ORNL/TM-11323

ornl

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

Bonneville Power Administration Communication Alarm Processor Expert System: Design and Implementation

R. Goeltz
S. Purucker
B. Tonn
T. Wiggen
D. MacGregor

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

ORNL/TM-11323

ENERGY DIVISION
Transportation and Systems Research Section

BONNEVILLE POWER ADMINISTRATION COMMUNICATION ALARM PROCESSOR EXPERT SYSTEM: DESIGN AND IMPLEMENTATION

Richard Goeltz
Steven Purucker
Bruce Tonn

Thomas Wiggen
Oak Ridge Associated Universities

Donald MacGregor
MacGregor-Bates, Inc.

June 1990

Research sponsored by the Office of Engineering and the Office of Operations, Maintenance, and Construction
Bonneville Power Administration

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
Operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
Under Contract No. DE-AC05-84OR21400

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This report describes the Communications Alarm Processor (CAP), a prototype expert system developed for the Bonneville Power Administration by Oak Ridge National Laboratory. The system is designed to receive and diagnose alarms from Bonneville's Microwave Communications System (MCS). The prototype encompasses one of seven branches of the communications network and a subset of alarm systems and alarm types from each system. The expert system employs a backward chaining approach to diagnosing alarms. Alarms are fed into the expert system directly from the communication system via RS232 ports and sophisticated alarm filtering and mailbox software. Alarm diagnoses are presented to operators for their review and concurrence before the diagnoses are archived. Statistical software is incorporated to allow analysis of archived data for report generation and maintenance studies. The delivered system resides on a Digital Equipment Corporation VAX 3200 workstation and utilizes Nexpert Object and SAS for the expert system and statistical analysis, respectively.

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 are generated in power system control centers. This is an important problem because, in crisis situations, control center operators cannot cope with the volume and rate of arrival of alarm data. Expert systems have the potential to provide a number of services, including: reducing alarm data, filtering alarms, monitoring alarm events, diagnosing alarm events, providing diagnostic consultation, prioritizing alarms, optimizing alarm displays, and suggesting or implementing control actions.

This report describes a prototype expert system developed by the Oak Ridge National Laboratory (ORNL) for the Bonneville Power Administration to process alarms from Bonneville's Microwave Communication System (MCS). The organization of the report is as follows. First Bonneville's power system operations and the domain of the expert system are introduced. Next, the recommendations of a feasibility study that preceded the current work are summarized. In Section 3, the expert system module is described. The fourth section details the software developed to collect the alarms from the MCS, input the alarms into the expert system and archive the results. Sections 5 and 6 describe statistical software used to analyze archived data and the entire system's interface, respectively. Appendix A contains alarm fault trees used to construct the expert system knowledge base.

2.0 BACKGROUND

2.1 BONNEVILLE POWER ADMINISTRATION SYSTEM OPERATIONS

The Bonneville Power Administration is part of the U.S. Department of Energy and has responsibility for transmitting power from federal power generation facilities to utilities in the U.S. Pacific Northwest. Bonneville's Dittmer Control Center is the hub of activities concerned with the safe, reliable, and economic operation of the Federal Columbia River System, which includes Bonneville's transmission network. 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[1]. The network covers a 350,000 square mile region that encompasses four Pacific Northwestern states: Oregon, Washington, Idaho, and Montana.

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 considerations.

To facilitate management of the power transmission system, Bonneville operates a region-wide microwave communications system for protective relaying, load and generator dropping, telemetering of critical quantities, Supervisory Control and Data Acquisition (SCADA) systems operation, and management of Automatic Generation Control (AGC) (Fig. 1). The microwave system consists of seven major networks with 141 microwave stations -- 80 mountain-top repeaters and 61 substations. Each microwave network consists of a main backbone with spurs to substations (Fig. 2 illustrates one of the seven networks called the N-System). To improve MCS availability and reduce

ORNL-DWG 88Z-12830

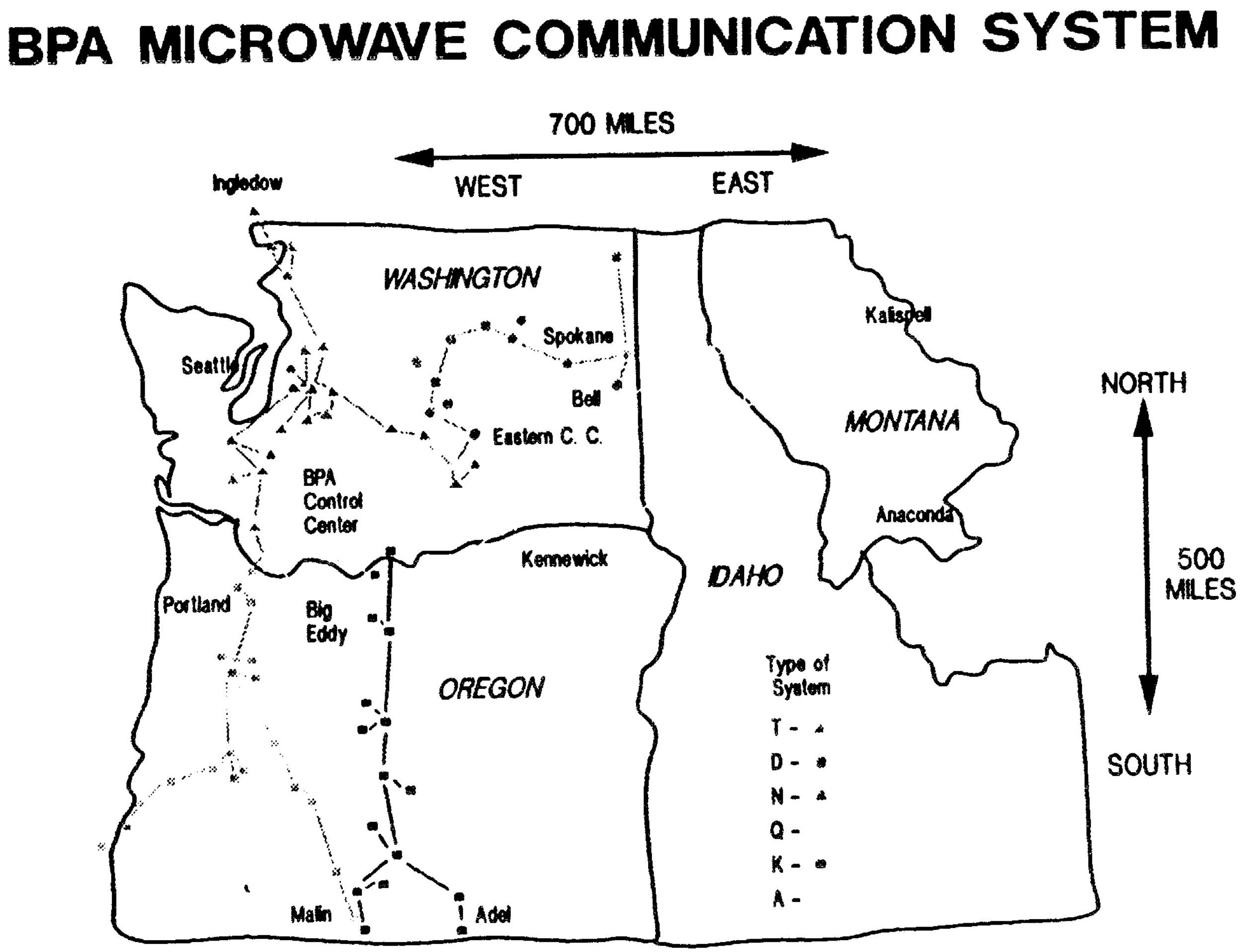


Figure 1. BPA Microwave Communication System

operating costs, Bonneville developed several automatic monitoring systems that measure MCS performance and generate alarms to the Dittmer Control Center[2].

2.2 EXPERT SYSTEM DOMAIN

Over the years, Bonneville has been very successful in designing and developing advanced technologies to collect power system data. The next logical step is to develop advanced technologies and methods to process alarm data to support operations and maintenance decision making. The Bonneville Power Administration has identified expert systems as one such technology that possesses great potential. A major reason for this conclusion is that Bonneville has numerous and complementary prospects for expert system applications. At Dittmer, for example, there are a number of potential expert system applications involving on-line processing of operations data.

All of these potential application areas are complementary because operations data essentially share a common data format:

[date/time...location...alarm-message...occur or clear]

An alarm occurs on a specific date at a specific time and location. The typical message identifies a specific equipment or system problem. An alarm exists in one of two states: an "occur" state indicates the initiation of an abnormal situation; a "clear" state tes when the abnormal situation has ended. Often, an alarm condition will toggle, creating a stream of alarm messages. Other alarms may be "open" for days.

In addition to this format, the various systems which accumulate alarm data at Dittmer share several other characteristics: (1) they involve large amounts of data, (2) the rate at which messages are received can be very high, (3) sufficient time for humans to process alarm data may not always be available, and (4) the alarm data are complex enough to require expertise for interpretation.

At the very least, lessons learned from developing an expert system for any one alarm system application could be readily transferred to any other application. However, it may be possible to develop a generalized expert system solution technique for other alarm systems that have the above data format and characteristics.

The potential benefits from expert systems are substantial. On-line expert systems technology could benefit Bonneville by: reducing alarm data, filtering alarms, prioritizing alarms, diagnosing problems, and monitoring situations. Because expert systems are expected to perform a significant amount of data reduction, refinement, and interpretation, the resulting data will provide a wealth of information on the performance characteristics of the monitored equipment. By using conventional statistical analysis methods to study equipment performance over time, Bonneville will be able to improve equipment maintenance, planning, design activities, and possibly even optimize its maintenance schedules. Thus, in general, successful application of expert system technology has the potential to improve operation system reliability and reduce maintenance costs.

2.3 THE FEASIBILITY STUDY

In 1987, Bonneville requested ORNL to study the feasibility of using an expert system to perform power system alarm processing[3]. Drawing on power system-artificial intelligence studies by the Electric Power Research Institute[4] and Control Data Corporation[5], the ORNL study determined that potential Bonneville applications were amenable to expert systems technology. The feasibility study also provided a sound foundation for Bonneville's first expert system prototype, because it: (1) rigorously defined an application area; (2) specifically defined the prototype domain; (3) evaluated hardware and software tools; and (4) identified and assessed research issues and implementation challenges.

The Microwave Communications Alarm Processor (CAP) was chosen as the first prototype application for a number of important reasons. First, the four separate alarm systems that comprise Bonneville's MCS share the data format described previously. Second, the MCS application allowed relaxation of a number of operational parameters of the expert system, including volume of input data to be processed (only about 1000 alarm messages per day are expected), resolution of date/time stamps to be manipulated (only a one-second resolution is required), complexity of diagnosis to be performed (the MCS is straight-forward to model), and response time required (within 30 seconds is desired). Finally, the CAP will operate as an advisor; no control actions will be taken. Thus, CAP would be easier to develop than a system to process real-time transmission system alarms and would be easier to implement in Bonneville's organization because CAP will not implement control operations.

Even given CAP's relaxed operational parameters, the feasibility study still identified a number of very difficult research issues associated with prototype development (discussed in Sect. 2.6). Therefore, the study recommended a limited prototype domain. First, of the seven microwave networks, only one, the N-system, is encompassed by CAP (see Fig. 2). Second, only two of the four possible microwave alarm systems are used for input data. One, the Microwave Monitor System (MWM) provides alarm triggers for expert system diagnosis while the other, the Badger system, provides corroborating evidence. The MWM generates alarms related to how well microwaves are being transmitted between stations whereas Badger alarms are related to microwave station equipment problems. Use of these two systems should allow diagnosis of over 90 percent of abnormal microwave system events. Finally, only two of the four classes of MWM trigger alarms are addressed by the prototype. These are the most important classes to the MCS operations and maintenance staff and are described in more detail in Section 3.

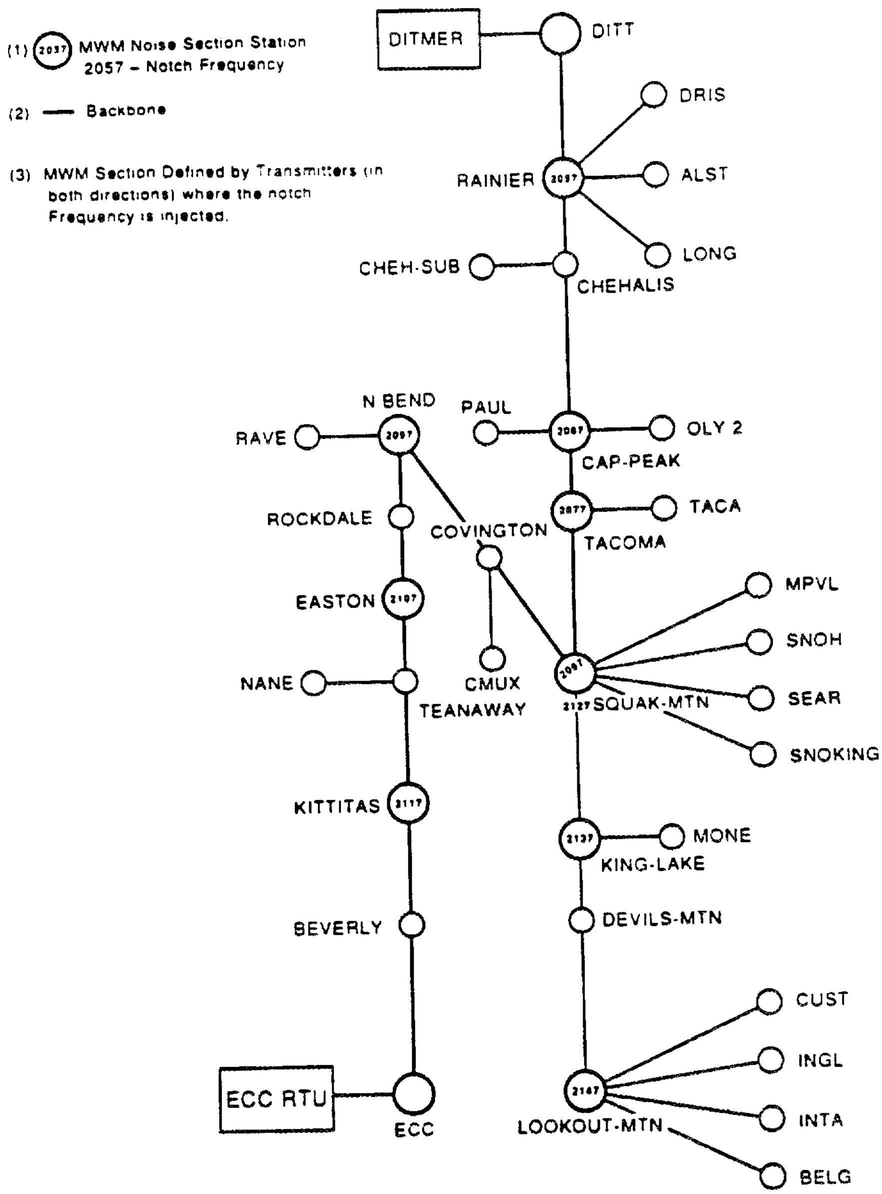


Figure 2. N System with MWM Noise Sections

2.4 COMMUNICATIONS ALARM PROCESSOR (CAP) SYSTEM ARCHITECTURE

This section describes the architecture of CAP, which is illustrated in Fig. 3. Separate modules handle the input alarms, the expert system, the archive database, the statistical package, and the interface. The first task that CAP must accomplish is the input of the MWM and Badger alarms. Currently, incoming alarms are stored in two buffers, called the Input Data Buffers (IDBs). As described in Section 4, this "C"-coded software also manages alarm records and passes the alarms to a "mailbox" for the expert system. This software also passes the processed alarm data to archive databases.

The alarms manager reads messages placed in the mailbox by the IDBs and inserts new alarms, or modifies existing alarms, in the fact base of the expert system. The alarms manager also extracts from the expert system the diagnosis for each trigger MWM alarm. At any convenient time, the operator is able to access the alarm conclusion files and confirm which, if any, of the conclusions are correct. Confirmed conclusions are then archived. Unconfirmed conclusions are stored in yet another file for future consideration.

The statistical package is the basic, commercially available software shell sold by SAS[6]. As described in Section 5, a user is able to invoke SAS routines to analyze archived alarms and produce periodic reports. The interface design, described in Section 6, allows a user to start and stop each module and to peruse the various files. The entire architecture is unique in that it encompasses a mixture of off-the-shelf software and specially coded software, all designed to operate in a real-time, but asynchronous manner.

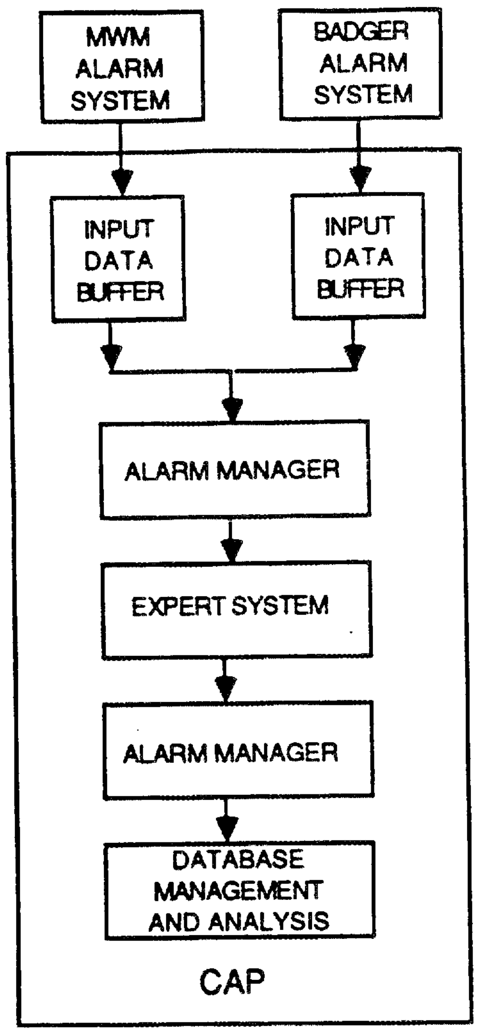


Figure 3. Architecture of CAP

2.5 HARDWARE AND SOFTWARE CHOICES

As mentioned previously, the CAP feasibility study evaluated and recommended expert system shells, computer hardware and operating systems, and database management software. The evaluations focused first on the fit between problem characteristics and capabilities of various expert system shells. Next, the characteristics of the computer hardware and operating systems on which the shell operated were carefully evaluated. These hardware and software characteristics are discussed below.

