger_alarms class. Figure 6 illustrates this representation, which is similar to the MWM alarm representation. The properties of Badger alarms are: alarm type (e.g., receiver, transmitter, etc.), the alarmed station, an associated station, a unique alarm id, the time-in and time-out of the alarm.

After all new alarm data is read in, the expert system determines which microwave monitor (MWM) alarms require diagnosis. A MWM alarm becomes eligible for diagnosis ten seconds after it is received. This time delay ensures that associated Badger alarm data are present when diagnosis begins. If the expert system has no MWM alarms to diagnose, it goes to sleep (stops execution) until aroused by new alarm data.

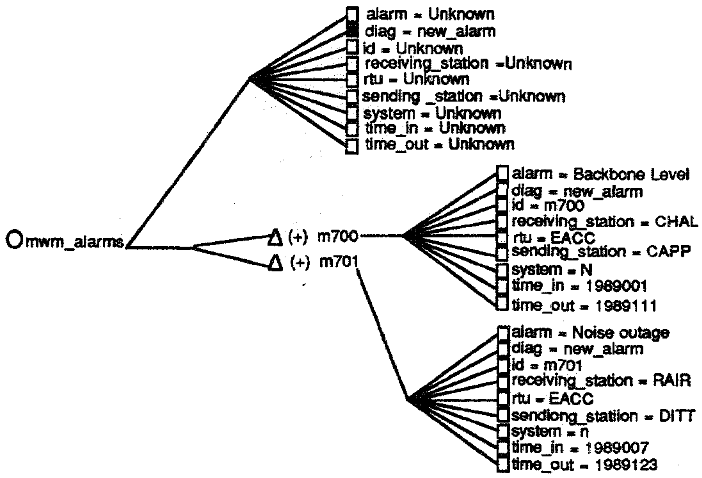


Fig. 5 MWM Alarms

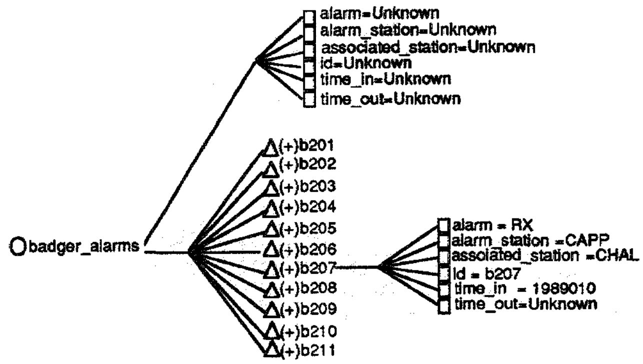


Fig. 6 Badger Alarms

Select Highest Priority MWM Alarm

Figure 4 shows that once the alarm data is received and represented, the expert system chooses the MWM alarm with the highest priority to diagnose. Table 1 contains the alarm priorities used by the expert system. When there is more than one undiagnosed alarm with the same priority, the oldest alarm is diagnosed first. The chosen alarm is then copied into an object called trigger, and more detailed investigation begins.

| Alarm Priority | MWM Alarm |
|---|---|
| 15 | Noise outage |
| 14 | Backbone Level outage |
| 13 | Baseband Load outage |
| 12 | Intermod outage |
| 11 | Spur Level outage |
| 10 | Phase Jitter outage |
| 9 | Turnaround outage |
| 8 | Noise performance |
| 7 | Backbone Level performance |
| 6 | Baseband Load performance |
| 5 | Intermod performance |
| 4 | Spur Level performance |
| 3 | Turnaround performance |
| 2 | Frequency Response performance |

Table 1 MWM Alarm Priorities

An example MWM alarm is used from this point on to help illustrate the inferencing process. The Backbone Level performance alarm, priority seven, was selected because its problem can be attributed to either station or hop (the microwave path between two stations) problems. Also, it is representative of topology considerations required to diagnose MWM alarms.

