ornl

ORNL/TM-10429

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0266771 8

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

Expert Systems and Microwave Communication Systems Alarms Processing: A Feasibility Study

Bruce Tonn
Richard Goeltz
Steve Purucker

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this report, send in name with report and the library will arrange a loan.

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

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A05 Microfiche A01

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-10429

ENERGY DIVISION

EXPERT SYSTEMS AND MICROWAVE COMMUNICATION SYSTEMS ALARMS PROCESSING: A FEASIBILITY STUDY

Bruce Tonn
Richard Goeltz
Steve Purucker

Date of Issue - July 1987

Research sponsored by the
Bonneville Power Administration

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

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0266771 8

TABLE OF CONTENTS

Page

EXECUTIVE SUMMARY vii

INTRODUCTION 1

1. ARTIFICIAL INTELLIGENCE, EXPERT SYSTEMS, AND POWER SYSTEMS ... 4

1.1 AI BACKGROUND 4
1.2 AI AND POWER SYSTEMS 7

2. PROBLEM STATEMENT 13

2.1 THE MICROWAVE COMMUNICATION SYSTEMS AND ALARMS 13
2.2 VALUE OF EXPERT SYSTEM ALARM PROCESSING 17
2.3 THE COMMUNICATION ALARMS PROCESSOR 18

3. PROTOTYPE SCOPE 22

4. SOFTWARE SPECIFICATIONS 24

4.1 EXPERT SYSTEM SOFTWARE SPECIFICATIONS 24

4.1.1 Selection of the Expert System Architecture 24
4.1.2 The Rete Algorithm High Performance 28
4.1.3 Optional Software Specification 31
4.1.4 Expert System Software Issues - Weighing the Pros and Cons 33

4.2 DATABASE MANAGEMENT SYSTEM SOFTWARE SPECIFICATIONS 35

5. EVALUATIONS OF SOFTWARE AND HARDWARE CONFIGURATIONS 38

6. RECOMMENDATIONS AND PROTOTYPE DEVELOPMENT 44

REFERENCES 47

APPENDIX A. PRODUCTION SYSTEM SPEED IMPROVEMENTS THROUGH ADVANCED HARDWARE 50

APPENDIX B. DATABASE MANAGEMENT SYSTEMS EVALUATION CHECKLIST 52

APPENDIX C. EXPERT SYSTEM SOFTWARE DESCRIPTIONS 55

APPENDIX D. DATABASE MANAGEMENT SYSTEM SOFTWARE DESCRIPTIONS 61

LIST OF FIGURES

Page

Figure 2.1. BPA Microwave Commmunication System 14

Figure 2.2. Example of CAP Configuration 19

LIST OF TABLES

Page

Table 1.1. Power System/AI Work 11

Table 1.2. Expert System Features (A Wish List) 12

Table 5.1. Expert System Software Evaluation 39

Table 5.2. Hardware/Software Configuration Evalutions 42

EXECUTIVE SUMMARY

This report presents the products of a study by Oak Ridge National Laboratory for the Bonneville Power Administration on the feasibility of an expert system to process alarms emanating from Bonneville's Microwave Communication System (MCS). Expert systems, a branch of Artificial Intelligence (AI), have been cited by the Electric Power Research Institute as an important new research and development area within the power systems community (EPRI 1986). Expert systems could aid in power system planning, design, operation, maintenance, education and training, all resulting in potentially significant improvements in power system management and reliability. Based on literature reviews, direct research experience, and discussions with Bonneville staff, it is concluded that alarms processing is an appropriate domain for expert systems technology.

A prototype communications alarm processing (CAP) would contain a data base management system (DBMS) as well as an expert system module. The prototype would encompass an appropriately challenging subset of alarm systems, alarm networks, and alarm types. It would be capable of intelligently processing alarms, efficiently storing alarm-based historical data, providing custom reports, and supporting statistical analysis. Such a system has the potential to improve response to critical alarms, increase the information content of large volumes of complicated data, free operators from performing routine analysis, and provide alarm information to operators, management, and field personnel through queries and automatically produced reports.

The CAP prototype is a logical project to initiate Bonnevilles' Artificial Intelligence activities. The communication system application is less critical than a power system application, which means that mistakes made in the learning process will not have potentially costly consequences. The volume of alarms into the prototype will be neither too large nor too small. The MCS application requires only a relaxed real-time response to operators, which is the conceptual first step toward designing subsecond response hardware/software configurations. Lastly, the AI lessons learned and experiences gained in this project should be directly transferable to other Bonneville power system applications.