Nexpert Object, by Neuron Data (Palo Alto, Calif.) is the expert system development system used for CAP. Nexpert is rule-based and supports class-based objects, inheritance of properties, methods, and links between objects. The inference engine supports both forward and backward chaining and uses the same rule format for both. Rules can be prioritized for conflict resolution. Also, the developer has control over which actions affect the expert system's agenda.

Nexpert Object has a number of features that facilitate integration of the expert system into the CAP architecture. First, Nexpert can call conventional programming languages from both the left-hand side and the right-hand side of the production rules. More importantly though, a library of subroutine calls is available which allows manipulation of the expert system's fact base and allows control of the inference engine from any conventional programming language. Also, Nexpert provides hooks into the inference cycle and other important shell services such as the interfaces. These facilities allow customization of the shell's components, and they provide the support necessary in building data paths between the asynchronous alarm data, both input and output, and the expert system.

Digital Equipment Corporation's (DEC) VAXstation is the computer hardware being used for the prototype. The central processing unit (CPU) has adequate power; a

1-million-instructions-per-second (MIPS) CPU has been used for development and a 3-MIPS CPU for delivery. DEC's VMS operating system supports multitasking, interprocess communication, process priorities, and a large address space. These computer capabilities support the multiple, specialized, and cooperating CAP processes that the solution architecture requires. Also, the VAX-station supports the large primary and secondary memories needed for this application. Finally, color graphics, windows, and a mouse pointing device are important for providing effective user interfaces. Lastly, as mentioned above, SAS has been used both for statistical analysis and database management.

2.6 RESEARCH CHALLENGES

There are a number of significant challenges to developing on-line, real-time expert systems[7]. In this section, we list many of the research challenges these applications present and briefly assess how the CAP project is dealing with them. The purpose of this section is to highlight technical topics that are covered in more detail in the remaining sections.

Asynchronous data: One technique used in CAP to handle asynchronous data is the use of data buffers. Each data path between two processes uses a data buffer of some type. The operating system provides buffers between the input ports and the IDBs. Also, the expert system mailbox is between the IDBs and the expert system. Finally, a disk file holds data passed from the expert system to the user interface.

Uncertain or missing data: An uncertainty representation must be able to indicate strong support for diagnoses, when warranted, and provide clear discernment between diagnoses. Certainty factors are used in CAP which satisfy the first of the two criteria. Future work will focus on implementing belief functions in CAP, which may provide more discernment between diagnoses and will provide a rigorous approach to handling alarms with incomplete records and to updating beliefs with historical alarm data.

Expert system/operator interface: The operator must find CAP easy and valuable to use. Currently, CAP has on-line help facilities and menu-driven interfaces. Field testing will provide insights into future improvements of the interface.

High performance: An important concern is that CAP keep abreast of incoming alarms. Careful scoping of the prototype has been our primary means of dealing with performance concerns. On the other hand, our choice to use conventional hardware and, for the most part, commercial software for the prototype, limits our ability to customize solution elements for improved performance.

Nonmonotonicity: The nonmonotonic nature of the CAP processes (e.g., constant addition and deletion of alarm data which could immediately affect the activities of the expert system) is not yet well understood. Field testing will provide valuable insights into this research challenge.

Temporal reasoning: Concepts are being developed concerning "windows of inference", aged data, and manipulation of time intervals. However, the full extent of the difficulties of temporal reasoning with the CAP alarm data has not yet been measured.

Focus of attention: The expert system tool, Nexpert Object, has sufficient facilities to guide inference, prioritize operations, and generally control the agenda of eligible rules to be fired.

Integration with procedural components: The prototype expert system shell is well integrated with the operating system and conventional programming languages.

Guaranteed response time: No research has been conducted on guaranteeing CAP response times. Instead, this topic will be explored during the field testing stage.

In summary, careful scoping of the prototype has been the most important means of dealing with the challenges listed above, followed by the extensive evaluation of problem characteristics and characteristics of expert system software tools. However, a number of interesting and challenging research issues remain to be addressed before this expert system technology can be fully integrated into Dittmer's power system operations. The following sections describe in detail each of CAP's modules.

3.0 DESCRIPTION OF THE EXPERT SYSTEM MODULE

This section describes the expert system module of CAP. The expert system was designed and developed with respect to numerous criteria and constraints. Not only must the expert system be able to quickly process incoming alarms and output understandable and timely diagnoses, but the system must also be able to utilize knowledge about the microwave communications system and information about the MCS 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 exploring exhaustively all potential diagnoses.

CAP has a novel and complex expert system architecture, the technical details of which can be described from two viewpoints, inferencing and knowledge representation. Inferencing refers to the manner in which the expert system operates. A description focusing on this viewpoint would make use of a flow diagram and concentrate on explaining each step in the inferencing process. The second viewpoint concentrates more on how knowledge and topology are represented in the expert system. We have chosen to emphasize inferencing over representation in the following discussion because an understanding of inferencing more naturally leads one into discussions of representation. As a result, representation issues are addressed as they arise during the inferencing descriptions.

Figure 4 presents a flow diagram of the inferencing process at work inside of the expert system module. In general, the process may be described as following a backward chaining methodology. This is because the expert system is continually attempting to satisfy goals it sets for itself. This will become clear as the discussion proceeds. It must be made clear, however, that the expert system does not employ classical backtracking techniques. That is, in accomplishing

FLOW

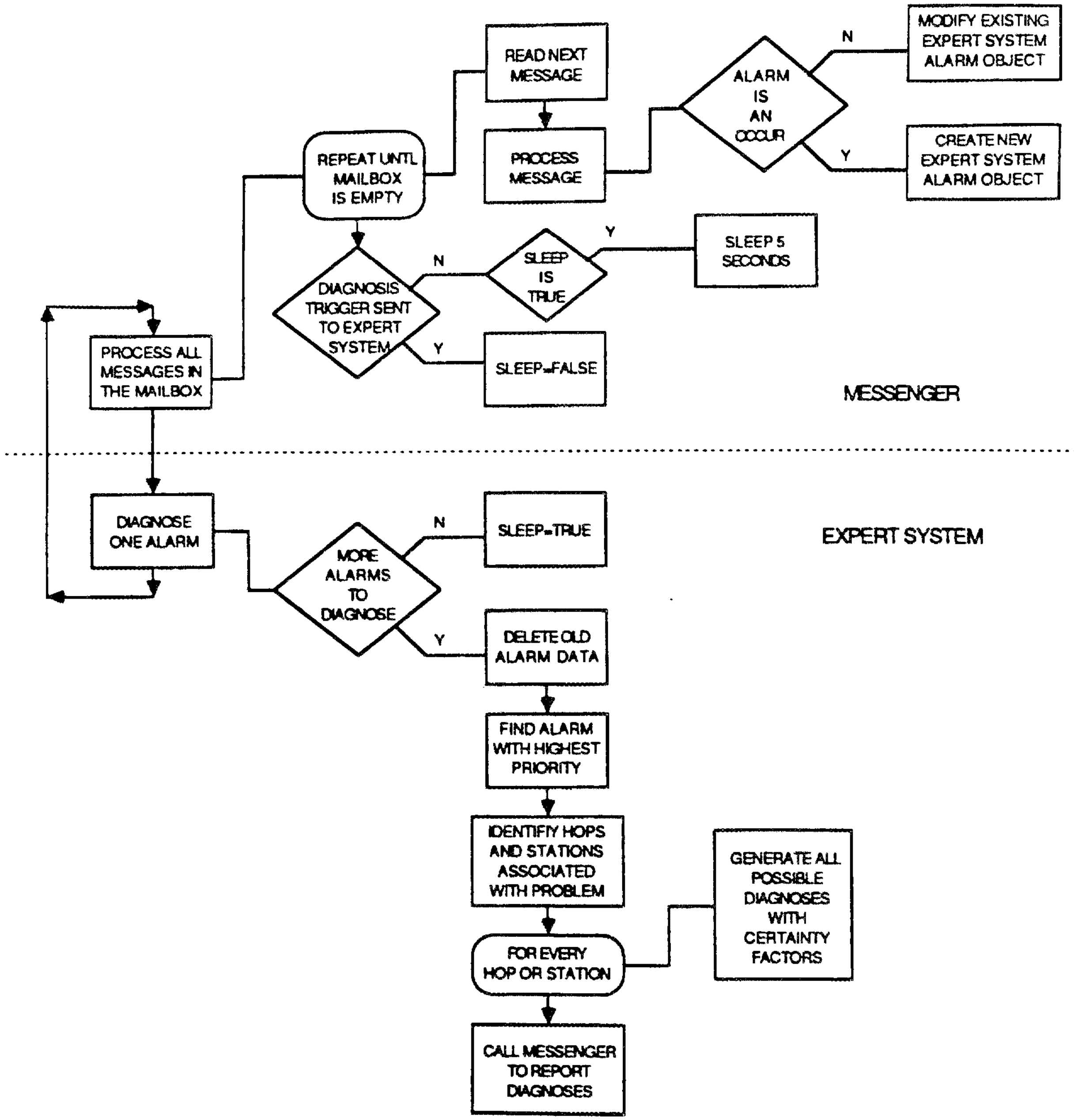


Figure 4. Expert System Flow Diagram

the actual diagnosis of alarms, the expert system does not attempt to hypothesize possible causes for alarms and then attempt to instantiate conditions for each possible hypothesis. Instead, all potential causes are considered for each potential alarm. How the expert system accomplishes this is also described in more detail below. The point to be made here is that due to unique features of this application, CAP is not a classical backward chaining application.

The expert system runs continuously by repeating a diagnosis cycle with two steps: (1) a subroutine of Messenger processes all new alarm messages in the expert system's mailbox and (2) the expert system performs a diagnosis on one MWM alarm (Fig. 4). There are two important characteristics of the design: (1) the input of new alarm data is synchronized with the inference process and (2) new alarm data is incorporated into the inference process as quickly as feasible.

A variable named "SLEEP" is maintained by both Messenger and the expert system and is used to communicate the internal state of the mailbox and the expert system. When there is no work for the expert system to perform, SLEEP is true and Messenger causes the process to pause for a specified time. This allows the computer to work on other tasks.

The expert system will initiate the diagnostic process upon receipt of a Microwave Monitor alarm. Each MWM alarm is represented as an instantiation (i.e., an object) of the class of objects known as MWM alarms. Figure 5 illustrates this approach to representing alarms in the code of the expert system. Each MWM alarm is described by eight pieces of information. The first piece is the name of the alarm (e.g., noise outage, turnaround outage). The second piece describes whether the alarm is new or has been diagnosed. Third, each alarm receives a unique identification (ID) generated by a sequential number generator in the IDB. Next, where the alarm emanated is captured in the receiving station/sending station slots. Fifth, the branch of the MCS is recorded, which is always the N-system in this prototype. Lastly, the time-in and the time-out

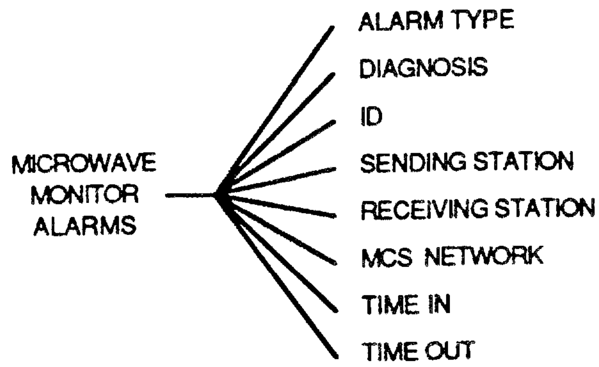


Figure 5. Microwave Monitor Alarm Knowledge Representation

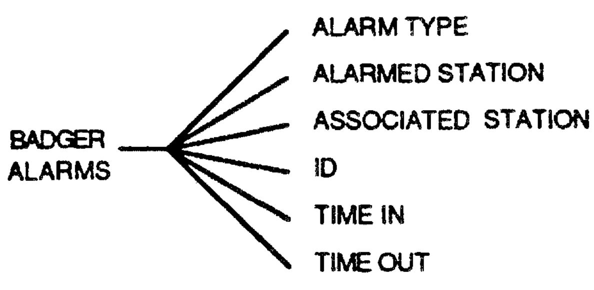


Figure 6. Badger Alarm Knowledge Representation

of the alarm is recorded, although often times the time-out or clear may not be received before the expert system begins its diagnostic processes.

Once each new MWM alarm object is specified according to the representation discussed above, the expert system addresses Badger alarms that may be useful in the diagnostic process. All the new Badger alarms are represented as instantiations of the class of objects known as Badger alarms. Figure 6 illustrates this representation, which is very similar to the MWM alarm representation. The six pieces of information for each Badger alarm are: alarm type (e.g., receiver and transmitter alarms), the station from which the alarm emanated, an associated station, a unique alarm ID, and the time-in and the time-out of the alarm.

Next, the expert system chooses the MWM alarm with the highest priority to diagnose first. Table 1 presents the priorities used by the expert system. When there is more than one alarm with the same priority, the oldest alarm is diagnosed first. The chosen alarm is then copied into a class of alarms, known as trigger alarms, and is then ready to investigate.

The next step of the inferencing process addresses topological issues. More specifically, for several types of Microwave Monitor alarms, it is not possible to determine from the alarm data exactly which hop or station caused the MWM alarm.[1] This is due to limitations within the architecture of the MCS. However, the section of the alarm network can be determined. With this section information, knowledge of the network is used to determine the hops or stations in the section from which the MWM alarm might have come. The diagnostic process is performed over each eligible hop or station for each MWM alarm.

---

[1]A hop is a microwave path between two MCS stations. A section may contain one or more hops.

Table 1. Microwave Monitor Alarm Priority Rankings

Alarm Priority	Microwave Monitor 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

Before continuing with the discussion of the process, a few words are in order concerning how the topological information is represented in the expert system. First, each network of the MCS is represented as an instantiation of a class of objects known as "networks." Each MCS network is describable by subclasses, related to its noise sections, backbone level sections, spur level sections, hops and stations. Actual sections, hops and stations are prespecified in the expert system and not determined during processing.

Once the eligible hops and stations have been determined for the MWM alarm, the diagnostic process enters the next phase. For each MWM alarm, a fault tree has been created to guide the diagnostic process. Figure 7 illustrates the fault tree for the noise outage and noise performance alarms and Appendix A contains all the fault trees for the fourteen MWM alarms that are diagnosed by the prototype. The fault trees were elicited from the Bonneville MCS alarm diagnostic experts by the knowledge engineer. The fault tree approach proved to be quite effective because the experts were able to visually inspect the representations of knowledge destined for the expert system and the fault trees fostered intellectual rigor.

Essentially, the expert system considers each potential cause of the MWM alarm independently and reports a certainty factor for each cause. Thus, with respect to a noise outage alarm (see Fig. 7), the expert system would first explore an unidirectional equipment problem. Badger transmitter, receiver, and noise differential alarms, if received for the particular hop in question, are used for support of this particular diagnosis. Next, a weather path fade problem would be explored and finally a bidirectional equipment problem would be explored. There is no backtracking in this approach. Rather, the approach is exhaustive. For each potential cause, an instantiation of an example of an object of a class of objects known as causes is created (Fig. 8).