Focus on Area of Interest

This inferencing step considers topology. Most microwave monitor alarms don't identify which specific hop or station caused the MWM alarm. However, it does identify a section of the alarm network either by its alarm data or by association because it is a particular type of alarm. With the section information and a knowledge of the network, hops and stations are selected which can cause the MWM alarm. Diagnostics are then generated for all 'eligible' hops and stations. Thus, the MWM alarm focuses diagnostics to a subset of hops and stations.

A few words are in order concerning how the topological knowledge is represented in the expert system. Knowledge is retained in layers (refer to Fig. 7) where each lower level inherits property values from its predecessor. At the highest level, each network of the microwave system is identified separately, the n_system class for this prototype. The N system breaks into subclasses, defining noise sections, backbone level sections, spur level sections, badger alarms, and hops and stations. Hop and station data resides at the lowest or base level. Sections, hops and stations are prespecified in the expert system frame structure and are not determined during processing.

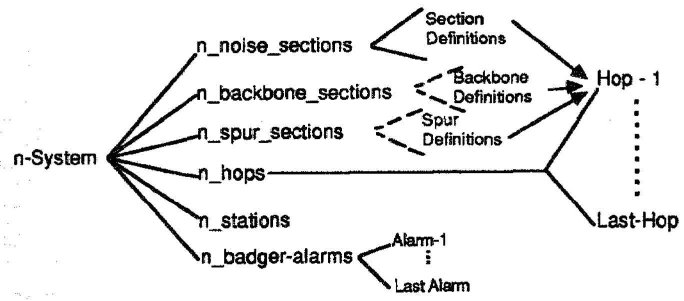


Fig. 7 Stations and Hops Inherit Topological Knowledge

Generate all Potential Diagnoses

Once the eligible hops and stations are selected based on the MWM alarm, the detailed diagnostic process begins. This stage also follows a backward chaining methodology, because the expert system is continually proving goals it sets for itself. However, this stage does not employ classical backtracking techniques. That is, in accomplishing the actual diagnosis of alarms, the expert system does not attempt to hypothesize possible causes for alarms and then attempt to prove conditions for those hypotheses. Instead, all potential causes are considered for each MWM alarm. The point to be made here is that due to unique features of this application, CAP uses an exhaustive backward chaining approach in evaluating all potential causes.

The expert's knowledge was captured by creating diagnostic fault trees for each type of MWM alarm. Fault trees were elicited from the Bonneville operators by the knowledge engineer. Figure 8 illustrates the fault tree for the backbone level performance alarm. The fault tree approach proved to be effective because the experts were able to visually inspect the knowledge destined for the expert system. The fault trees also fostered intellectual rigor because diagnostic procedures for different types of MWM alarms are compared to each other. Also, other experts can review the diagnostic logic. Inferencing as mentioned before is based on exhaustive backward chaining. Rules match predefined alarm patterns (fault tree) to current alarms. The operators confidence in a particular diagnosis increases as alarm data coincides with the diagnostic pattern.

The expert system considers each potential cause of a MWM alarm independently and calculates a confidence factor for each cause. Thus, with respect to a backbone level performance alarm, the expert system first explores a station problem, followed by investigations of potential transmitter and receiver hop problems. Small sets of Badger alarms provide evidence for each diagnosis. These diagnoses are supported by Badger ENTRY, transmitter (TX), receiver (RX), and noise differential (NDIF) alarms, if present for the hop being investigated. Each cause's confidence factor is calculated based on the value of evidence present. Figure 8 shows, if "A TX B" Badger alarm is true and "B RX A" alarm is not true, then a confidence factor of 80 is assigned to the cause "Transmitter Problem from station A to station B". Additional support for this cause would be forthcoming if either "B NDIF A" were true and "A NDIF B" were not true or "B RX A" were true and "A RX B" were not true. Inferencing is repeated for all eligible stations and hops.