The major expert system requirement is speed. Alarms will need to be processed in near real time if the expert system is to aid human MCS operators. Forward chaining, Rete-based algorithms offer the best processing speed and should form the heart of any expert system software chosen for the prototype. The software should also offer a powerful development environment, and some flexibility in knowledge representation and inferencing. The DBMS module presents no challenges that numerous commercially available packages cannot handle. Therefore, DBMS selection should be driven by software availability, cost and compatibility with the expert system software.

It is recommended that the expert system module of the CAP prototype be developed on a dedicated AI workstation using a major, commercially available expert system development system. The DBMS prototype can reside on any convenient computer that can be networked with the AI

workstation. This development environment will maximize productivity, allowing Bonneville to produce a prototype system in the most timely and cost efficient manner. The CAP prototype should be delivered on a general purpose workstation that will effectively network with, but not interfere with, existing Bonneville computer hardware.

INTRODUCTION

This report presents the results of a feasibility study conducted by Oak Ridge National Laboratory (ORNL) for the Bonneville Power Administration concerning the applicability of Artificial Intelligence (AI) technology to process alarms associated with Bonneville's Microwave Communication System (MCS). Specifically, the discussion focuses on the characteristics of a prototype expert system/database management system (DBMS) configuration capable of intelligently processing alarms, efficiently storing alarm-based historical data, and providing analysis and reporting tools. Such a system has the potential to improve response to critical alarms, increase the information content of a large volume of complicated data, free operators from performing routine analysis, and provide alarm information to operators, field personnel, and management through queries and automatically produced reports.

This report begins with a general discussion of Artificial Intelligence and expert systems and AI applications to power systems. This section provides definitions of key concepts and illustrates expert systems capabilities. The second section describes, again in general terms, the Bonneville power systems environment and, in detail, the potential benefits of integrating intelligent systems into the current MCS. Developed is an understanding concerning the general capabilities that the prototype system should possess.

The third section presents suggestions on limiting the scope of the prototype project. Expert system development is very much an art and leading practitioners strongly advise a cautious, incremental approach (Waterman 1985). The constrained problem offered in this section is sufficiently challenging to serve as a true test of the applicability of the expert system technology to processing alarms but is manageable in size and complexity.

The fourth section discusses in detail suggestions pertaining to the configuration of the prototype system and specifications for the expert system and database management system modules. The configuration encompasses links between incoming alarms, the expert system, the DBMS, the operator, and district personnel. A forward-chaining production system based on the Rete algorithm, and the Structured Query Language (SQL) standard head the list of specifications for the expert system and DBMS modules, respectively.

The fifth section contains an assessment of the prototype system development options. A wide range of hardware and software options are considered; specifications made in Section 4 allow the analysis to concentrate on a small number of possible hardware and software configurations. The highest rated option consists of a development environment encompassing a powerful AI workstation and a state-of-the-art artificial intelligence software package to be networked with a conveniently available computer system that would host a relational DBMS. The prototype would be delivered on a dedicated workstation able to host both the AI and DBMS software.

The report concludes with a summary of our recommendations and an outline of how the prototype development effort would proceed. The latter discussion includes suggestions on level of effort and schedule.

1. ARTIFICIAL INTELLIGENCE, EXPERT SYSTEMS, AND POWER SYSTEMS

1.1 AI BACKGROUND

The term artificial intelligence was purportedly coined in 1956 by Stanford Professor John McCarthy (Feigenbaum 1979). It came to represent the goal of programming a computer to possess intelligence theretofore an attribute of human beings and other higher life forms. Early research emphasis rested on problem solving algorithms and means to heuristically search very large solution spaces.

Expectations among the original AI researchers were very high. It was thought that general problem solvers could be built to cover the entire range of human problem solving skills. Newell and Simon (1972) were in the forefront of this effort. In the last thirty years, AI has grown to encompass natural language processing, robotics, cognitive science, automated reasoning, machine learning, pattern recognition, and expert systems. In each area, except possibly the last, high expectations have been replaced with an appreciation of the monumental challenges associated with programming computers to carry out these functions.

For example, researchers in problem solving have found that computers require immense amounts of knowledge in addition to problem solving methods in order to function in specialized fields such as medicine. From this realization evolved the sub-field of expert systems, which are computer programs that encompass large amounts of painstakingly acquired

human expert knowledge. Most expert systems have knowledge represented as if-then rules and employ one of two basic means for searching the knowledge base, backward or forward chaining, to make inferences between the knowledge and user-supplied data. Diagnostic systems employ backward chaining and monitoring systems typically employ forward chaining. Expert systems have also been developed to interpret data, design processes, and render predictions (Stefik et al., 1982).