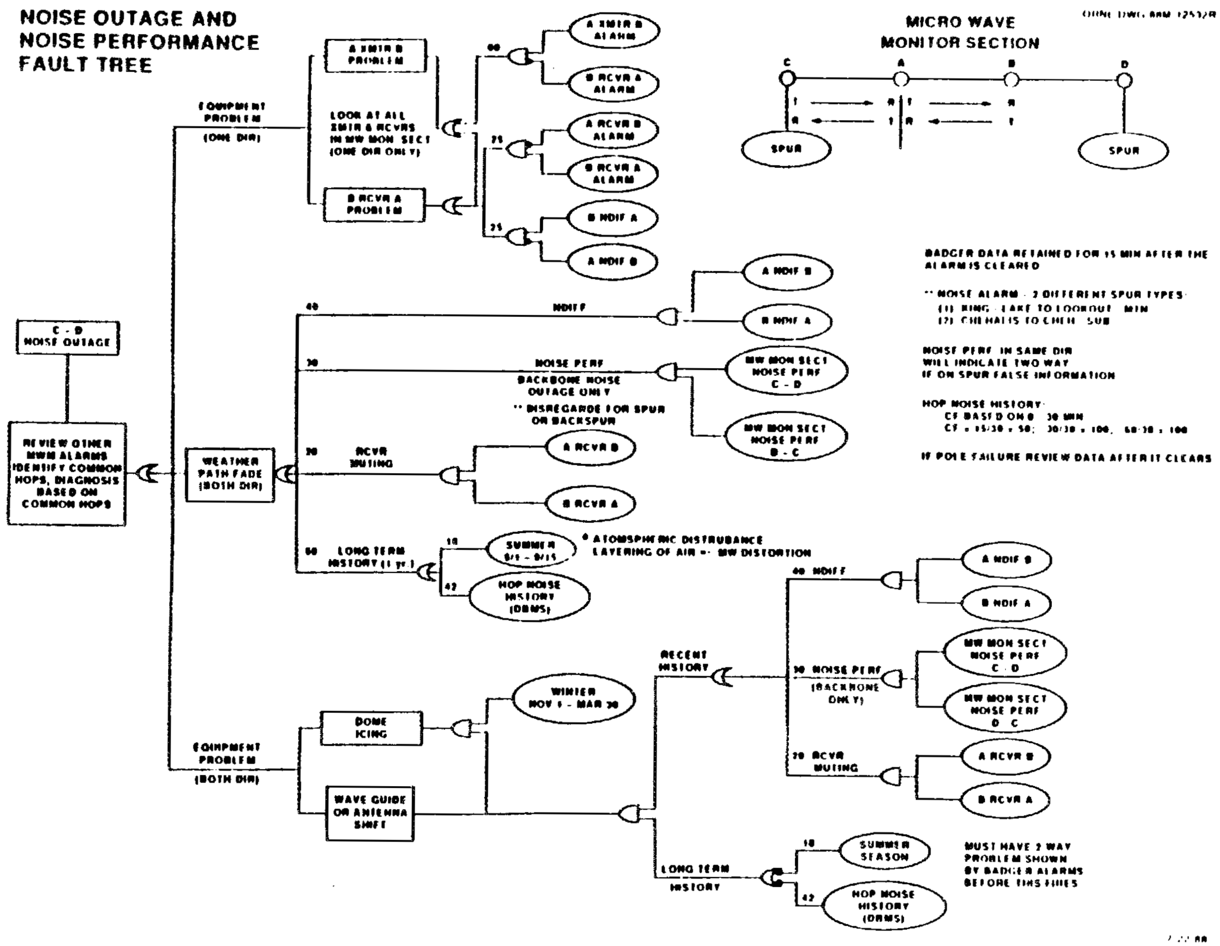


Figure 7. Noise Outage and Noise Performance Fault Tree

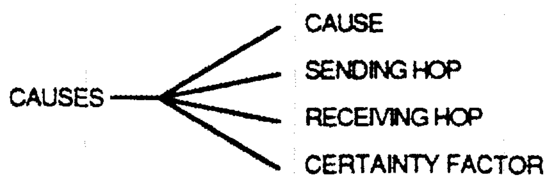


Figure 8. Microwave Monitor Alarm Cause
Knowledge Representation

For each instantiation of a cause, recorded are the cause name, the certainty factor, and the receiving and sending hop.

From Fig. 7, one can discern that small sets of Badger and MWM alarms provide partial evidence for each diagnosis. For example, from the top of the figure, if "A XMTR B" Badger alarm is true and "B RCVR A" alarm is not true, then a certainty factor of 80 would be assigned to the cause of "A XMTR B PROBLEM." Additional support for this cause would be forthcoming if "B RCVR A" were true and "A RCVR B" were not true or if "B NDIF A" were true and "A NDIF B" were not true. If certainty factors were being combined with the certainty factor rule[8], then the following rule would be used:

$$CF\ (D)= [Cf(a)/100 + (CF(b)/100)((100\text{-}CF(a))/100)]*100,$$

where CF(a) represents one certainty factor to be combined with another, CF(b), to yield the total certainty factor, CF(D), for the diagnosis. Using the example begun above, if we were to combine a CF(a) of 80 and a CF(b) of 25, the resulting CF(D) would be 85.

It was found that this rule, which is widely used in expert systems, provides satisfactory discernment between possible diagnoses but not optimal discernment. The following combination rule was tested:

$$CF\ (D)= [(CF(a)*CF(b)/100)[CF(a)*CF(b)/100 + (100\text{-}CF(a))*(100\text{-}CF(b))/100]]*100.$$

This rule assumes that the uncertainty estimates depicted in Fig. 8 are additive probabilities, whereas the certainty factor rule assumes nonadditive probabilities. Shafer[9] provides an excellent discussion about these rules and their history. Using this rule with the above example produces

CF(D)=57. Unfortunately, this rule acts to overly degrade evidence when the certainty factors are less than 50. Future research will focus on replacing certainty factors with belief functions[10].

The knowledge found in the appropriate fault diagram for each MWM alarm is applied for each eligible hop or station. The expert system accumulates diagnoses for all the hops and stations and then a subroutine of Messenger outputs to the screen, and to a file, all the potential diagnoses listed from highest to lowest certainty factor. As described in Section 5, the operator is able to review the file and designate an accepted diagnosis. The chosen diagnosis is then stored in a historical file and the unaccepted diagnoses are deleted from CAP.

In summary, the expert system operates very straightforwardly by systematically processing incoming MWM and Badger alarms, identifying the highest priority alarms for immediate diagnosis, identifying the topological parameters, and exploring in an exhaustive fashion all the potential causes for each MWM alarm for each relevant topological breakdown. The next section details the software used to prepare the input alarm messages for the expert system.

4.0 ALARM DATA PREPROCESSING

4.1 INTRODUCTION

Unlike the majority of expert systems, CAP doesn't receive data from users, but from the environment, in an asynchronous manner. Thus, a major challenge of this project was to prototype a module to transfer alarms from the environment into the expert system. Alarm data preprocessing refers to the set of real-time cooperating data processing activities that take place between the Microwave Monitor and Badger alarm systems and the expert system. The major functions performed during alarm data preprocessing are:

1. Receive data from the input ports connected to the Microwave Monitor and Badger alarm systems;
2. Maintain a history of relevent alarm activity for each input alarm system, using an on-line database, called the Input Data Buffer (IDB);
3. Generate a message for the expert system by processing each alarm message in the IDB;
4. Update the expert system's fact base by reading the expert system alarm messages generated by the IDBs; and
5. Archive from the IDB into a history file.

This section documents the alarm data preprocessing and archiving software programs and how they cooperate and communicate. Included are descriptions of the algorithms and data structures. The discussion begins with an overview of the software architecture. Then, the input alarm and data buffers are described.

4.2 ARCHITECTURE OF ALARM PREPROCESSING SOFTWARE

Figure 9 shows the architecture of the input alarm data processes, including the IDBs. Input data to the CAP alarm processor comes from existing, independent alarm systems. As discussed in Section 2, two of these alarm systems, the Badger alarm system and Bonneville's Microwave Monitor, are being used to provide data for the CAP expert system prototype.

The input alarm systems are tapped at appropriate places (i.e., logger ports) using RS-232 connections. The alarm systems generate asynchronous alarm messages which appear at the CAP serial ports (step 1, Fig. 9). The operating system, VMS, reads the characters, terminates the message at a carriage return, and buffers the messages for the IDBs (step 2).

An IDB performs the following: reads the next input alarm message (step 3), updates the IDB database (step 4), and generates the expert system message (step 5).

The IDB output messages are written to a common mailbox, a VMS message queue (step 6). At step 7, the expert system comes to a point where it is receptive to additional input data and performs a call to the Messenger. The Messenger reads the next message from the mailbox (step 8), decodes the message, and makes appropriate changes to the expert system's fact base (step 9). As a result, either a new alarm object is created in the expert system's fact base or an existing object is modified. The Messenger processes all messages in the mailbox and then returns control to the expert system.

The IDB maintains an active alarm file. An alarm is active until the alarm has been cleared for 15 minutes. At step 10, the IDB archives to a disk file the records of old alarm events.

The discussion so far has focused on the architecture of the IDBs when operating on-line. A second mode of operation developed for testing purposes uses an alarm simulator during expert system development (Fig. 10). Thus, to facilitate development, the Simulator provides the expert system programmer complete control over what input alarm data to process and when the alarms occur. The Simulator reads disk files of alarm scenarios and writes input alarm messages to mailboxes connected to the IDBs. The mailboxes look like serial ports to the IDBs, therefore, the development environment fully exercises the set of processes that make up CAP's on-line alarm processor.

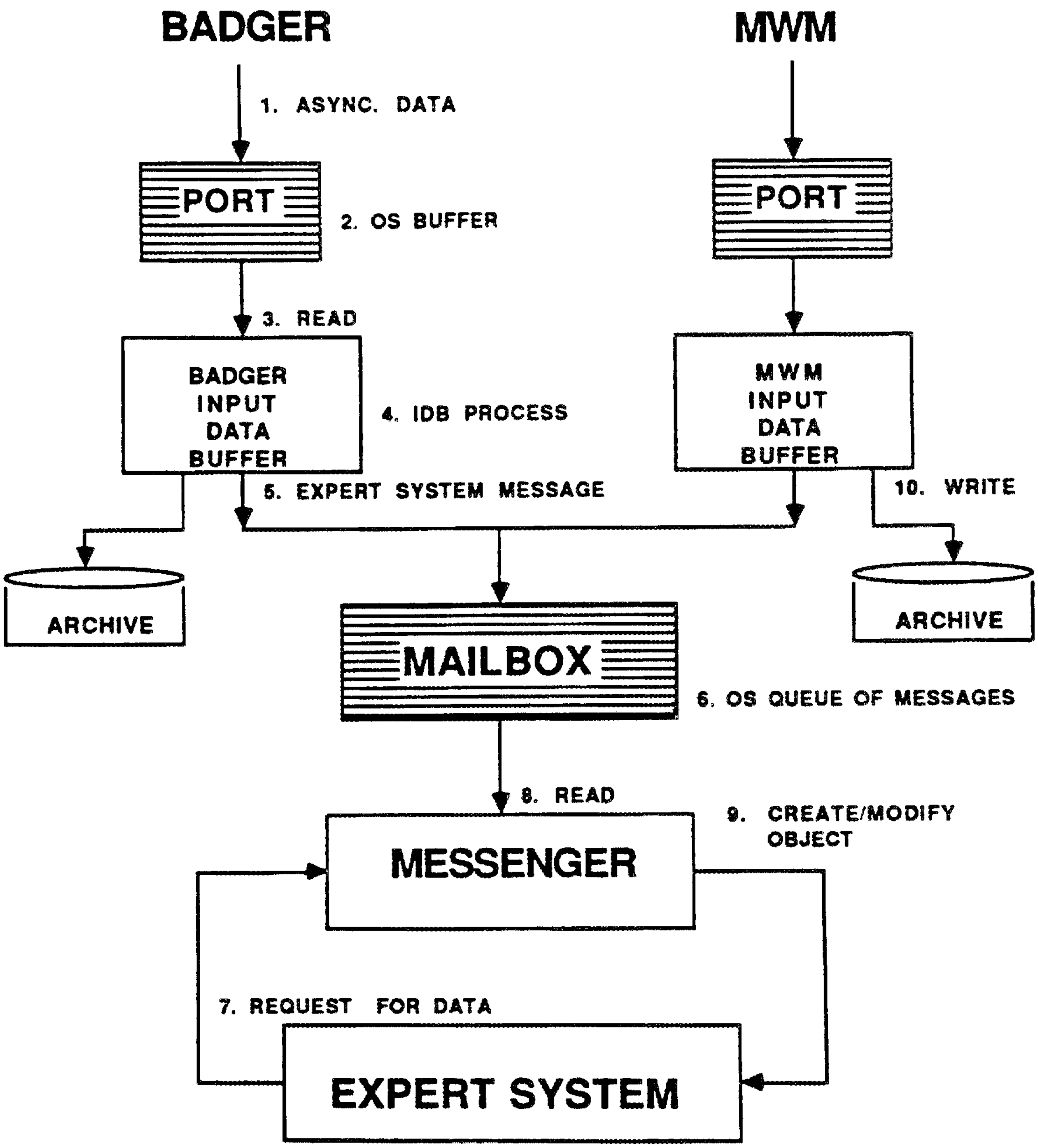


Figure 9. Architecture of the Software that Preprocesses the Input Data When Operating On-Line

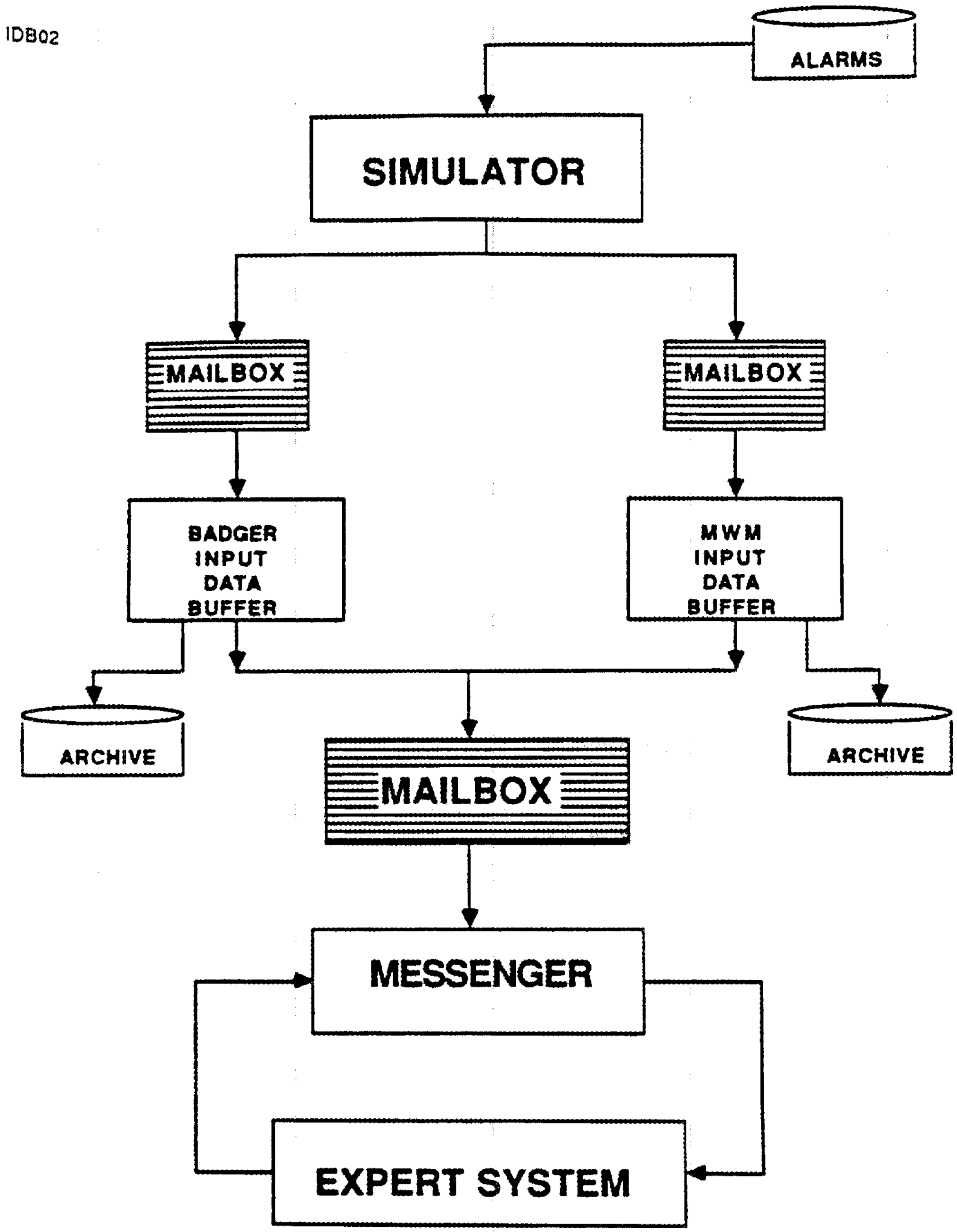


Figure 10. Architecture of the Software that Preprocesses the Input Alarm Data When Operating Off-Line

The architectural design of the IDBs has a number of advantages and few disadvantages. The architecture employs a separate IDB process for each input alarm system. Written in "C," each IDB process concentrates on servicing the serial port connected to its respective input alarm system. This design allows each IDB to be highly specialized and this helps to optimize IDB performance. In addition, the IDB design is highly modular and this provides a high degree of flexibility. For example, expansion of CAP by adding another input alarm system would be a simple task. The additional IDB would in no way interact with existing IDBs and their operation. One disadvantage of the design is that the input data are not integrated until they are delivered to the expert system. Therefore, it would be very difficult to implement an algorithmic task (e.g., in "C") that requires data from both IDBs. For example, one such task would be to instruct CAP to ignore certain MWM and Badger alarms associated with a planned maintenance activity.

4.3 INPUT ALARM SYSTEMS

A thorough understanding of the MCS alarm systems is required in order to precisely handle the alarm input process. CAP input data are provided by two existing, independent microwave alarm systems, the Badger and the MWM. Other microwave alarm data which could be incorporated in later phases of CAP development include: Bonneville's SCADA systems; transfer trip; and telemeter. An alarm condition or event causes two alarm messages: an "occur" followed later by a "clear" when the condition ends. The alarm messages provide the time interval, location, and alarmed condition.

An alarm condition may occur and clear many times in rapid succession, creating an intermittent alarm. It is possible to view an intermittent alarm as a single alarm event and to compress the alarm messages into a single IDB record and a single piece of information for the expert system.

4.3.1 The Badger Alarm System

The Badger alarm system receives data from over 140 remote microwave stations on seven different microwave systems of the microwave network. This alarm system provides information about a microwave station's performance including those related to noise differential, transmitter and receiver problems. Also included are engine generator, open door, multiplex, and station service fail alarms. To provide Badger alarm data to the CAP processor, a tap is made on an existing serial communications line between the Badger computer (PDP-11/34) and the center's central computer (RODS PDP-10). Table 2 shows the Badger serial port characteristics.

Table 2. Badger Port Characteristics

Port Characteristic	Value
baud	300
number of data bits	7
parity	even
number of stop bits	1
line terminator	<CR>

Figure 11 shows a typical Badger alarm message and labels the fields of interest. Table 3 provides a detailed description of each of the fields in a Badger alarm message. Note that the date/time stamp does not provide the date and the time has one-minute resolution. There are also other Badger alarm message formats, not shown, that are used for other purposes.