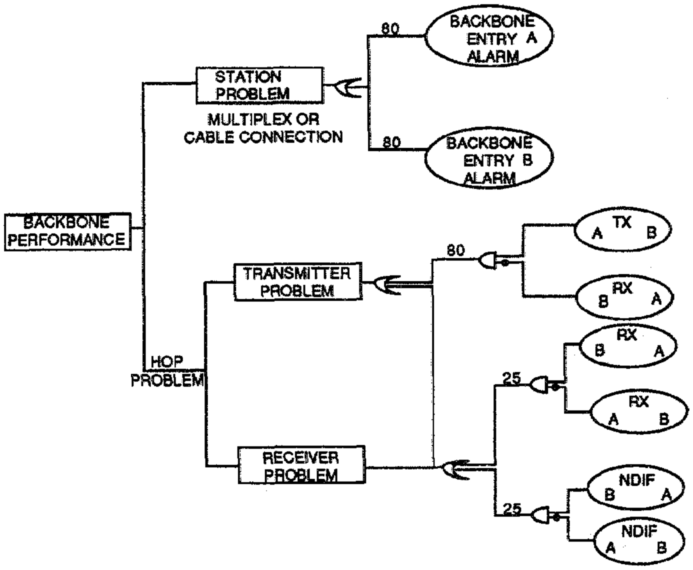


Fig. 8 Backbone Level Performance Alarm Fault Tree

Diagnosis concludes with an object being created for each cause, provided its confidence factor is greater than a threshold value. These objects belong to the class named 'causes'. For each instantiation of a cause, the cause's name, confidence factor, receiving and sending hop are transmitted to the operator's screen and diagnostic file. The operator reviews this file at his convenience, logs operator comments and selects the correct diagnosis. The chosen diagnosis is then stored in a historical file for future reference.

The Value of Confidence Factors

Confidence factors afford a number of benefits. First, they reflect operator judgement. Generally, when an alarm occurs the operator examines alarm data, uses his experience, then draws a conclusion establishing a cause. His reasoning process is based on observing alarm patterns, and weighing the evidence. Judgement reflects the operator's degree of conviction, in this case a number between 0 and 100. Secondly, the number of rules required to reflect the operator's knowledge is greatly reduced. If all contingencies are discreetly represented in the knowledge base, it would require in excess of 2000 rules, instead of the 230 rules used. The number of rules is closely related to the third benefit, maintainability and adaptability. A knowledge base consisting of 230 rules is much easier to maintain and change than one consisting of 2000 or so rules. Lastly, a smaller knowledge base means the expert system requires less memory and will probably run faster. Confidence factors simplified and enhanced CAP development.

## 5.0 VERIFICATION TESTING

The CAP prototype grew from about 30 rules with no external calls to the final version containing 230 rules and numerous calls to external C routines. CAP development plans called for a separation of coding and testing tasks. Testing was done in three stages, desk-checking the code, diagnosis based on small 'batch' alarm files and, diagnosis based on large 'simulator' alarm files. Each stage of testing represented a higher degree of integration.

Each portion of the rule-based code was independently evaluated (i.e., desk-checked or eyeballed) by an independent tester not involved in the original coding task. As a result, each rule, each object, each class, etc. in the expert system was viewed with suspicion until it made sense to the independent tester. This allowed the knowledge engineer to develop code faster because the code was being independently verified. A number of errors were detected in this desk-checking stage. As a matter of fact, most of the errors detected were identified in this stage. Error descriptions were passed back to the knowledge engineer who modified the code.

After the initial desk-checking step, the expert system was tested using small 'batch' files. Each fault tree's logic as well as the high level flow of control was checked by stimulating the expert system with selected alarm data. In this phase of testing, alarm data was read into the expert system from a file, essentially by the use of a READ instruction. Because of its simplicity, this method of testing was usable with the earliest versions of the CAP prototype. Typical input files consisted of 3-4 microwave monitor alarms (triggers) and 6-12 badger alarms. The expert system read in all the alarms, then it diagnosed each of the trigger alarms. All alarms were diagnosed within a minute, unless there was a problem. One objective was simply to confirm that the errors identified during the earlier desk-checking stage were corrected. A few additional bugs were discovered during this testing phase. This testing is not exhaustive due to the extremely large number of possible scenarios for alarm combinations. Basically, the alarm files which were constructed in this phase tested each branch of each fault tree at least once and tested for certain common types of mistakes (like failure to reset hypotheses before diagnosis).