Buchanan (1986) states that expert systems have four distinctive characteristics. (1) Expert systems reason with symbolic information and use non-algorithmic inference methods. (2) They render high performance; the systems will yield correct decisions on par with the experts. Quality expert system performance is achieved in narrow domains that have stable knowledge. Some expert systems have achieved correct decision rates that exceed 80%. Most successful systems have been very technical in nature (e.g., medicine, computer fault diagnosis). (3) The programs are flexible in that new knowledge can be easily added to knowledge bases and they can be employed with different kinds of users. (4) Decisions made by the systems are understandable, in that the expert systems can report what knowledge was used to make the decision. These four characteristics make expert systems very different from conventional programs (Waterman 1986).

Hundreds of expert systems have reached the research prototype stage but few have been fully implemented (Waterman 1985). Among the more notable systems are MYCIN, R1, and PROSPECTOR. The first is implemented in LISP and is designed to diagnose blood diseases and recommend appropriate therapy. Though never put into actual use in the

medical field because of its slow processing capabilities (Buchanan and Shortliffe 1984), the MYCIN project became the progenitor of dozens of other expert systems because its inference engine and user interface were stripped out and reused.

R1, a commercial success, was jointly developed by Digital Equipment Corporation (DEC) and Carnegie-Mellon University and is designed to configure all VAX family systems using data included on purchase orders. R1 has saved DEC millions of dollars by eliminating expenses associated with sending wrong equipment. PROSPECTOR, developed by Stanford Research Institute, analyzes geological, geophysical, and geochemical data concerning a particular geographical area and predicts the existence of certain mineral deposits. Although it is not in general use, it has accurately predicted a substantial molybdenum ore deposit in a previously unexplored location (Harmon and King 1985).

The seemingly endless potential of expert systems, contrasted with the present level of limited success, indicates that expert system development projects are risky endeavors. At least three major reasons contribute to the riskiness. (1) Development time is substantial. Waterman (1985) estimates anywhere between 2-5 years for a complex and valuable expert system. The problem is the "knowledge acquisition bottleneck" (Feigenbaum and McCorduck 1983). Knowledge engineers spend long periods of time interviewing experts, making sense of the answers, coding the answers into representations suitable for the computer, and testing the resulting knowledge bases. Often the sponsoring organization withdraws support for the project before the expert system is finished.

(2) While guidelines exist relating to what is and what is not a suitable expert system domain, the guidelines are not rigorous enough. As a result, many expert system developers find their potential applications to be too simple or way too complex only after investing in prototype systems. (3) The cooperation of the expert is not assured. Projects fold when the expert is too busy to be interviewed or cannot verbalize his/her expertise. For these three reasons, expert system practitioners very strongly recommend that the systems be built in stages, starting with a rapid prototype to test domain suitability and the expert's performance.

1.2 AI AND POWER SYSTEMS

Within the past five years interest in artificial intelligence technologies has increased tremendously in the power system community. The Electric Power Research Institute (EPRI) has commissioned exploratory studies to determine suitable power systems applications of AI (EPRI 1986). In addition, a small number of utilities, universities and research organizations are working to produce test-of-concept systems. Most utility related work has focused on applying expert systems to power system problems, although there is a realization that other AI technologies such as voice recognition may also hold significant potential. This section documents important work in the expert systems/power systems area.

The application of expert systems to power systems falls into five general categories: (1) planning, (2) design, (3) operation, (4) maintenance, and (5) education and training. Planning encompasses contingency

analysis and system design. Sound planning requires the application of standard planning tools such as load flow, economic analysis, stability, etc. Expert systems could aid planning by assisting in load flow analysis, contingency selection and analysis, and specification and analysis of planning alternatives. With respect to design activities, expert systems are envisioned to evaluate engineering alternatives and associated economics by applying diversified expertise and standard design procedures.

The last three areas - operation, maintenance, and education/training - have direct applications to the Bonneville microwave monitoring project. Operation expert systems are divided into contingency analysis (studies) and alarm-restoration [real-time (RT)] activities. The former could assist operators in selection and analysis of contingency situations as well as improve maintenance. The latter type of expert system, which is of most interest to this project, can be designed to only monitor systems or to make split-second decisions that control systems.

Expert systems can suggest repairs and preventive maintenance on an as-needed basis, which should result in a more economical maintenance program. Opportunities also exist for including repair guidance into diagnostic expert systems. Lastly, the expertise of several microwave monitoring experts can be captured in expert systems so company expertise is retained and expert consultation is available to the on-duty operator when an emergency occurs. New operators could be trained more easily because operator expertise could be available in expert system "fault simulators."