4.3.2 The Microwave Monitor Alarm System

The Microwave Monitor alarm system has a Remote Terminal Unit (RTU) at each end of the seven major systems of Bonneville's microwave network. Each RTU continuously monitors the following microwave system parameters: noise, backbone and spur pilot levels, intermodulation distortion, turn-around crosstalk, baseband load, phase jitter and baseband frequency response. Data from the 14 remotes are transmitted for final processing at the Master Terminal Unit (MTU), a PDP-11/24. During an alarm condition, a diagnosis is made of the trouble and its location. MWM outputs are directed to several locations including monitors, a printer, a floppy disk, and to terminals at field maintenance headquarters. Table 4 shows the characteristics of the MWM serial port. Figure 12 displays a typical MWM alarm and Table 5 provides a detailed map of the fields that make up a MWM alarm message. The MWM alarm messages are shorter yet significantly different from the Badger alarm messages.

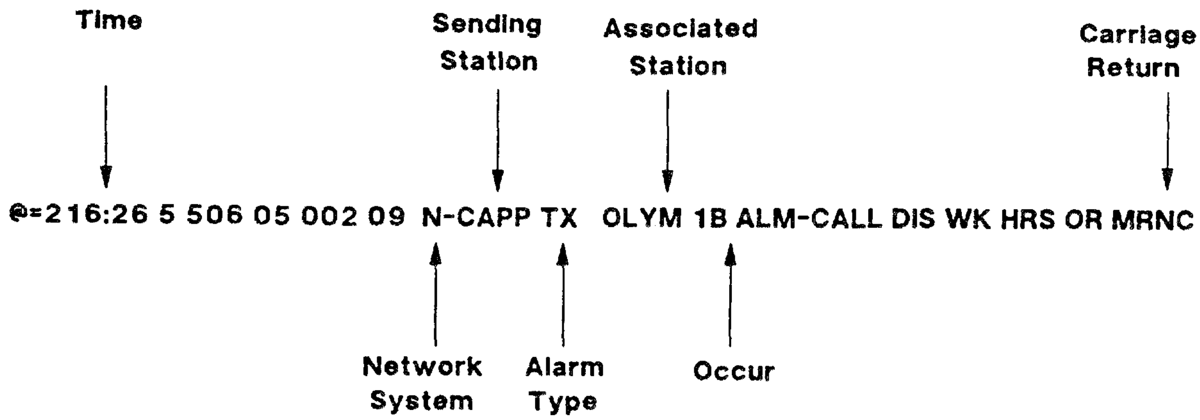


Figure 11. Typical Badger Alarm Message

Table 3. Badger Input Record Format

Col	Format*	Description
1	C1.	"@"
2	I1.	message type (31, 32, or 62 decimal)
3	+1	blank
4	C1.	reporting level (1, 2, 3, or 4)
5	+1	blank
6	HH:MM	time
11	+1	blank
12	I1.	channel number (not used)
13	+1	blank
14	I3.	station address ("1xx" for the N system)
17	+1	blank
18	I2	region (not used)
20	+1	blank
21	I3.	group (not used)
24	+1	blank
25	I2.	point number (not used)
27	+1	blank
28	C1.	system (e.g., "N")
29	C1.	"-"
30	C4.	station
34	+3	blank
38	C12.	alarm type
49	+2	blank
51	I1.	status ("1" for Occur, "0" for Clear)
52	C28.	response string

*C=Character, I=integer

Table 4. Microwave Monitor Port Characteristics

Port Characteristic	Value
baud	1200
number of data bits	7
parity	none
number of stop bits	1
line terminator	<CR>

Table 5. Microwave Monitor Alarm Message Format

Col	Format*	Description
1	MM:DD	date
6	+1	blank
7	HH:MM:SS	time
15	+1	blank
16	C1.	"N" (Microwave System)
17	+1	blank
18	C4.	RTU ("DITT" or "EACC")
22	+1	blank
23	variable	alarm category
-	+1	blank
-	variable	alarm type ("outage" or "performance")
-	+1	blank
-	C9.	location
-	+2	blank
-	I3.	alarm sequence number

*C=character, I=integer

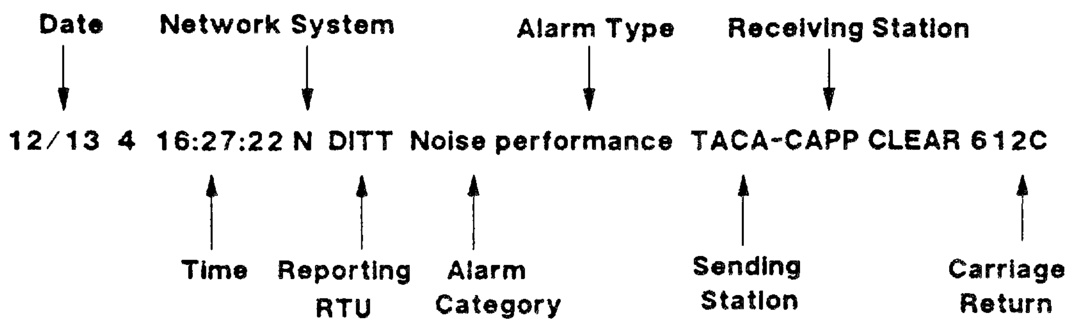


Figure 12. Typical Microwave Monitor Alarm Message.

4.4 INPUT DATA BUFFER PROCESSING

Once CAP is started, each IDB runs continuously, performing a simple loop of operations (Fig. 13). The next input alarm is read and the input is processed. In the first processing step the IDB transfers data from the input alarm message to the IDB's format (Table 6).

The alarm ID is a 12-character string that has a single character prefix ('B' for Badger and 'M' for MWM) followed by a unique sequence number generated from a 32-bit integer.

The input alarm systems use various formats for representation of the date and time of the alarm events. These formats are standardized by the IDB; the "C" format is used which is the number of seconds since midnight January 1, 1970. A 32-bit integer holds the result.

To process an input alarm message, the IDB first searches the IDB database for a "similar alarm." Two alarms are similar if they have identical alarm locations and alarm types. The state of the IDB can be one of three: a similar alarm was found but is too old to be intermittent, a similar alarm was found and is eligible to be intermittent (i.e., time-in minus time-out is less than the period of intermittency), or a similar alarm was not found. Then, the IDB considers the state of the input alarm message (i.e., occur or clear) and the state of the IDB to determine the action to be taken. The states and resulting actions are shown in Fig. 14. The actions taken update the IDB database and generate an alarm message to the expert system. The IDBs generate six types of expert system alarm messages. These are shown in Table 7.

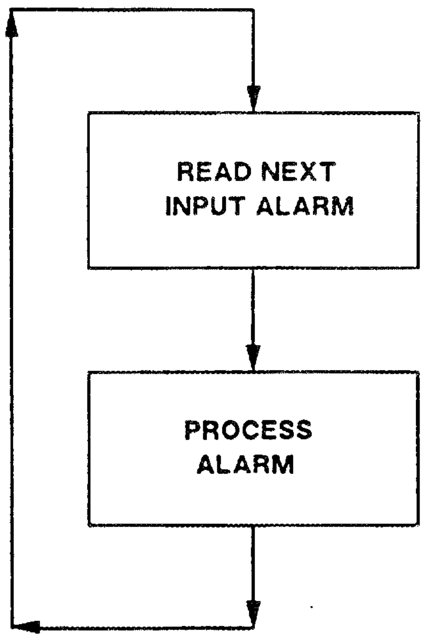


Figure 13. IDB Processing Loop for Continuous Operation

Table 6. IDB Record Format

Field	Description
ID	assigned by IDB
System	MCS Network
Time-in	number of seconds since 1/1/1970
Time-out	number of seconds since 1/1/1970
Alarm type	alarm description
Station #1	alarm location
Station #2	additional alarm location (optional)
Count	greater than 1 if intermittent
Occur	yes or no

IDB Contains A Similar Alarm	Input Alarm Message Is: Occur	Input Alarm Message Is: Clear
Yes, the Similar alarm occurred recently	• Set Similar alarm to Occur, increment occur counter • Send Recur message to the Expert System	• Update the time-out of the Similar alarm • Send Clear Message to the Expert System
Yes, but the Similar alarm is old	• Archive old alarm • Insert new alarm • Send Occur message to the Expert System	• Update the time-out of the Similar alarm • Send Clear Message to the Expert System
No	• Insert new alarm into the IDB • Send Occur message to the Expert System	• Discard input alarm message, no further action

Figure 14. IDB Actions According to the State of the IDB Database and the State of the Input Alarm Message

Table 7. Expert System Messages Generated by the IDBs

Message Code	Message Fields
Badger Occur	Message code='1' ID Time-in Alarm Text
MWM Occur	Message code='4' ID Time-in Alarm text
Clear	Message code='2' ID time-out
Recur	Message code='3' ID Count
Stop the Expert System	Message Code = '5'
Delete Alarm	Message Code = '6 ID

Other database functions that the IDBs perform include IDB initialization, record insertion, record deletion, display, record archival, garbage collection (i.e., memory management), and shutdown.

4.5 MESSENGER: THE EXPERT SYSTEM MESSAGE PREPROCESSOR

The Messenger is a "C" subroutine that processes all of the inputs and outputs of the expert system (Fig. 15). The Messenger is linked with the expert system software, therefore the Messenger and the expert system operate serially. The Messenger is called by the expert system when the expert system is ready for additional input data. Then the Messenger reads and processes the messages sent by the IDBs to the expert system's mailbox, until the mailbox is empty. Also, the expert system calls Messenger when there is a diagnosis to be reported. Messenger extracts the diagnosis information from the expert system's fact base and writes it to the screen and to a log file. Then it returns control to the expert system and to minimize interprocess communications.

The task of processing an expert system message is determined by the message code. If the message indicates an Occur, a new alarm object is created in the expert system's fact base. If the message indicates a Clear or Recur, an existing alarm object is modified. These latter messages are used to conserve memory in the expert system.

The Messenger must perform several function calls in order to insert a value into an object's slot. First, the object name and slot name are built. These are passed to the expert system, which returns a pointer to the slot. Then, the expert system slot value must be extracted from the message. Embedded in the expert system message text are the alarm location, alarm type, and a date/time stamp. Finally, the slot value is inserted into the fact base of the expert system.

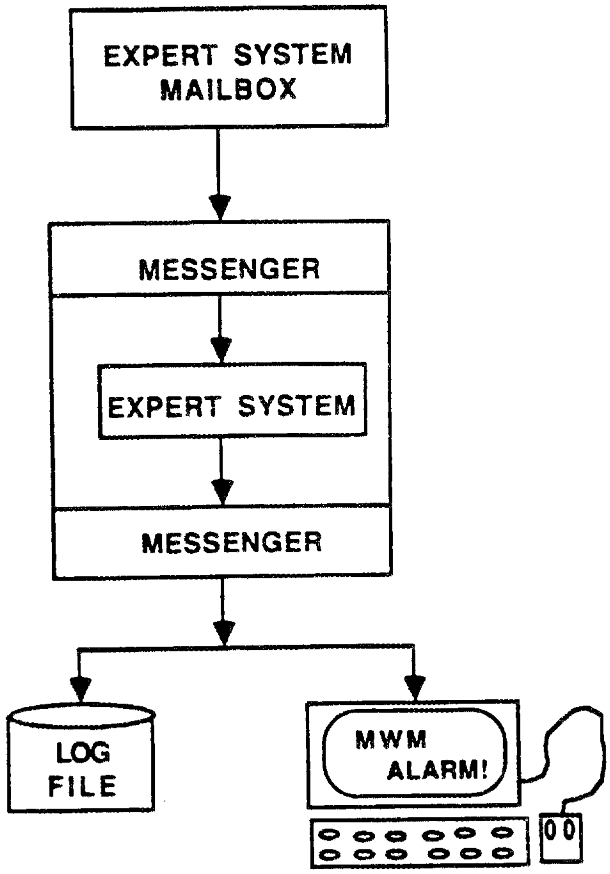


Figure 15. Expert System Message Processor

When control is returned to the expert system, its fact base has been updated to reflect current alarm activity. The expert system has full control over when new data are to be integrated with the existing fact base, which helps to coordinate inference tasks with the data used for inferencing. Also, the expert system determines when alarm data objects are no longer useful for inferencing and can be deleted from the fact base.

5.0 DESCRIPTION OF CAP STATISTICAL SOFTWARE

There is a wealth of information available in the volumes of data archived by the IDBs (i.e., raw alarm data) and the expert system (i.e., diagnosed alarm data). In addition to the real-time reports of alarm activity, which are most useful to MCS operations staff, this phase of the CAP project initiates the first steps necessary to provide longitudinal studies of the archives, which are expected to benefit MCS maintenance, planning, and design activities. Well-designed statistical analyses can measure performance characteristics of the MCS equipment including: changes in performance, locations of problems, correlations of equipment characteristics, and outliers. CAP can generate statistical reports as well as graphics. This section discusses in more detail the types of statistical analyses that may be useful to perform and the statistical package chosen for CAP.

CAP data analyses can provide information useful for both preventive and corrective maintenance. In particular, information is available in CAP files to help optimize MCS maintenance schedules. Thus, through the results of reliability analysis on equipment using archived data, it is possible that numerous more costly scheduled maintenance activities could be converted to "as needed" maintenance activities. Statistical analyses can also help Bonneville focus on maintenance problems. Currently, trial and error procedures are sometimes necessary to pinpoint the exact location of a problem. However, with CAP-supplied diagnostic information, maintenance problems should be pinpointed quicker, thus providing the potential for improving maintenance response times. Finally, with experience, it may be possible to predict MCS problems (e.g., equipment failure rates), thereby allowing Bonneville to practice less corrective maintenance and more cost efficient preventive maintenance.

In addition to maintenance, CAP data analyses can support MCS design and planning activities. Two primary resources are needed to accomplish this task. First, CAP data can support standard statistical summaries that identify equipment that is performing well or poorly. The second resource is an existing Bonneville database that contains information such as MCS equipment descriptions and maintenance activities. When these two resources are merged, it will be possible to correlate performance characteristics with equipment vendor, equipment type, and maintenance parameters. This information can aid MCS design activities by flagging suspect equipment or even prompt the redesign of the MCS to exclude certain types of equipment. Furthermore, these analyses could aid system planning activities by providing data useful to determine the timing of system upgrades.

CAP data analysis software allows the preparation of regularly scheduled reports (e.g., daily, annual) and ad-hoc queries by MCS staff. In both cases, programming the software is fairly straightforward. Through a menu, it is necessary to indicate the data to be analyzed and the analysis to be performed. Input data are fully specified by selecting the sources (e.g., Badger, expert system), time period (starting and ending date), locations (e.g., Dittmer Control Center to Capital Peak), and alarm types. Examples of potential analyses include mean values, medians, standard deviations, and frequencies. In addition to these analyses, Bonneville can explore more complicated questions. As an example, total alarm time as a percentage of the analysis interval could be calculated and analyzed. Analysis results can be presented graphically, including bar charts, pie charts, and plots.

The analysis software is written in SAS and includes a menu-driven interface. The basic SAS package offers statistical procedures more than satisfactory for any routine reporting. An advantage of choosing SAS is that additional routines can be easily added to CAP to perform more sophisticated analyses. The menu is also very straightforward to use.[2] All user inputs are verified

---

[2]The next section describes how to access SAS through CAP's user interface

for correct specification (e.g., correctly spelled MCS station names). The menu system takes large requests for analysis, submits the job for batch processing, and continues with menu processing. The batch job runs at a lower priority than the time-critical processes of CAP (e.g., expert system).

The combination of SAS on a multitasking, multiuser workstation offers the potential for numerous enhancements to the CAP. As mentioned previously SAS will allow Bonneville to conduct nonroutine, highly sophisticated data analyses. Much work is needed to determine which kinds of analyses will be most valuable to Bonneville. In the future, CAP may even become available for data analysis activities to Bonneville's district offices via modem links. In order to implement this enhancement, some thought should be given to computer security and controlled access to CAP.

6.0 DESCRIPTION OF THE CAP INTERFACES

This phase of the prototype development included a model interface design for the CAP workstation[11]. Critical design elements relevant to decision support were identified based on (a) the information processing load placed on the operators by the real-time nature of the task; (b) limitations in the background of MCS staff in artificial intelligence technology; (c) training requirements to provide an understanding of the expert system logic; (d) administrative functions associated with supervisory control and monitoring; and (e) life-cycle maintenance considerations. The interface design provides comprehensive decision support for diagnostic and troubleshooting procedures, embedded training through a simulation facility, training utilities for the knowledge engineering products, access to the statistical databases, and overall system administration support. This section of the report describes progress made in implementing the CAP interface design.

CAP was built using commercially available software and hardware that represent state-of-the-art workstation technology. The hardware is Digital Equipment Corporation's VAXstation, which includes a 19-inch, high-resolution color-graphics display and a mouse pointing device. The system software is composed of VMS, the VAX operating system, and VWS, the workstation software for graphics. Applications software includes Nexpert Object for expert system development, "C" for programming algorithmic tasks, and SAS for statistical analysis.

The top-level functions of CAP are accessed using the top-level menu of the workstation (Fig. 16). This was accomplished by expanding the basic menu facility that is provided by VWS. Therefore, it is not necessary to login to the workstation or to know a password. Also, extensive help and training facilities are provided for new users. The major functions of CAP that are provided include: the Badger and Microwave Monitor IDBs, the Expert system, the Simulator, Help

CAP Help
VWS Help

CAP Applications

Start the Badger IDB
Start the Microwave Monitor IDB
Start the Expert System
Start the Simulator
Archive/Log Interfaces

Terminals

Create new VT220 window
Create new Regis window
Create new TEK4014 window

Workstation

Print (portion of) screen
Set up the workstation
Exit this menu

Figure 16. Menu of Workstation Operations

interfaces to CAP's output data, and a CAP help facility. Menu options are selected with the mouse.