"Simulator" testing is the final stage of testing. It involved using a software module which provides alarm data to the expert system in a simulated real-time environment. After the simulator was built and all known bugs had been removed from the system. The expert system got its alarms by calling a "messenger" module. The messenger checks to see if the real-time simulator is running; if it is, the messenger retrieves alarms from it, otherwise it receives alarms directly from import ports. The simulator gets its data from an alarm file updated with current time stamps. This provides the messenger with simulated alarm data, as if the alarms were coming from the input ports. Thus, we were able to test the interface between the messenger and the expert system as well as the ability of the expert system to survive in a laboratory real-time environment. Most of the bugs found were in the interface between the messenger and the expert system. For instance the messenger, from time to time, put input data in the wrong area of memory, causing difficulties. These problems were corrected.

We were able to test the system continuously, providing the simulator with an inexhaustible supply of alarm data. Actually, the simulator was in a loop, reading the same alarm file every 15-20 minutes. These test files were relatively large having 50-100 trigger alarms and 50-200 badger alarms. After the simulator finished reading the file, the time stamps were updated and the process repeated. One feature which we were unable to test in the 'batch' testing mode was the deletion of old alarms. The expert system is supposed to delete alarms which are cleared longer than 15 minutes. Deleting old alarms is essential for continuous operation, since memory must be reclaimed and reused. The real-time simulator provided us with this capability. The simulator also allowed us to test continuous diagnosis and arrival of new alarms.

In summary, testing was done on a qualitative basis. Each set of alarm data, whether designed for the batch mode or the more sophisticated real-time simulator, was constructed in an ad hoc manner. Testing guidance was provided by: (1) code which looked suspicious to the reviewer, (2) exhaustive testing of each branch on each fault tree, including several branch combinations, (3) insight gained from problems discovered in earlier test runs, and (4) potential problems associated with continuous operation. For instance, one series of tests stimulated the expert system twice with the same problem. Duplicate diagnoses assure the tester that the expert system is reinitializing correctly. When any error was detected, tests were rerun after coding changes to ensure that these errors had been corrected. Continuous simulator tests tested all the fault trees and ran until the system either died or was stopped. One purpose of the continuous test was to saturate CAP with alarms to see if it would crash. Complete testing of all alarm combinations for all network components was not feasible, since this would have required more than 100,000 test scenarios.

## 6.0 INSTALLATION EXPERIENCES

This section describes installation and preliminary operating experiences. Two general classes of experiences occurred: (1) unanticipated problems that arose during software installation and (2) performance of the expert system.

CAP installation problems were no different than those one might expect with any software project. The expert system did not cause any unique problems. Our first problem was caused by differences between delivery and development system software. The operating system and statistical database were different versions, causing several CAP software changes. The next installation problem was traced to incorrectly coded operational characteristics of Bonneville's serial computer communication ports. The solution involved a trial and error coding process. The port communications problem did not surface during development because Oak Ridge's system did not access alarms from the real-time system. Next, the alarms contained undocumented characters, null characters to compensate for the logging devices's slowness. Unfortunately, the 'C' program which reads alarm data from the input port required extensive modification and redesign. Finally, field testing the software while in on-line mode (as opposed to simulation mode) was difficult because the developer had no control over input data. For example, the microwave alarm systems were very quiet, so the developer had to wait for long periods of time for data to arrive.