A partial list of power industry expert systems work is shown in Table 1.1. The work falls into three of the five basic categories discussed above - planning, operation (operating studies and real-time assistant) and maintenance. Two projects in the Real-Time Advisor category are worth special mention. First is the work being done by Control Data Corporation (CDC) on an intelligent alarm processor for an energy management system (EMS) (Wollenberg 1985). A prototype system was developed by CDC and the University of Illinois to demonstrate the feasibility of an alarm processor on an EMS system. Currently CDC is developing an expert alarm processor for Northern States Power using EPRI funds. The alarm processor will be rule-based, without graphics, and will be developed in LISP on a Symbolics machine then converted to PASCAL. The language conversion is required to accommodate the CDC computers.

The second project is not a power system application. It involves monitoring satellites in real time and diagnosing equipment problems. The Ford Aerospace project is included in Table 1.1 because it is similar to Bonneville's microwave monitoring and diagnostic tasks. It is also a second generation expert system (Siemens et al., 1986). Experiences gained from StarPlan I, a rule-based system written in OPS5, led to the development of StarPlan II. StarPlan II's knowledge architecture was restructured to a "model-based system" (i.e., object-oriented programming) using Intellicorp's KEE. This work is interesting because EPRI selected KEE for its power plant expert system modeling and operation diagnostic research.

Power system alarm processors have been identified by EPRI as being a good application for expert systems (EPRI 1986). However, our literature review indicated that power system applications using expert systems appear to be limited and preliminary, involving prototypes. In contrast, expert system applications developed by NASA and the Department of Defense are broader in scope and are technologically more mature. Nevertheless, it appears that a microwave alarm monitoring expert system is feasible because: (1) an expert can successfully solve the problem, (2) the expert's ability is based on specific knowledge, judgement and experience, and (3) the operator's knowledge can be expressed explicitly.

A "wish list" of expert system features is shown in Table 1.2. These features were obtained from the literature, and a knowledge of real-time utility systems and trends in expert system technology. All the features shown are not recommended for the Bonneville microwave expert system, since some are beyond existing technology. This table is intended to show the range of features an expert system could possess. The final selection of expert system features are dependent upon hardware and software capabilities and tradeoffs as well as features that best fit the microwave monitoring expert system problem.

Table 1.1. Power System/AI Work

Type/Location	Description	Language	Maturity
PLANNING			
Mitsubishi (Fujiwara et al., 1985)	Intelligent Load Flow Engine	-	Test
University of Washington (Liu et al., 1985)	Reactive Power/Voltage Control	OPS5	Prototype
ORNL	Loss Reduction	Intelligence Compiler	Development
OPERATION: STUDIES			
Carnegie/Allegheny/EPRI (EPRI 1986; Talukdar et al., 1985)	Protection System - Simulator and Diagnosis	OPS5	Prototype
University of Washington	Protection System - Simulator and Diagnosis	OPS5	Prototype
Mitsubishi	Protection System	-	-
OPERATION: REAL-TIME ADVISOR			
CDC/University of Illinois (Wollenberg, 1985)	EMS - Intelligent Alarm Processor	LISP	Prototype
CDC/N. States Power	EMS - Intelligent Alarm Processor	PASCAL	Development
EPRI - Intellicorp (EPRI 1985)	PLEXIS - Generation Plant	KEE	Operational
CDC	Unit Commitment	PASCAL	Planning
ORNL	Operator Advisor	LISP	Development
ORNL	Distribution	PASCAL	Development
Ford Aerospace (Siemens et al., 1986)	STARPLAN II: Satellite Monitor	OPS5/KEE	Development
MAINTENANCE			
GE/Honeywell (Koch et al., 1987)	Power Plant Repair	-	Prototype
CDC/PE	Training Simulator - EMS	-	Development

Table 1.2. Expert System Features (A Wish List)

- Fast and reliable alarm analysis with temporal reasoning
- Explanations for the operator
- Alarm screening - identify significant events and important alarms
- Include uncertainty
- Transform raw data => big picture
- Crisis management - consistency and multiple expertise
- Assist with economic maintenance program
- Detect data discrepancies
- Graphics interface
- Uses real-time data - conclusions and recomendations based on best data
- Speech and speech recognition
- Ability to learn
- Natural language interface
- Easy to update - add new station and microwave links
- Include microwave system topology
- Include microwave station and device parameters including temporal aspects
- Include behavorial characteristics - network response to stimuli
- Incorporate procedural programs
- Incorporate database operations
- Keep historical profiles
- Communicate efficiently with other computers
- Operator friendly - avoid blackbox syndrome
- Supplement-<u>not</u> replace existing microwave monitoring system

2. PROBLEM STATEMENT

This section first describes the environment of the Bonneville microwave communications system, its alarms systems and the problems that the alarms systems present to users of the alarm data. Second, reasons for processing alarms with expert system technology are given. Finally, a design for a communications alarm processor (CAP) is presented that would address a large number of the problems identified.