The CAP help facility provides the top-most level of system documentation. It is implemented using the VMS help facility, which organizes information in a tree structure; greater detail on a subject is obtained by descending the branch of information. The top level of CAP documentation includes information on such topics as (a) getting started, (b) help, (c) problems and solutions, (d) programmer's notes, and (e) shutting down.

There are two main modes of CAP operation: on-line and simulation. The CAP help facility describes how these operation modes are controlled. The default operation mode is on-line; to run in simulation mode the Simulator must be the first function selected. That is, the "Start the Simulator" line should be the first CAP application the user should mouse-on. Then, to get CAP running in either on-line or simulator mode, the Badger and MWM IDBs and the expert system must be started.

Each major function of CAP operates from its own terminal window. For example, Fig. 17 shows the window of the Simulator. Every CAP workstation window also has a Window Options menu (see upper left-hand portion of Fig. 17). The default size of each window will vary according to the needs of the application but the location of the window is arbitrary. The user has full control over the sizes and locations of all windows.

The CAP workstation-window menu includes an option to shrink the window to an icon. This provides a method to select and suspend the interface software of a CAP function and to maintain the appearance of the display. Fig. 18 shows the lower left corner of the CAP workstation display after four of the CAP windows have been closed and the icons placed next to the display of the CAP clock. An icon is opened with the mouse. To stop an IDB, for example, the user would mouse on the IDB's icon, and then give the shutdown command requested by the IDB menu. Each IDB maintains a menu which allows access and control of the IDB database (Fig. 19). The first three options provide displays of IDB contents. Figure 20 shows the appearance of the IDB window when real-time inputs to the IDB are being monitored.

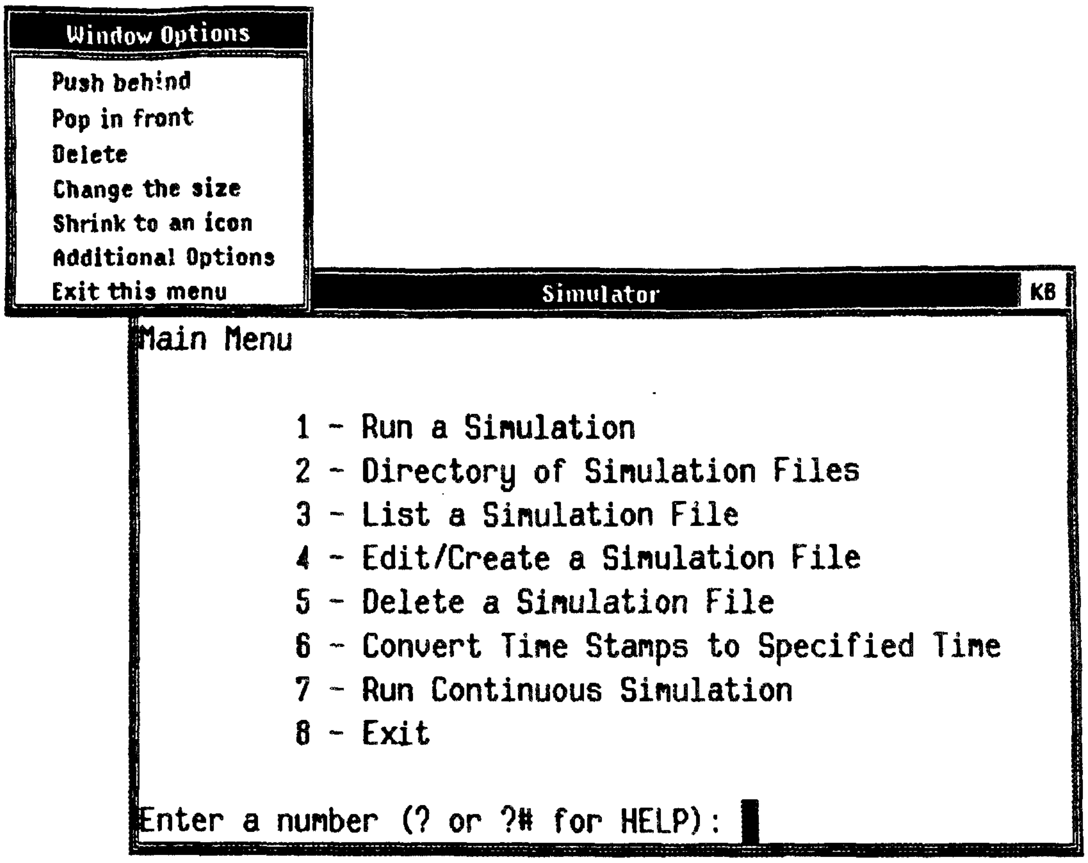


Figure 17. Simulator Menu

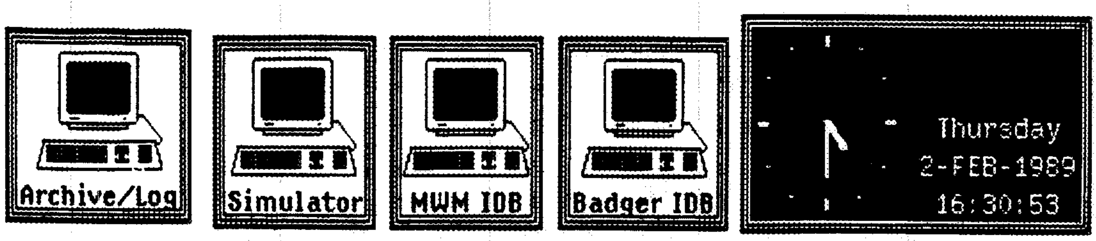


Figure 18. CAP Application Icons

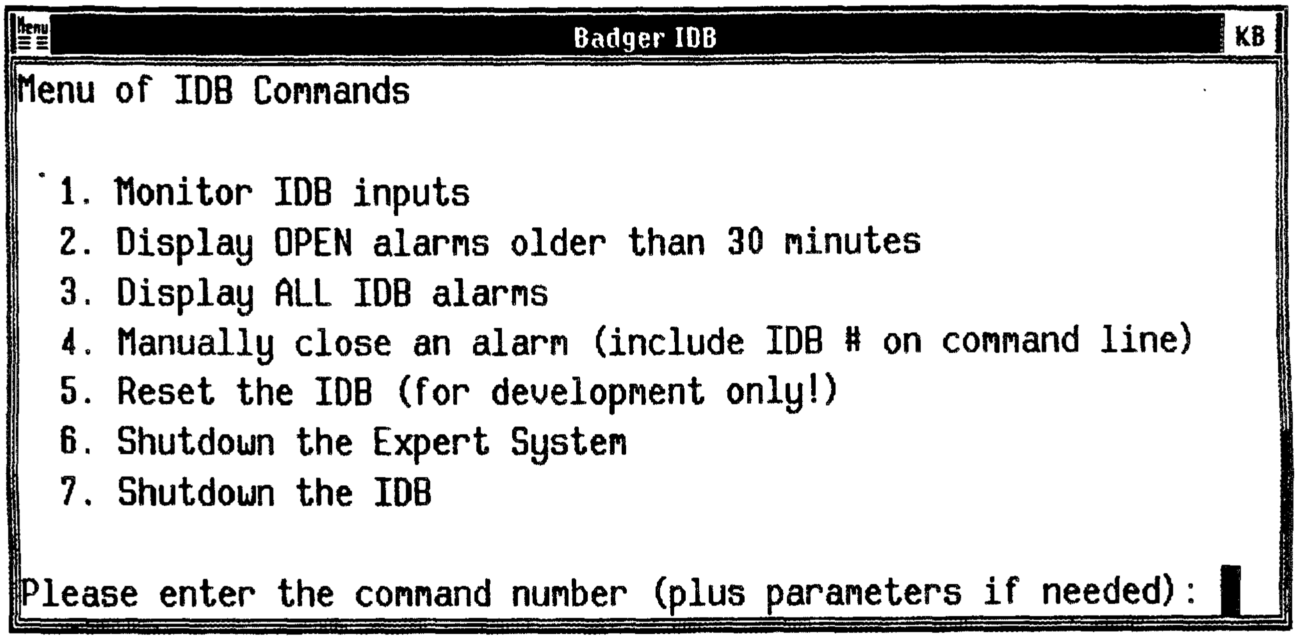


Figure 19. IDB Menu

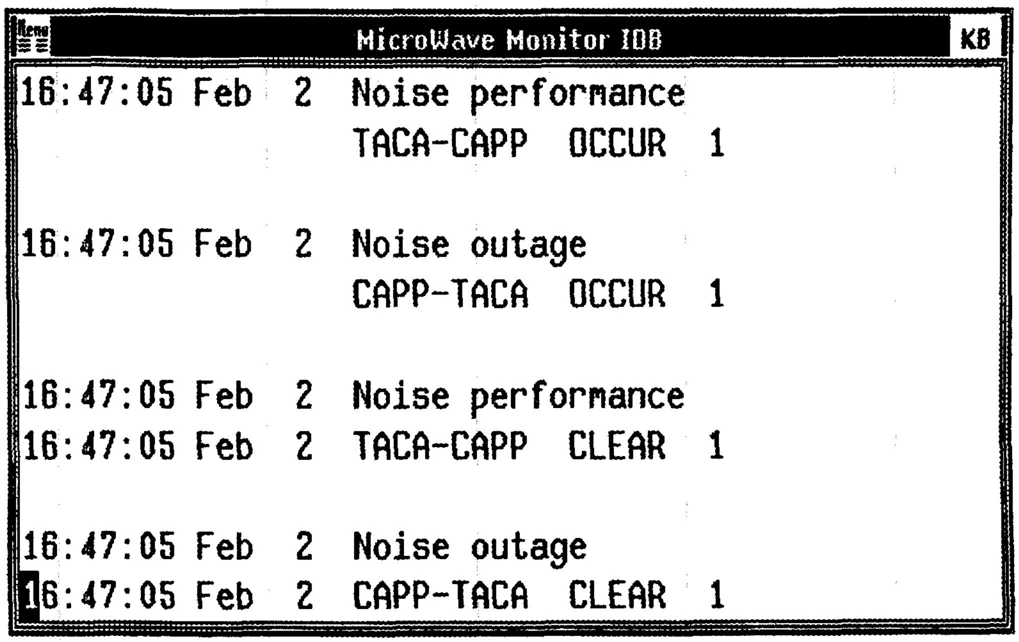


Figure 20. Report of IDB Transactions

The expert system's diagnoses are sent to two locations: (1) a real-time display in the expert system's window, and (2) a log file. The Archive/Log Interface function of CAP includes several options for browsing and maintaining the expert system's diagnoses. One of the functions presents the expert system log file to the user via an editor (Fig. 21). The editor allows the user to browse the file, to search for strings (e.g., "outage"), delete diagnoses, and to attach comments. The editor is very powerful yet easy to learn and use because the major operations are labeled on the workstation's keyboard.

The expert system's development interface provides the user access to the logic and design of the rule base, object network, and flow of control. Nexpert Object represents state-of-the-art technology for knowledge engineering. Extensive use of graphics and the mouse provide easy access to the rule and object networks. The rule syntax of Nexpert Object is very powerful and, therefore, somewhat complex. An explanation facility is available but is not currently being programmed. Figure 22 shows one of Nexpert Object's menus.

The development interface is most useful when applied in conjunction with the Simulator in the off-line mode of operation. In addition to browsing facilities, the expert system development environment provides full control of the operational parameters of the expert system. What-if scenarios can be executed.

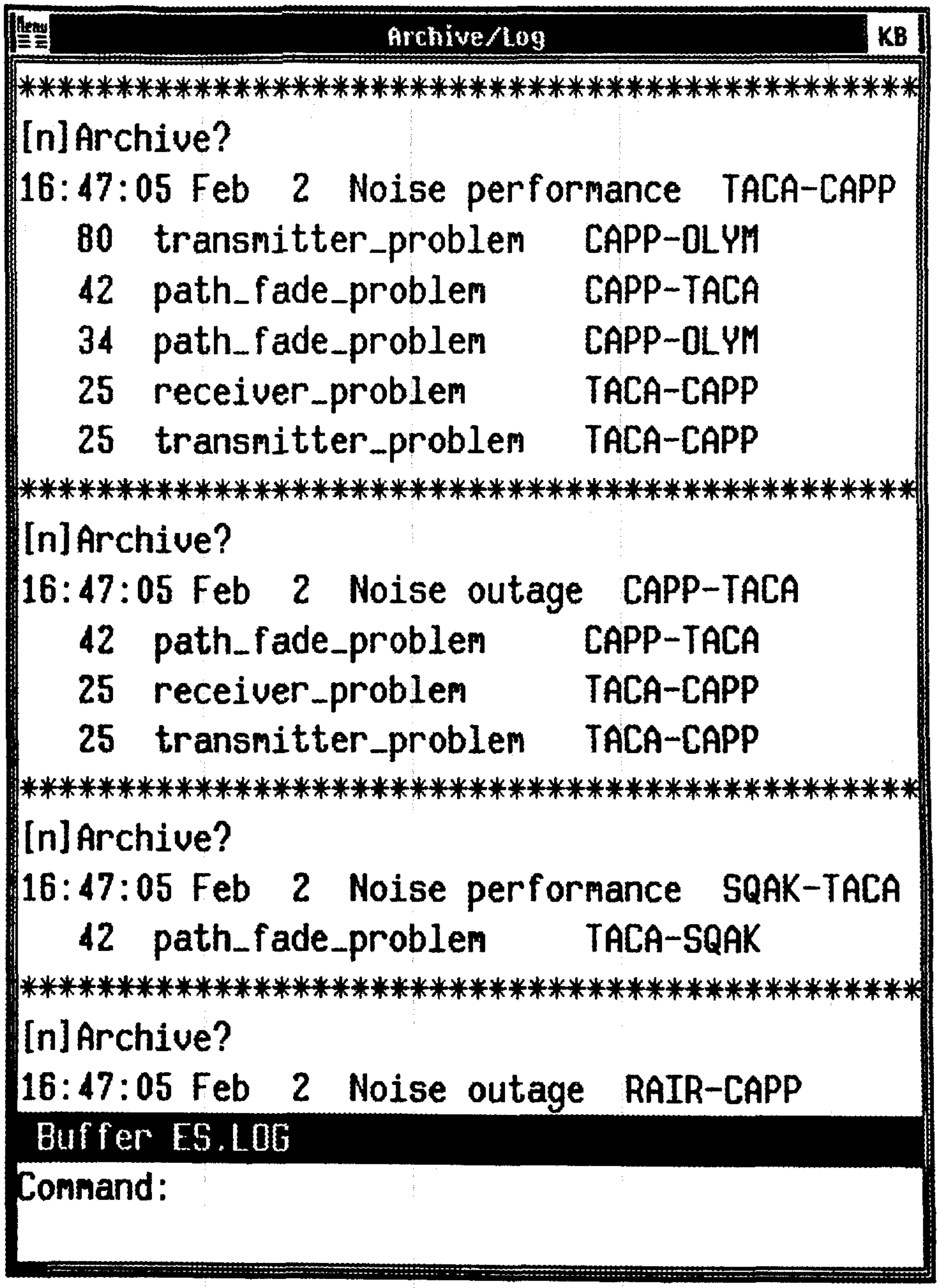


Figure 21. Expert System Log File Window

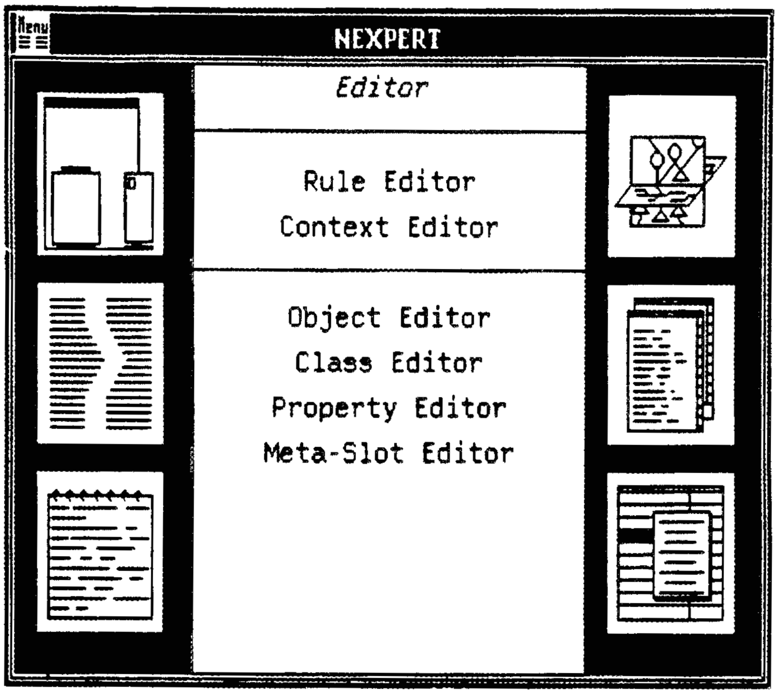


Figure 22. An Example of a Nexpert Object Menu

The SAS programs are started by clicking the mouse on "Archive/Log Interfaces," which is also menu driven. In addition, an editor for the expert system log files is provided. The SAS applications are built by the user with forms for option selection and specification. All user inputs are verified. The analysis request is submitted for batch processing by SAS. The batch jobs run at a lower priority than the on-line processes of CAP.

In summary, a comprehensive, user-friendly interface to CAP is provided; however, there are several opportunities for improvement. The CAP user has easy access to every function and data resource of CAP (Fig. 23). Most of these functions are specified with the mouse or a menu selection. Training requirements are minimized and customized help facilities are on-line. The VWS workstation software is used to provide a consistent interface to the multiple processes that make up the CAP. On the other hand, it was not possible to implement all of the features of the model interface design because the major focus of this phase of the CAP project was on development of the expert system knowledge base. For example, color graphics of the MCS network have not yet been utilized for effective display of complex and dynamic information. Also, CAP could provide easier access to the products associated with the expert system development cycle including the fault trees, simplified expressions of the rules, schematic diagrams showing information flow through the system, and elementary expressions of uncertainty. Finally, a high performance database management system, that is fully integrated such that both on-line and off-line operations are supported, would provide a foundation for more powerful presentation, simulation, and training facilities.