Initial experiences with the run-time characteristics of the expert system are positive. The response time of the expert system module, when measured as the difference between the alarms arrival at the expert system and the expert system's diagnosis, was always less than five seconds. Also, the CAP prototype demonstrated successful continuous operation. Simulation tests showed that, under very heavy input data loads, the memory allocation of the expert system grew continuously and eventually caused serious degradation of the expert system's performance. Once the expert system's garbage collection routines were tuned, actual field experiences showed the CAP prototype does not experience these problems. A shortcoming of CAP is that its users at Bonneville believe the granularity of some of the diagnoses are too vague to be useful (e.g., "station problem at location X"). Therefore, additional knowledge acquisition based on experiences is to be done. This means the fault trees need to be fine tuned and the expert system modified to reflect operational experiences.

## 7.0 VALIDATION

Validation occurs during field operation (after installation and verification). Verification tests assure developers that CAP is doing what it was designed to do (i.e., CAP processes alarms accurately according to the fault trees). Validation examines how well CAP does its intended job (i.e., the fault trees are complete and accurate). Validation requires operator input and direction. As anticipated, preliminary operating experience shows that the fault trees need to be refined. Over time, additional operating experiences will probably bring to light new situations which were not anticipated. To improve CAP, the scenarios must be documented, then the fault trees and expert system refined to address these and other anticipated situations.

Validation also includes determining how well the operator interface does its intended job. Early operating experiences indicate that the interface needs to allow the operator to examine alarm data present when the MWM alarm was diagnosed. Since operators may not review noncritical MWM alarm diagnoses for several hours, CAP must take a 'snap shot' of the alarm knowledge base and provide the operator the option of reviewing alarm states at the time the MWM alarm was diagnosed.

BPA will evaluate CAP over the next year. One of the reasons BPA is interested in expert systems is this technology's adaptability and capability to capture operator expertise. They will determine if expert system technology can do the job for BPA operators and if so, what the next step should be in providing additional operator support.

## 8.0 SUMMARY

CAP is designed to support the operator. The operator makes the final decision and establishes a cause for each problem. Testing and installation experiences are discussed as well as how the system is being refined and validated. The expert system operates in a logical and predictable manner. It systematically processes incoming MWM and Badger alarms, identifies the highest priority alarm for immediate diagnosis, recognizes candidate stations and hops, and exhaustively explores all potential causes of the MWM alarm on each candidate station or hop. Diagnoses include a confidence factor expressing the operator's confidence in each cause. Confidence factors have a number of benefits. Among them is the capability to convey the value of supporting alarm data reflecting the operator's judgement. The expert system is based on the operator's well-known experience. The operator interface facilitates operator interaction for less well-known situations. The operator is in control of the expert system not the other way around.

## REFERENCES

1. M. A. Street and S. F. Borys, "Microwave Performance Monitor Application at the Bonneville Power Administration," IEEE/PES 1987 Winter Meeting, February 1987.

2. B. E. Tonn, R. T. Goeltz, and S. L. Purucker, "Expert Systems and Microwave Communication Systems Alarms Processing: A Feasibility Study," Oak Ridge National Laboratory, ORNL/TM-10429, July 1987.

3. Electric Power Research Institute, "Artificial Intelligence Technologies for Power System Operations," Palo Alto, California, EL-4323, 1986.

4. B. Wollenberg, "Feasibility Study for an Energy Management System for Intelligent Alarm Processing," IEEE-PICA, 1985, pp. 249-254.

5. S. L. Purucker, B. E. Tonn, R. T. Goeltz, K. M. Hemmelman, R. D. Rasmussen, and S. F. Borys, "Communication Alarm Processor Expert System," proceedings of the Symposium on Expert Systems Application to Power Systems, Stockholm-Helsinki, August 22-26, 1988.

6. M. Schwarzblat and J. Arellano, "An Expert Diagnostic and Prediction System Based on Minimal Cut Sets Techniques," CH2215, proceedings of the Western Conference on Expert Systems, June 1987.

7. K. Parsaye and K. Y. Lin, "An Expert System Structure for Automatic Fault Tree Generation for Emergency Feedwater Systems for Nuclear Power Plants," CH2463, proceedings of the Second Conference on Artificial Intelligence Applications, December 1985.