2.1 THE MICROWAVE COMMUNICATION ALARMS SYSTEMS

Bonneville's Dittmer Control Center houses the operations whose responsibility is the safe, reliable, and economic operation of the Federal Columbia River Power System, which includes Bonneville's power transmission network (BPA 1981). This network spans an area of over 300,000 square miles; Washington and Oregon are encompassed entirely as are parts of Idaho and Montana. The 13,000 miles of high voltage transmission lines interconnect 30 federal hydroelectric dams and 14 other power generating utilities at more than 150 connection points.

To support the transmission system, Bonneville operates a system-wide microwave communications network for supervision and control functions, which include: automatic generation control; protective relaying; load tripping; generation dropping; stability scheme analysis; analysis of critical telemetering quantities; Supervisory Control and Data Acquisition Systems (SCADA); and voice communications. The microwave network consists of 7 major systems designated A, B, D, K, N, Q, and T (Fig. 2.1) and includes 141 microwave stations at 80 mountain-top repeaters and 61 substations.

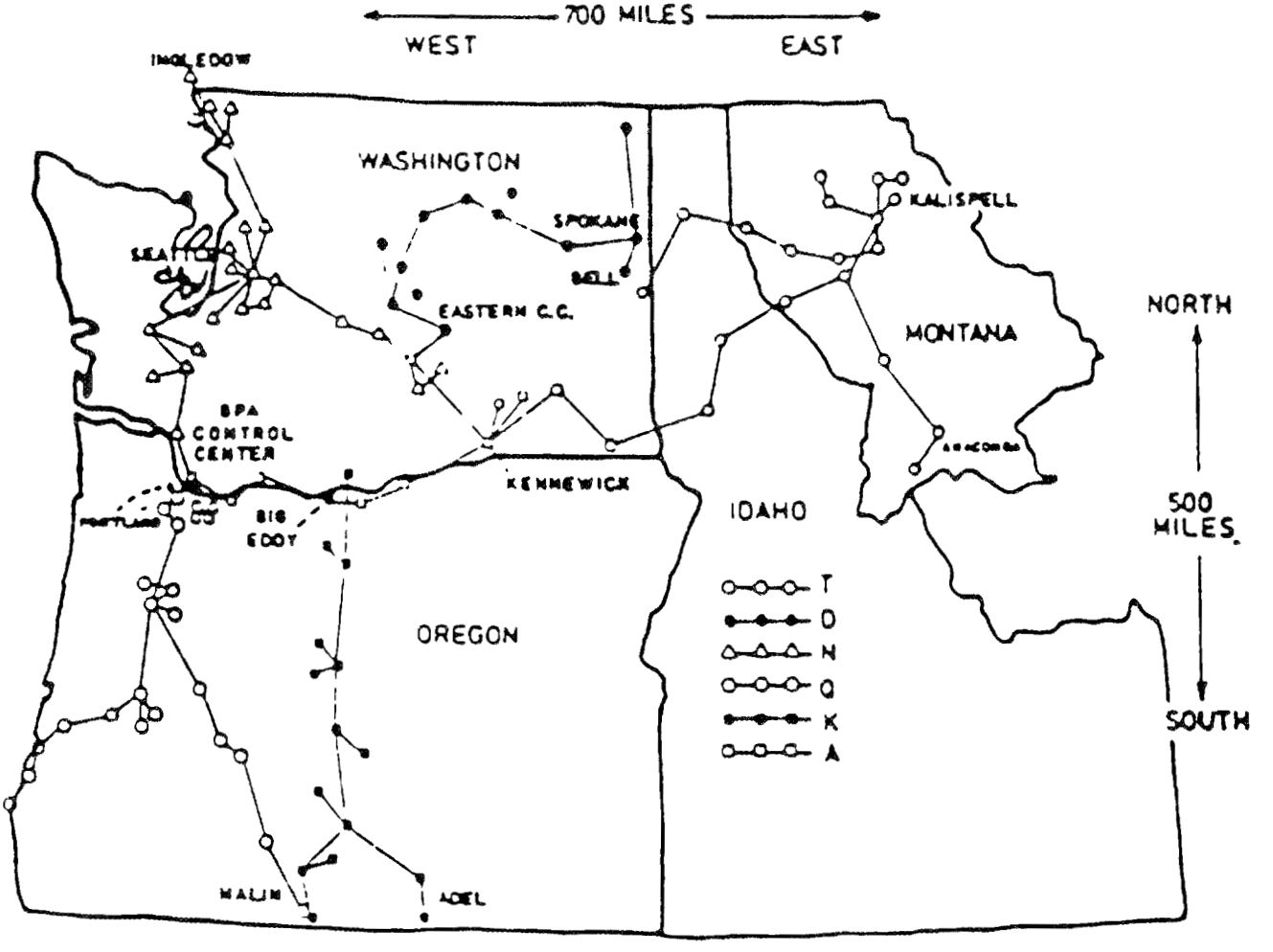

Figure 2.1. BPA Microwave Communication System

Since any problems with the communications network could seriously reduce the reliability of the power system, maintaining a reliable communications network is essential. The importance of optimized operation and maintenance of the communication system continues to grow because of the trends towards centralized operations, automation of control, and reductions in operations and maintenance personnel.

There are a number of alarm systems which are used to monitor the performance of the microwave network, report problems, and assist in problem diagnosis. These alarm systems are the Microwave Monitor (MWM), the Badger system, SCADA, transfer trip, and telemetering. The MCS alarm systems generate compact and informative alarm messages that include the problem description, problem location, and date and time of problem occurence. Typically, an alarm description is indicating that one of the MCS performance parameters has exceeded a normal limit. Also, alarms are generated if there are abnormal conditions with MCS support equipment (e.g., building, power generators, batteries). This level of information content is in contrast to many alarm systems which are reporting raw physical measurements that are not necessarily abnormal. The time field of MCS alarms typically have one second resolution; however, some are generated from summaries of five minutes of data.

There are several categories of Bonneville staff which rely on the alarm data for information during day-to-day work activities. First and foremost are the network operations and maintenance staff whose duties include monitoring the microwave alarm systems, interpreting alarm data, determining repair and maintenance actions (e.g., the operator may dispatch a repair team), computerizing and disseminating alarm data to

other groups, and analyzing and reporting historical trends. A second group which uses a subset of the microwave alarm data is the Control System Monitor (CSM). These operators maintain a 24-hour vigil of all monitoring and control system hardware. Finally, subsets of the alarm data are distributed to management functions within the Dittmer Control Center and externally to Bonneville's Power System Control Offices.