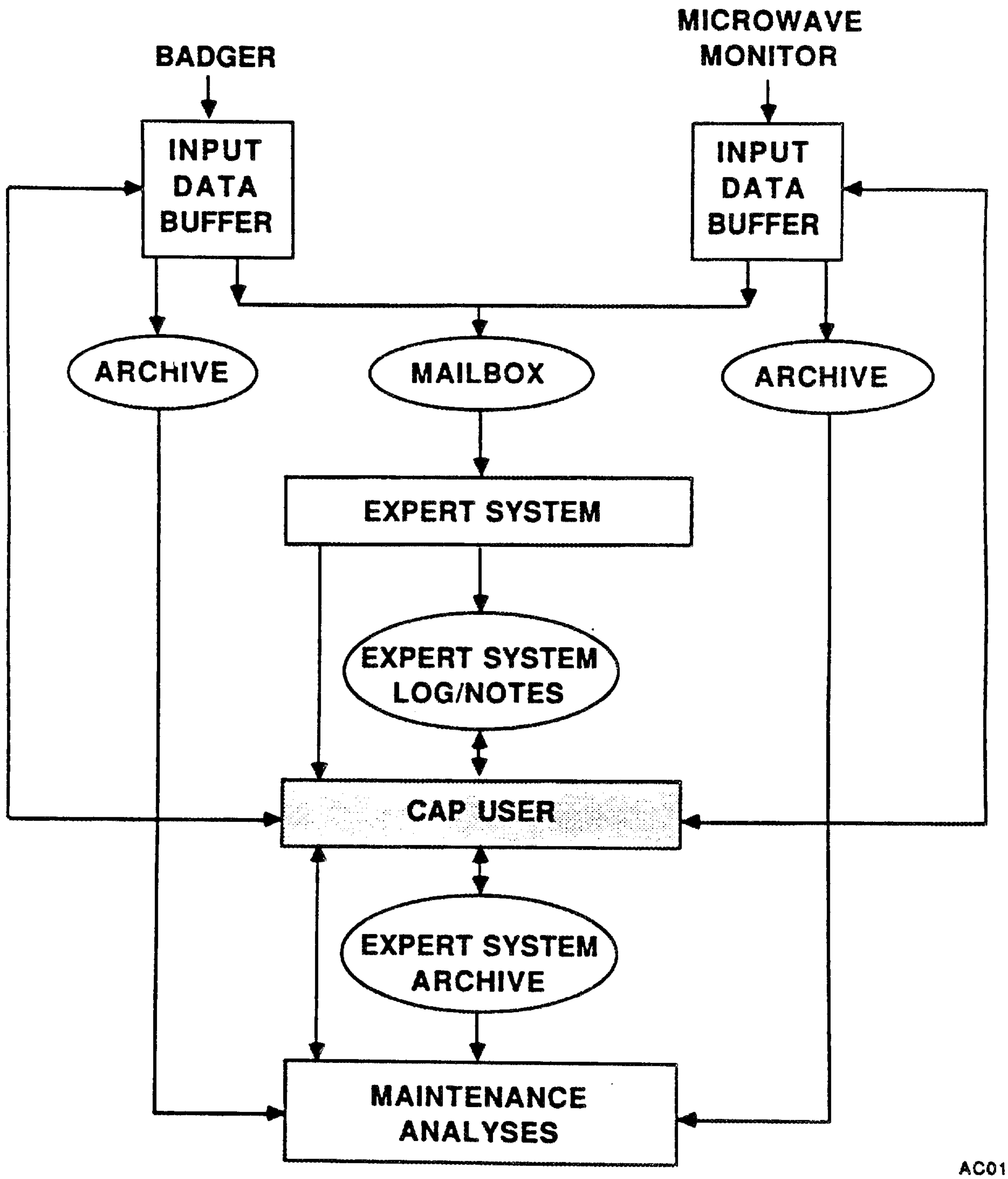


Figure 23. CAP User Interfaces

7.0 DISCUSSION AND RECOMMENDATIONS

This report describes the first phase of development of the Communications Alarm Processor. CAP has four major elements, the expert system module, the alarm input routines, the statistical software, and the interface screens and windows. As a system, CAP is highly unique in being able to coordinate and utilize custom and commercially available code. CAP also demonstrates the feasibility of using artificial intelligence technology to process alarms from power system operations. The CAP project team is anxiously awaiting results of CAP's initial field testing. The team feels confident in the product because of their quality assurance procedures.

A somewhat novel arrangement used during the coding phase of the Nexpert Object portion of the CAP project allowed for the complete separation of the coding and testing tasks for this module. Each portion of the Nexpert Object code was independently evaluated (i.e., desk checked or "eyeballed") by a second individual who was not involved in the original coding task. As a result, each rule, each object, each class, etc., in the Nexpert Object module was initially viewed with suspicion until it made sense to the independent reviewer. A number of errors were detected in this manner without actually running the system. Error descriptions were passed back to the knowledge engineer who then proceeded to modify the code without any advice from the reviewer.

After passing the reviewer's initial desk checking step, each of the fault tree implementations and the high level flow of control were further checked by running against selected variations of alarm data. A number of additional errors were found this way. All together, the test runs executed each branch of each fault tree at least once and tested for certain common types of mistakes (like failure to reset hypotheses).

The test runs should be viewed as "spot checking" only. Each set of alarm data for these tests was constructed on an ad hoc basis, with guidance provided by (1) chunks of code which looked suspicious to the reviewer and (2) insight gained from problems discovered in earlier test runs. Each test run examined one fault tree or group of related fault trees for continued correct diagnosis of the associated Microwave Monitor alarms. When errors were detected, the same tests were rerun after coding changes were made to ensure that these errors had been successfully eliminated.

Numerous challenges remain as CAP evolves into a full-scale production system. One challenge involves developing methods to update certainty factors using historical data. Because equipment is constantly being repaired and even replaced, the operational parameters of the MCS will change over time. Historical data will augment initial expert elicitations of certainty factors to ensure that the expert system module is up-to-date. This task will require advances in methods of updating current certainty factors with past data. Possibly, certainty factors may have to be replaced with a more robust representation of uncertainty, such as belief functions[9].

CAP will need to be expanded to encompass additional alarm systems, additional branches of the MCS, and additional alarm types from the Microwave Monitor system. Such expansion will warrant close attention to improving CAP's knowledge and topological representations in order to capitalize on commonalities among systems, branches and alarm types. This task may also require the solution of a difficult problem related to alarm diagnosis attribution. Currently, CAP has no rigorous method to track which alarms are used for which diagnoses. It is possible that a Badger alarm, for example, may be more useful to diagnose one MWM alarm than another. This is not a particular problem in the limited world of Phase I CAP, but could become an important problem in more complicated versions.

A third tasking area concerns the user interface. Work done by the project team on an advanced architecture for the interface[11] indicates how the current application could benefit from additional windows and powerful interfaces to data located in the expert system and the various data files. The interface may also be useful in providing training in the use of CAP and in the operation of the MCS. The workstation environment is extremely valuable for this avenue of research.

Lastly, this work can serve as a beginning for other tasks and projects. For example, the current CAP architecture could be compared to other architectures on the same hardware for improvements in processing speed. Along these lines, the current architecture could be implemented on different hardware, such as a parallel processor. Other applications might involve some aspects of power system control or planning.

8.0 ACKNOWLEDGEMENTS

The members of the ORNL project team wish to thank Ken Hemmelman, Stan Borys and Bob Rasmussen for their project management assistance and expertise in microwave communication systems. Jim Graffy and Vickie Van Zandt provided assistance in specification and installation of the computer hardware and software. C. Ed Ford and John Stovall reviewed the draft report and helped with numerous suggestions.

9.0 REFERENCES

1. Bonneville Power Administration, *William A. Dittmer Control Center: A Technical Description*, Portland, Oregon, 1981.

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

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

4. Electric Power Research Institute, *Artificial Intelligence Technologies for Power System Operations*, EL-4323, Palo Alto, Calif., 1986.

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

6. SAS Institute, Inc., *SAS User's Guide: Basics*, 1985.

7. T.J. Laffey, et al. "Real-Time Knowledge Based Systems," *AI Magazine*, 9(1), 27-45 (1988).

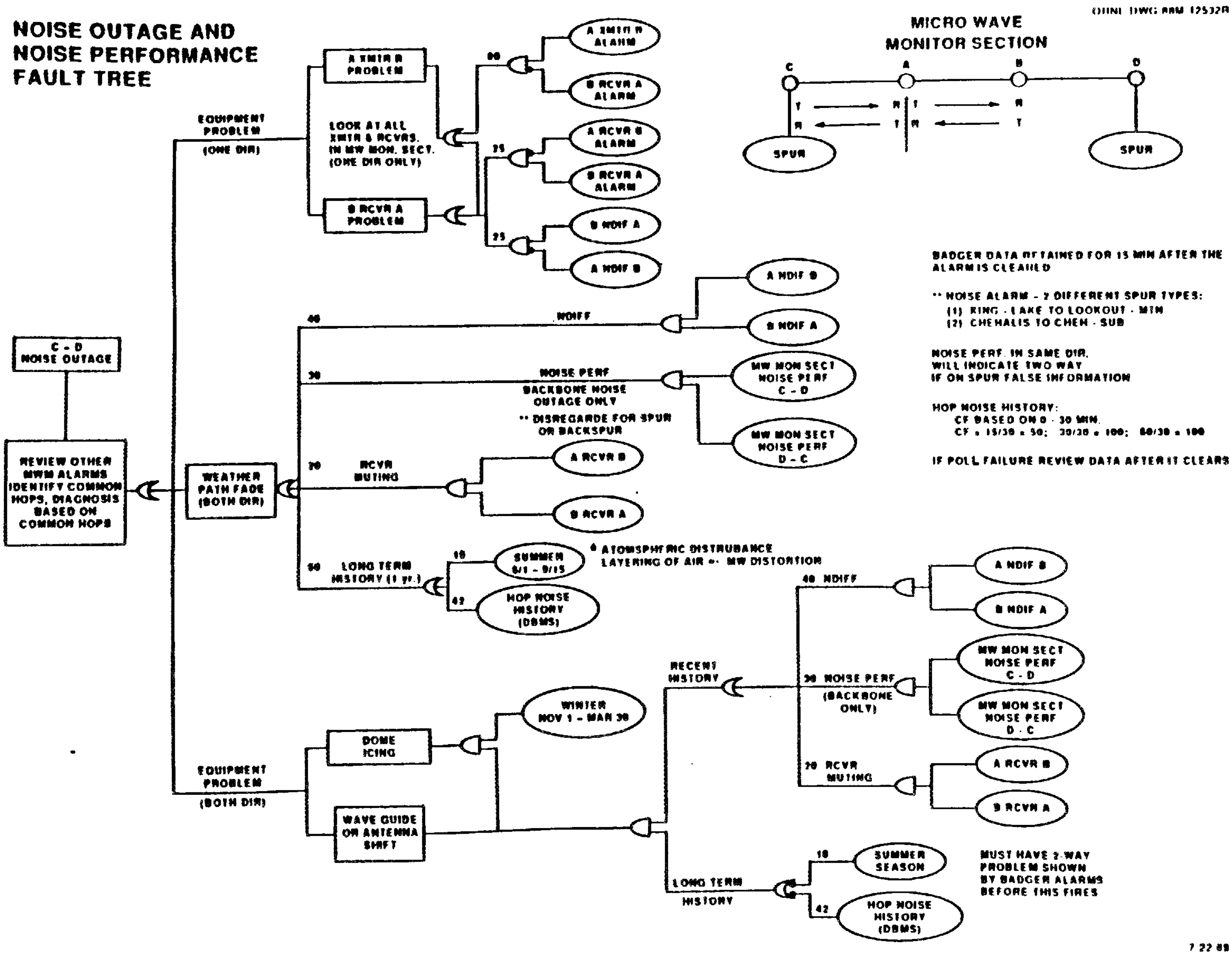
8. B. Buchanan and E. Shortliffe, *Rule Based Expert Systems*, Addison-Wesley, Reading, Mass., 1984.

9. G. Shafer, "Non-additive Probabilities in the Work of Bernoulli and Lambert," *Archive for History of Exact Sciences*, 19(4), 309-370 (1978).

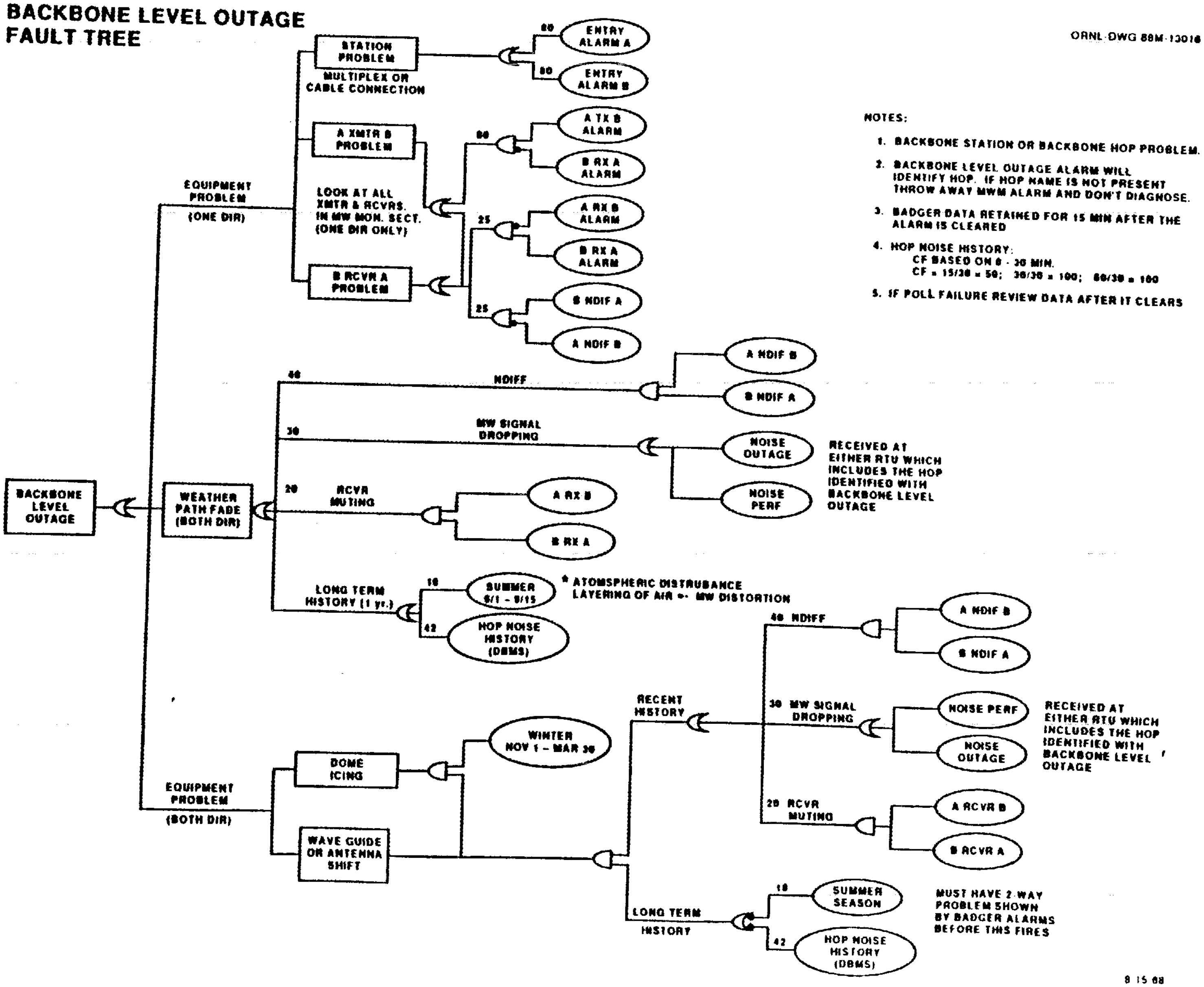
10. G. Shafer, *A Mathematical Theory of Belief Functions*, Princeton University Press, Princeton, New Jersey, 1976.

11. D. MacGregor and W. Valenta, "Modeling a User Workstation for a Real-time Expert System: An Alarm Processor Case Study," Eugene, Oregon, 1988.