The microwave network alarm systems present a number of problems to these three user groups. Most critically, the alarm systems provide too much data and too little information, a typical problem with current power system alarm and message systems. The microwave network is large and complex and the volumes of alarms are too great for a human to effectively process. For example, the Badger system generates an average of 2,500 alarms per day. During a network disturbance, however, the rate of alarms is much higher than the average and, unfortunately, this is precisely the time when quick and accurate interpretation by an alarm-monitor operator is needed.

This volume of alarm data could be reduced if redundant data could be filtered or condensed. Often, a long sequence of alarms really only presents information that could be effectively reduced to a single message or a small number of messages. Examples include alarm symptoms which are occurring intermittently and alarms which are side effects of only one equipment problem. Unfortunately, analysts now must sort through redundant data to find kernels of important information.

The microwave network systems transmit a large amount of data not only because the communications network is large and complicated but also because the three user groups are diverse and geographically dispersed.

It would be exceedingly valuable to filter from the data given to each group those portions that are not providing relevant information. Additionally, expert-level interpretation of alarm data could significantly condense the data and provide diagnostics to operators. Finally, it would be valuable to subset, prioritize, and route the messages to those locations where they are needed (Amelink et al., 1986). As the number of users and the value of the data grow, there is an increasing need for an effective alarm information management system.

2.2 VALUE OF EXPERT SYSTEM ALARM PROCESSING

Expert system technology can help overcome the data burdens mentioned above and an alarms-processing prototype expert system could benefit Bonneville's power system artificial intelligence activities.

With respect to the former, an expert system could easily filter redundant alarms. More work would be needed to compress alarms into concise conclusions of MCS equipment problems. However, an expert system is ideally suited to the task because the compression/diagnostic exercise employs symbolic rather than algorithmic problem solving; experts can solve an alarm problem in a reasonable period of time, and alarm data can be drawn directly from MCS channels. After expert-level interpretation the compressed information can then be made available to important user groups.

An alarms processing expert system is a good choice for BPA to begin its AI activities for several reasons. (1) This application is less critical than other potential application areas (e.g., SCADA). Thus, mistakes made in the learning process will not have potentially

expensive consequences. (2) The volume of input data is neither too much nor too little. (3) This application need only meet a relaxed real-time requirement, which is the conceptual first step toward designing a sub-second response-time expert system. (4) The lessons learned and experiences gained should be transferable to most other potential Bonneville power system applications. Finally, (5) a domain expert is available who exhibits a high level of expertise and is able to verbally express this knowledge.

2.3 THE COMMUNICATIONS ALARMS PROCESSOR

A communications alarms processor that includes the expert system prototype discussed above would alleviate a number of the problems associated with the use of the alarms data. Figure 2.2 presents an example CAP configuration which is composed of three major components: (1) input consisting of the integration of the microwave alarm systems; (2) an expert system to perform alarm filtering, compression, interpretation, and diagnosis; and (3) DBMS to manage the expert-system-processed alarms.

The CAP process begins by accessing the microwave alarm systems, which are currently independent, and integrating them into a set of inputs to the expert system (ES). An input/output (I/O) process accepts the alarm messages in real time from the microwave alarm systems, translates them into ES syntax, and buffers them until the ES performs a read operation.

In order for the ES to respond to alarms quickly and intelligently, it must possess the following five capabilities:

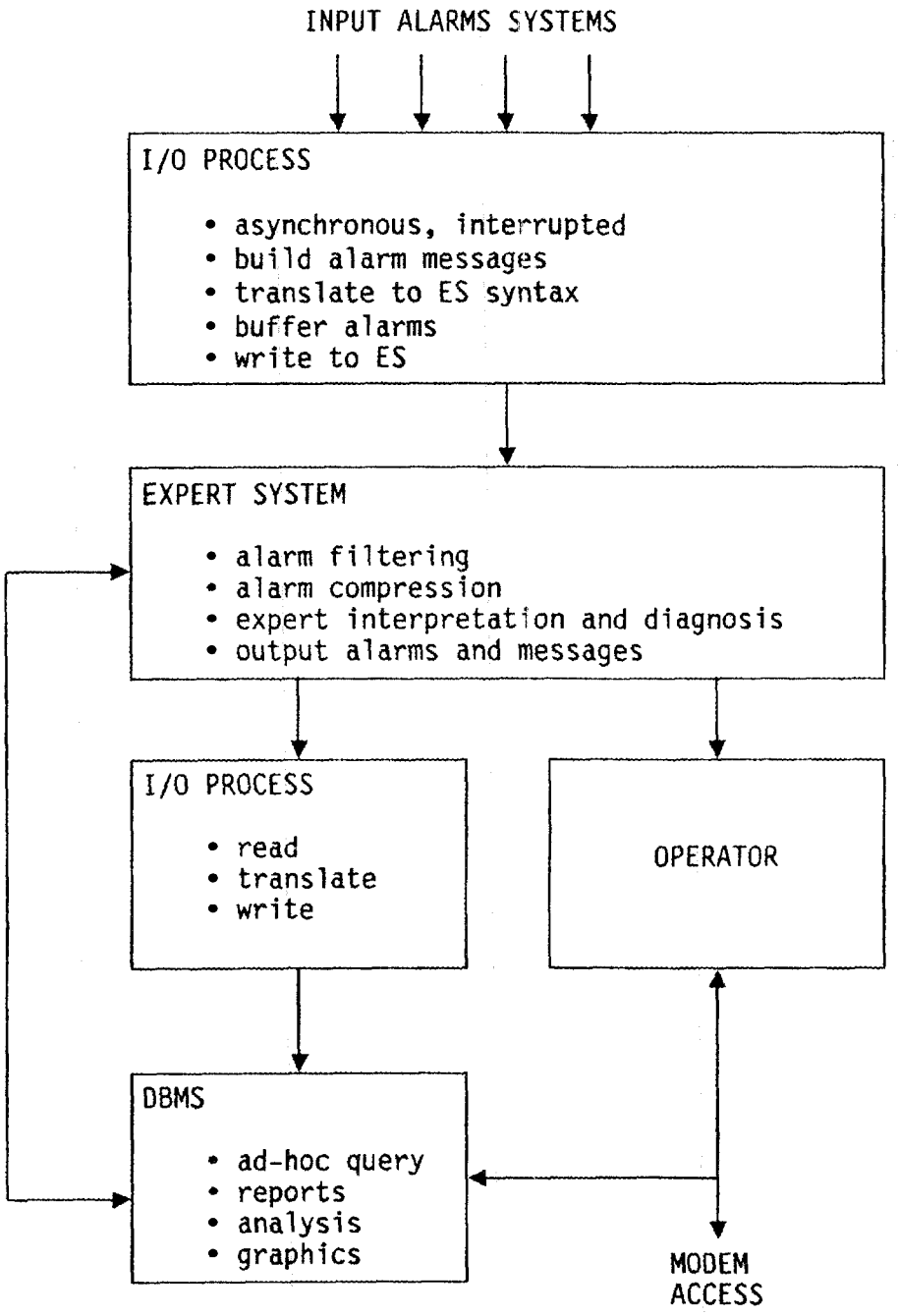

Figure 2.2. Example CAP Configuration

(1) Be able to read alarms from the I/O buffer promptly;

(2) Allow the incoming data to trigger the inference processes of the inference engine;

(3) Utilize an alarm priority scheme to efficiently and effectively control the inference processes;

(4) Tolerate interruptions of the inference subprocesses; and

(5) Deliver output messages without delay.