APPENDIX A. FAULT TREES

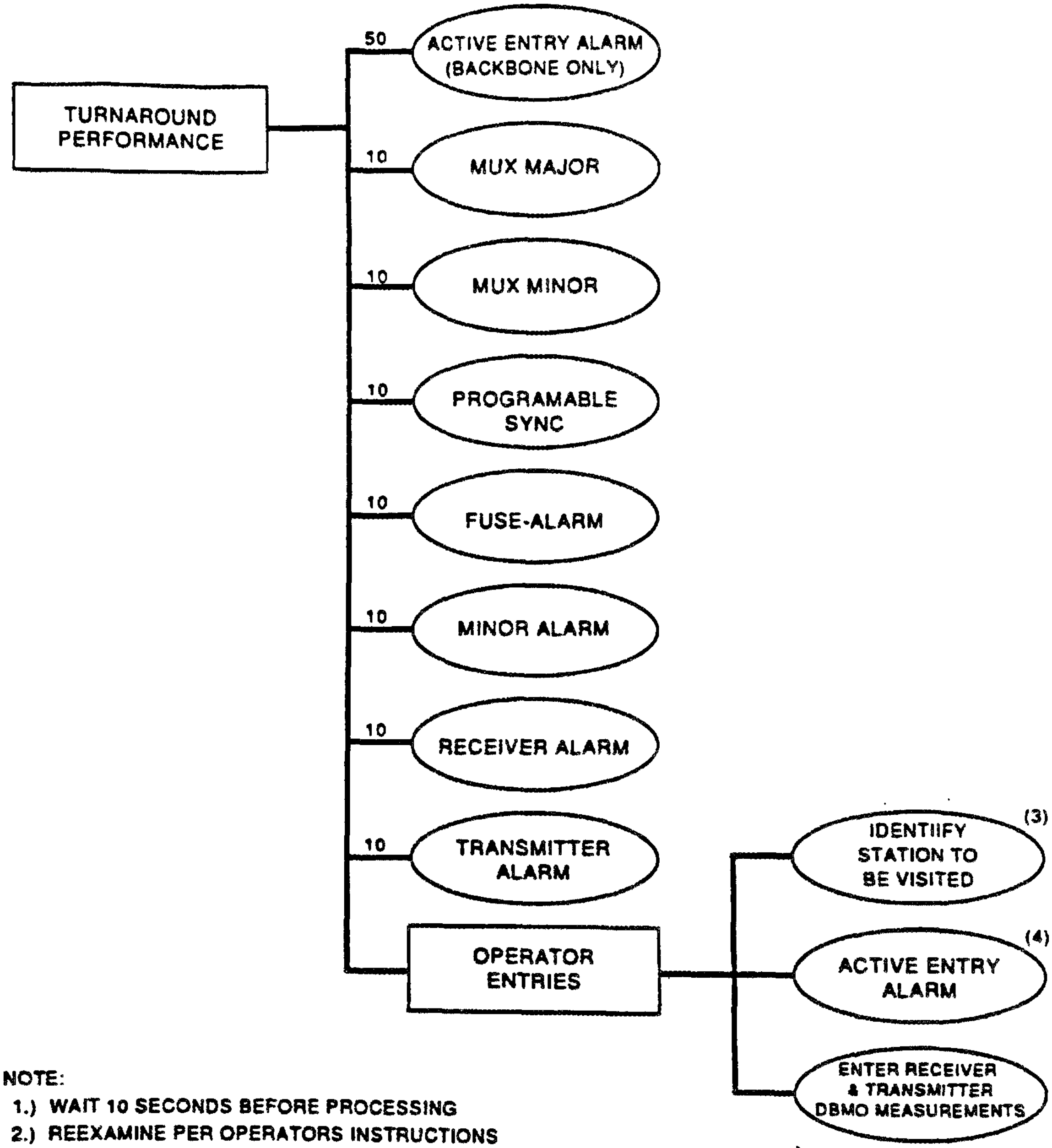

NOISE OUTAGE AND
NOISE PERFORMANCE
FAULT TREE
C - D
NOISE OUTAGE
REVIEW OTHER MWM ALARMS IDENTIFY COMMON HOPS, DIAGNOSIS BASED ON COMMON HOPS
EQUIPMENT PROBLEM (ONE DIR)
A XMTR B PROBLEM
LOOK AT ALL XMTR & RCVRS. IN MW MON. SECT. (ONE DIR ONLY)
B RCVR A PROBLEM
A XMTR B ALARM
B RCVR A ALARM
A RCVR B ALARM
B RCVR A ALARM
B NDIF A
A NDIF B
WEATHER PATH FADE (BOTH DIR)
NDIFF
A NDIF B
B NDIF A
NOISE PERF
BACKBONE NOISE OUTAGE ONLY
** DISREGARDE FOR SPUR OR BACKSPUR
MW MON SECT NOISE PERF C - D
MW MON SECT NOISE PERF D - C
RCVR MUTING
A RCVR B
B RCVR A
LONG TERM HISTORY (1 yr.)
SUMMER 6/1 - 9/15
HOP NOISE HISTORY (DBMS)
* ATOMSPHERIC DISTRUBANCE LAYERING OF AIR => MW DISTORTION
EQUIPMENT PROBLEM (BOTH DIR)
DOME ICING
WINTER NOV 1 - MAR 30
WAVE GUIDE OR ANTENNA SHIFT
RECENT HISTORY
NDIFF
A NDIF B
B NDIF A
NOISE PERF (BACKBONE ONLY)
MW MON SECT NOISE PERF C - D
MW MON SECT NOISE PERF D - C
RCVR MUTING
A RCVR B
B RCVR A
LONG TERM HISTORY
SUMMER SEASON
HOP NOISE HISTORY (DBMS)
MUST HAVE 2-WAY PROBLEM SHOWN BY BADGER ALARMS BEFORE THIS FIRES
MICRO WAVE
MONITOR SECTION
C
A
B
D
T
R
SPUR
SPUR
BADGER DATA RETAINED FOR 15 MIN AFTER THE ALARM IS CLEARED
** NOISE ALARM - 2 DIFFERENT SPUR TYPES:
(1) KING - LAKE TO LOOKOUT - MTN
(2) CHEHALIS TO CHEH - SUB
NOISE PERF. IN SAME DIR.
WILL INDICATE TWO WAY
IF ON SPUR FALSE INFORMATION
HOP NOISE HISTORY:
CF BASED ON 0 - 30 MIN.
IF POLL FAILURE REVIEW DATA AFTER IT CLEARS

BACKBONE LEVEL OUTAGE FAULT TREE



ORNL-DWG 88M-7150R2

TURNAROUND PERFORMANCE AND OUTAGE



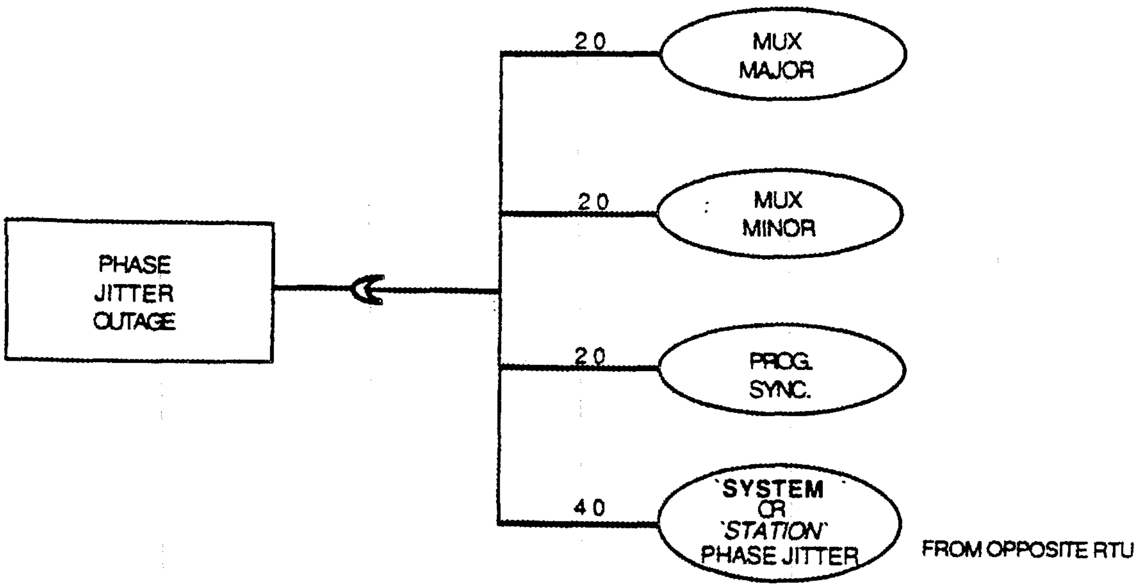
NOTE:

1.) WAIT 10 SECONDS BEFORE PROCESSING
2.) REEXAMINE PER OPERATORS INSTRUCTIONS
3.) EXPERT SYSTEM WILL IDENTIFY ELIGIBLE STATIONS AND PROVIDE A PICTURE OF THE SYSTEM
4.) IF THERE IS AN ACTIVE ENTRY ALARM FROM AN ELIGIBLE STATION EXPERT SYSTEM WILL NOTIFY THE OPERATOR SO FIELD MEASURES CAN BE MADE
5.) STATION SPECIFIC PROBLEM, NOT A PATH PROBLEM
6.) OPERATION INTERFACE TO INCLUDE OPERATOR LOG AND ACCESS TO PREVIOUS DIAGNOSIS

7-22-88

PHASE JITTER OUTAGE `SYSTEM` OR `STATION`

K-PJO-8-9-88



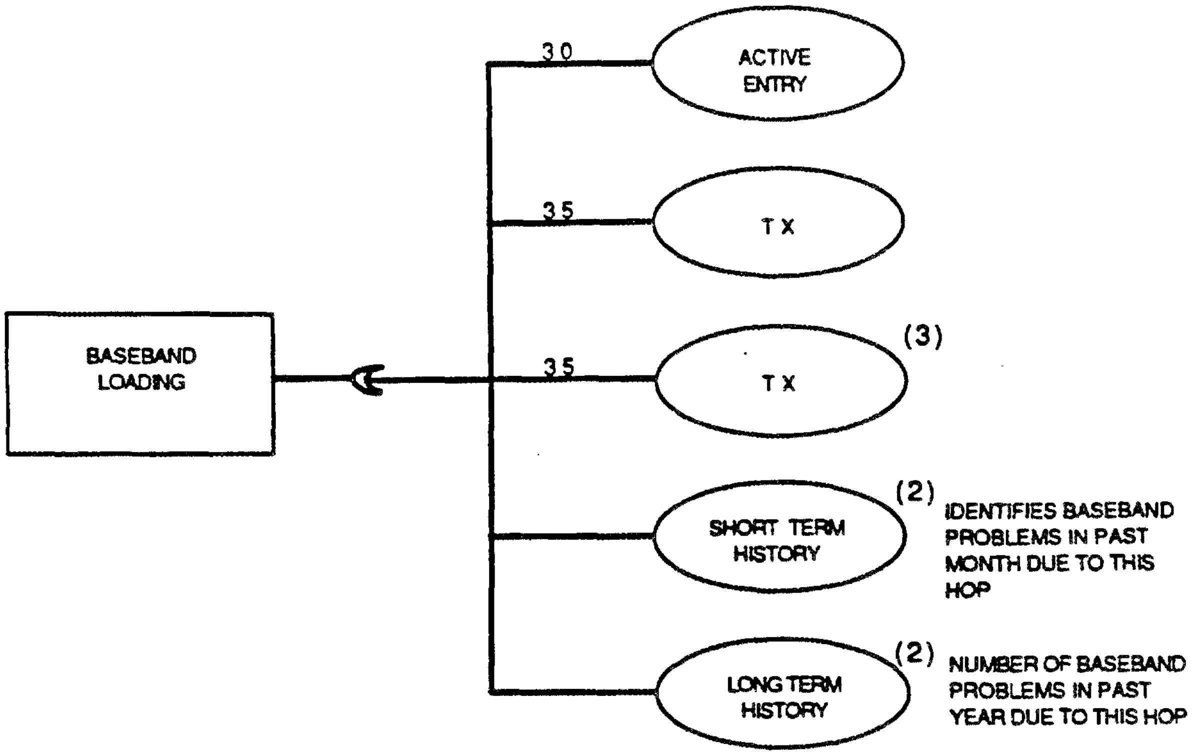
NOTES:

1. THESE ALARMS ARE CAUSED BY STATION PROBLEMS IT IS NOT A HOP PROBLEM.
2. THIS ALARMS CAN EITHER BE A `STATION` OR `**SYSTEM**` ALARM. THE `**SYSTEM**` ALARM IS PRESENT WHEN THE MWM CANNOT IDENTIFY SPECIFIC STATION CAUSING THE PHASE JITTER.
3. FOR `**SYSTEM**` OUTAGE ONLY-LOOK- AT DITT ONLY WHICH IS THE (SOURCE OF SYCN. PILOT) OTHER STATIONS WILL HAVE ALARMS, BUT FOR `**SYSTEM**` OUTAGE THESE ARE FALSE ALARMS.

7-22-88

K-BBL-8-8-88

BASEBAND LOADING OUTAGE AND BASEBAND LOADING PERFORMANCE



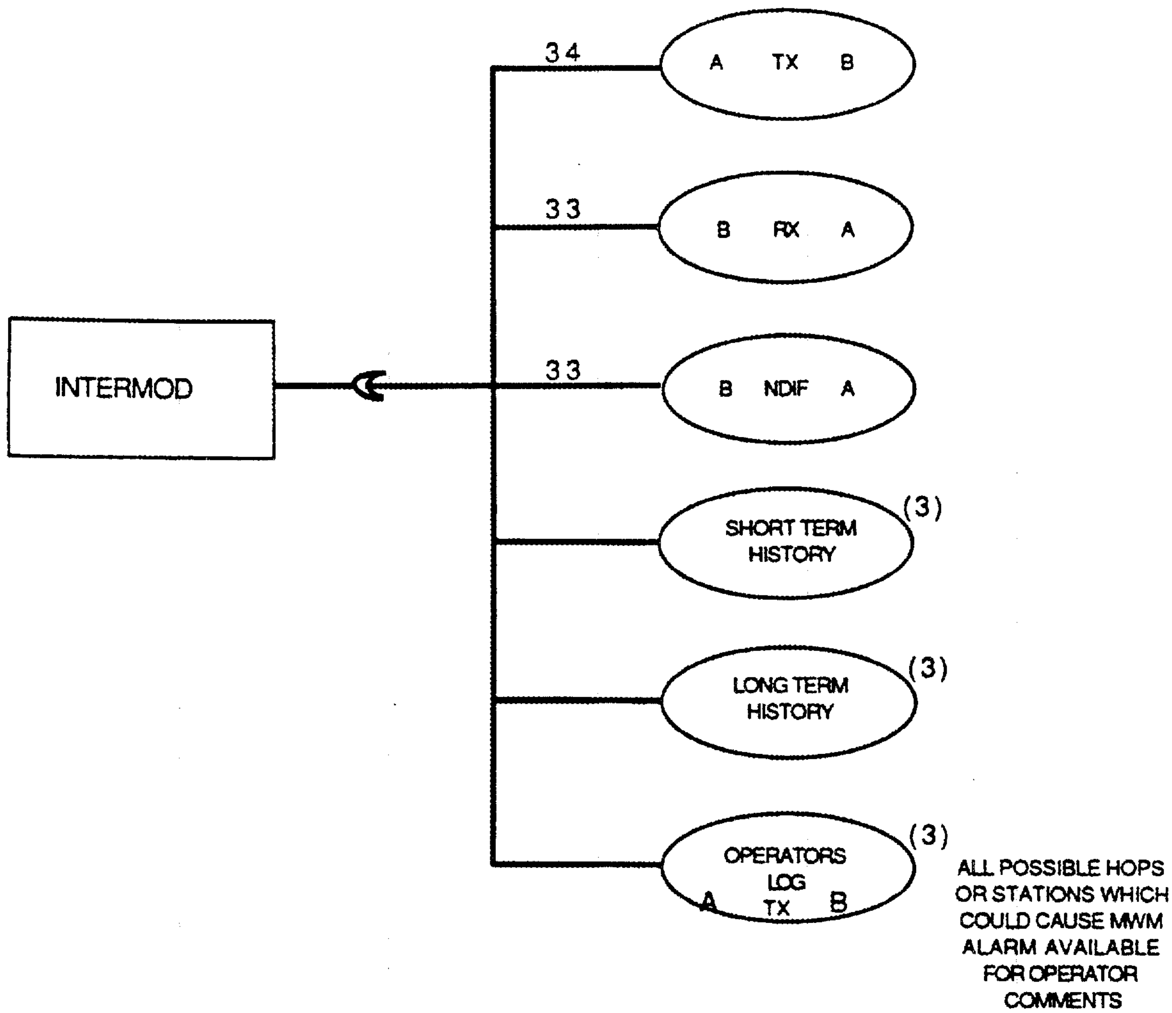
NOTES:

1. SYSTEM PROBLEM- ALL STATIONS ELGIBLE
2. HISTORY INFORMATION AVAILBLE THROOUGH OPERATION INTERFACE.
3. TOPOLOGY CONSIDERATION-TX ALARM FURTHEREST AWAY FROM RECEIVING RTU (CHAINS ALONG BACKBONE)
4. LONG TERM ALARM ~ 1 ~~SAY~~ DAY TO RESOLVE

7-22-88

Intermod Outage
Intermod Performance

K-IM-8-8-88



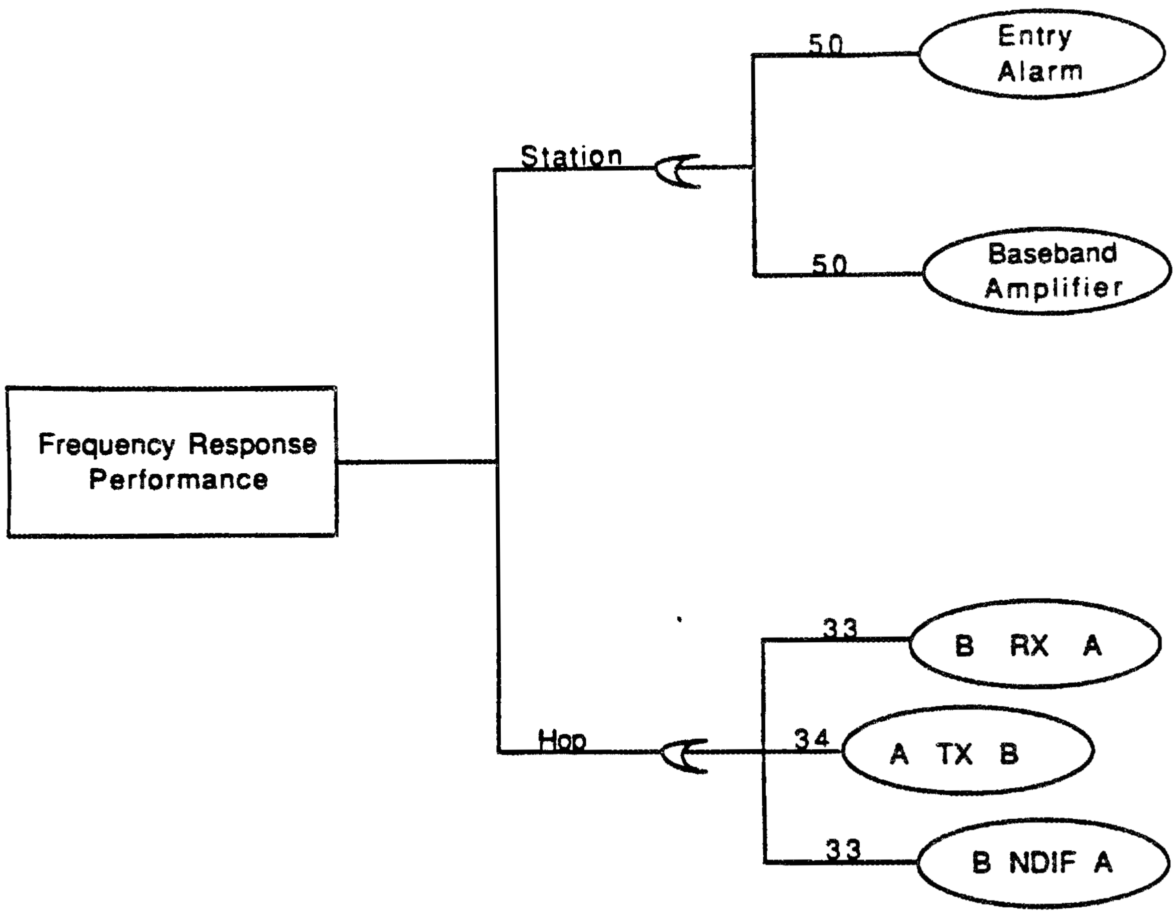
Notes:

1. THIS IS A SYSTEM PROBLEM AND IS ATTRIBUTLE TO BACKBONE HOPS ONLY
2. THIS PROBLEM IS DIRECTION SENSITIVE AND IS DETERMINED BY RECEIVING RTU
3. HISTORY INFORMATION AND LOG AVAILABLE THROUGH OPERATORS INTERFACE

7-22-88

Frequency Response Performance:

K-FRP-88



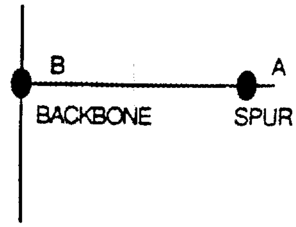
NOTES:

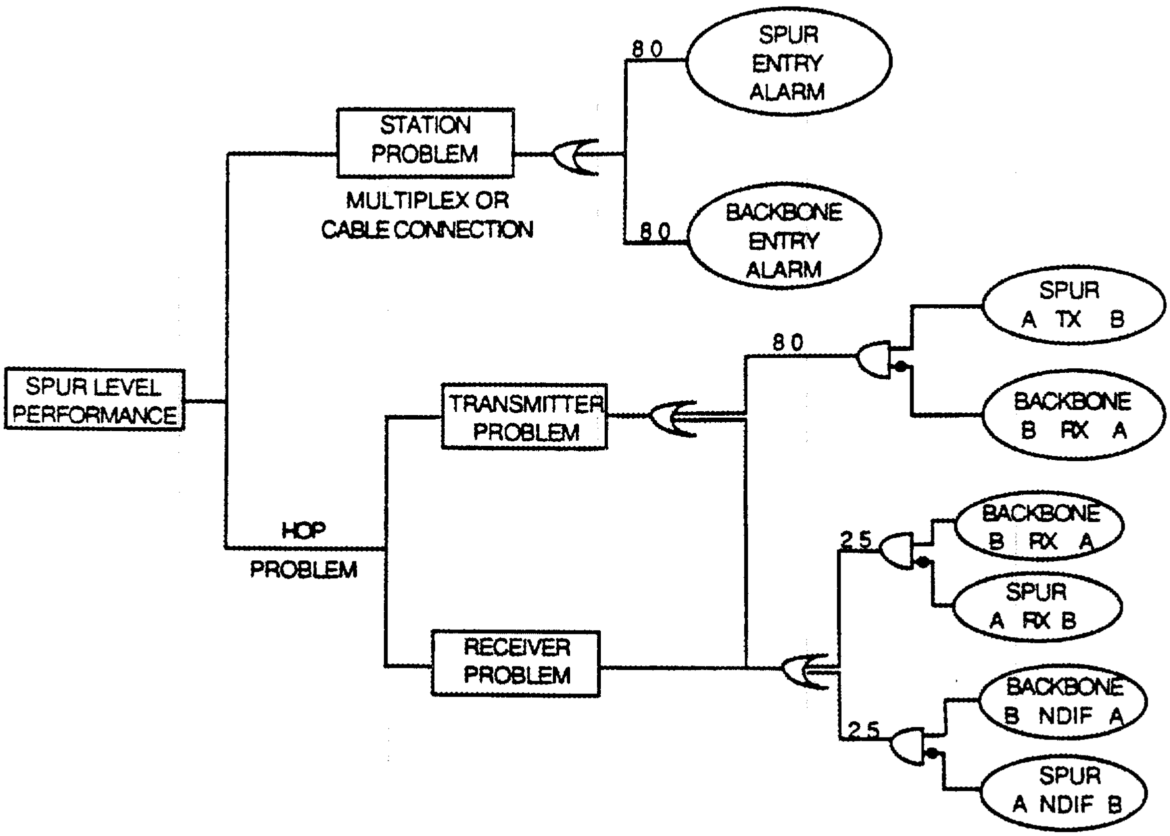
1. Backbone Station or Backbone Hop problem
2. Frequency responce is a system problem so each Backbone entity must be considered.

7-22-88

SPUR LEVEL PERFORMANCE

K-SPUR-8-88



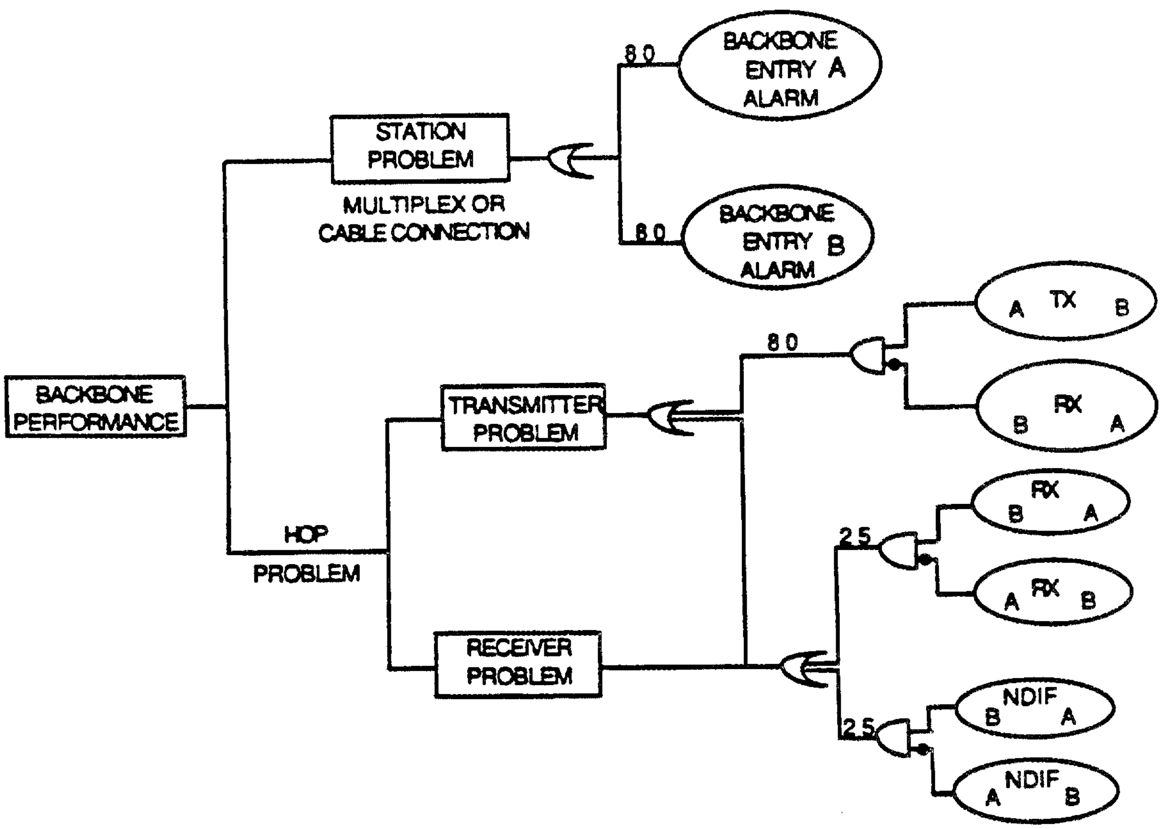


NOTES:

1. THIS ALARM IS SPUR STATION OR SPUR HOP (SPUR TO BACKBONE) PROBLEM.
2. SPUR LEVEL PERFORMANCE ALARM WILL IDENTIFY HOP. IF HOP NAME IS NOT PRESENT THROW AWAY MWM ALARM AND DON'T DIAGNOSE.

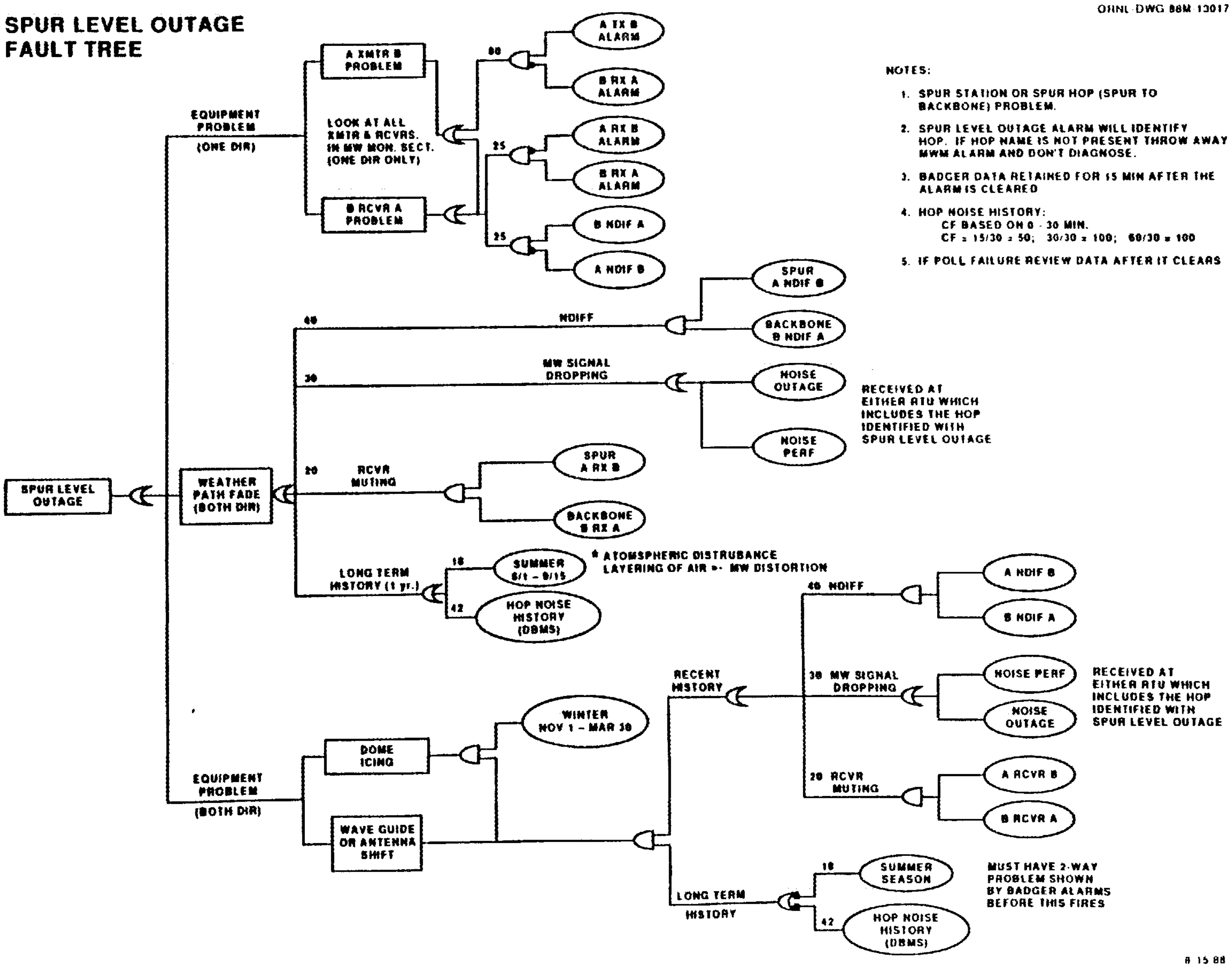
7-22-88

BACKBONE LEVEL PERFORMANCE K-BBLP-10-88



NOTES;
1. THIS IS A BACKBONE STATION OR BACKBONE HOP PROBLEM.
2. BACKBONE LEVEL PERFORMANCE ALARM WILL IDENTIFY HOP. IF HOP NAME IS NOT PRESENT THROW AWAY MWM ALARM AND DON'T DIAGNOSE.

7-22-88


SPUR LEVEL OUTAGE
FAULT TREE
ORNL-DWG 88M-13017
NOTES:
1. SPUR STATION OR SPUR HOP (SPUR TO BACKBONE) PROBLEM.
2. SPUR LEVEL OUTAGE ALARM WILL IDENTIFY HOP. IF HOP NAME IS NOT PRESENT THROW AWAY MWM ALARM AND DON'T DIAGNOSE.
3. BADGER DATA RETAINED FOR 15 MIN AFTER THE ALARM IS CLEARED
4. HOP NOISE HISTORY:
CF BASED ON 0 - 30 MIN.
CF = 15/30 = 50; 30/30 = 100; 60/30 = 100
5. IF POLL FAILURE REVIEW DATA AFTER IT CLEARS
SPUR LEVEL OUTAGE
EQUIPMENT PROBLEM
(ONE DIR)
A XMTR B PROBLEM
LOOK AT ALL XMTR & RCVRS. IN MW MON. SECT. (ONE DIR ONLY)
B RCVR A PROBLEM
50
A TX B ALARM
B RX A ALARM
25
A RX B ALARM
B RX A ALARM
25
B NDIF A
A NDIF B
WEATHER PATH FADE (BOTH DIR)
40
NDIFF
SPUR A NDIF B
BACKBONE B NDIF A
30
MW SIGNAL DROPPING
NOISE OUTAGE
NOISE PERF
RECEIVED AT EITHER RTU WHICH INCLUDES THE HOP IDENTIFIED WITH SPUR LEVEL OUTAGE
20
RCVR MUTING
SPUR A RX B
BACKBONE B RX A
LONG TERM HISTORY (1 yr.)
18
SUMMER 6/1 - 9/15
42
HOP NOISE HISTORY (DBMS)
* ATOMSPHERIC DISTRUBANCE LAYERING OF AIR => MW DISTORTION
EQUIPMENT PROBLEM
(BOTH DIR)
DOME ICING
WINTER NOV 1 - MAR 30
WAVE GUIDE OR ANTENNA SHIFT
RECENT HISTORY
40 NDIFF
A NDIF B
B NDIF A
30 MW SIGNAL DROPPING
NOISE PERF
NOISE OUTAGE
RECEIVED AT EITHER RTU WHICH INCLUDES THE HOP IDENTIFIED WITH SPUR LEVEL OUTAGE
20 RCVR MUTING
A RCVR B
B RCVR A
LONG TERM HISTORY
18
SUMMER SEASON
42
HOP NOISE HISTORY (DBMS)
MUST HAVE 2-WAY PROBLEM SHOWN BY BADGER ALARMS BEFORE THIS FIRES
8 15 88

INTERNAL DISTRIBUTION

1. F. P. Baxter
2. R. G. Edwards
3. S. J. Dale
4-6. R. T. Goeltz
7-11. A. M. Hutton
12. H. L. Hwang
13. J. O. Kolb
14. M. A. Kuliasha
15. W. R. Nelson
16. C. H. Petrich
17. S. L. Purucker
18. D. E. Reichle
19. D. T. Rizy
20. A. K. Sexton
21. R. B. Shelton
22. R. Solanki
23-25. B. E. Tonn
26. A. Zaltash
27. ORNL Patent Office
28. Document Reference Section
29-31. Laboratory Records
32. Laboratory Record - RC

EXTERNAL DISTRIBUTION

33. B. G. Buchanan, Computer Science Department, University of Pittsburgh, 206 Mineral Industries Building, Pittsburgh, PA 15260
34. J. J. Cuttica, Vice President, End Use Research and Development, Gas Research Institute, 8600 W. Bryn Mawr Avenue, Chicago, IL 60631
35. A. Hirsch, President, Dynamac Corporation, The Dynamac Building, 11140 Rockville Pike, Rockville, MD 20852
36. D. E. Morrison, 333 Oxford Road, East Lansing, MI 48823
37. M. Williams, Professor, Department of Economics, Northern Illinois University, DeKalb, IL 60115
38-40. D. MacGregor, MacGregor-Bates Inc., 221 Mill Street, P. O. Box 10105, Eugene, OR 97440
41-43. T. P. Wiggen, Assistant Professor, Department of Computer Science, University of North Dakota, Box 8181, University Station, Grand Forks, ND 58202-8181
44. Office of Assistant Manager for Energy Research and Development, DOE-ORO, P. O. Box 2001, Oak Ridge, TN 37831-8600
45-55. OSTI, U. S. Department of Energy, P. O. Box 62, Oak Ridge, TN 37831