For example, suppose a noncritical alarm is being processed by the ES and a critical alarm arrives at the input buffer. The noncritical process should be interrupted, the critical alarm read and fully processed (including output messages), and then the noncritical alarm process should continue unless it was affected by side affects of the interrupting process.

Expert system preprocessing of alarms will begin by detecting transient alarms. Next, any time-out alarm, which indicates that a condition has ended, will be combined with the alarm that earlier had reported the start of the condition (the time-in), effectively reducing two alarms to one. Furthermore, alarm preprocessing will combine multiple alarms into one when an intermittent condition is present (e.g., when three or more identical alarms occur within a five-minute period).

After input alarm preprocessing, the ES will continue to respond to new alarms, previously received alarms, intermediate conclusions, and initiate intelligent assessment of alarm conditions. On this last point, the set of possible actions the ES can choose from will be quite large and complex, due to the complexity of the input data and the complexity of the expertise that will be encoded and applied to the problem. In

very general terms, the ES will further reduce the volume of alarms while increasing the information content by explicitly reporting MCS problems. Also, operations and maintenance personnel will be assisted in operations requiring near real-time response because the ES will offer interpretive and diagnostic assistance, and automatically perform routine diagnostic procedures that currently require an operator's attention.

Another I/O process is needed to handle the data traffic between the expert system and the DBMS. The burden of translating from the syntax of the expert system to the syntax of the DBMS is assigned to an algorithmic task.

The DBMS will archive the expert-system-processed alarms for analysis and reporting. The DBMS will provide the user interfaces with the data and the expert-system interpretations. A number of services will be provided by the DBMS including ad-hoc query of the alarms data, and generation of of reports, statistics, and graphics. Also, the DBMS can provide, upon request from the ES, data from the archived records (e.g., historical trends) that the ES may need for diagnostics.

3. PROTOTYPE SCOPE

As mentioned in Section 1, expert system development projects must proceed incrementally. The most favorable approach is to first rapidly develop a small prototype over a period of six to eight weeks. This exercise will indicate the suitability of the domain, the software, and the chosen expert. After 10 to 12 months, what is known as a research prototype is developed. This second stage product covers all the challenges of a fully implementable system but encompasses only portions of the larger problem. This section contains recommendations for scoping the research stage of the CAP prototype.

As described in the previous section, the CAP project involves a large amount of data and requires ES processing that is likely to be very complex. To reduce the size of the CAP prototype to manageable dimensions, two of the configuration components are reduced: the input data and the expert system's responsibilities.

Data for the expert system will come from two of the real-time microwave monitoring systems, the Microwave Monitor (MWM) and the Badger. It is estimated that these two input systems can provide the data necessary to diagnose about 90% of the microwave system alarms. Future intelligent alarm processors may incorporate alarms from other control and monitor functions if the benefits outweigh the increased costs of complexity. The MWM, designed and built by Bonneville generates equipment alarms, processes field data, and displays processed data to operators in rigid formats. The Badger system contains detailed data from the microwave stations. It is expected that in the prototype the MWM alarms will drive the expert system, with the Badger alarms being used

for confirmation purposes. If time and resources permit, Badger alarms will also be processed independently of MWM alarms.

The prototype will be built around a subset of network stations and alarms. With respect to the former, discussions with Bonneville staff suggest that the prototype system be built around the N-system. It is the largest and most complex of the seven BPA microwave monitoring systems, and, therefore, presents the development team with all relevant challenges. Also, the N-system is composed of stations with modern, homogeneous communications equipment. Thus, selection of the N-system simplifies diagnosis that otherwise may be affected by equipment type. Also, the equipment will not be replaced soon, adding to the useful life of the rules.

With respect to alarms, the prototype will diagnose the probable causes of all MWM outage and performance alarms (two of the five MWM alarm types). These are the two most critical types of MWM alarms and apparently are the easiest types to diagnose. All of the Badger data for the N-system will be processed, and the most useful information will be archived in the DBMS.

The DBMS component of the prototype will <u>not</u> be limited in scope or operation. It will archive all alarm diagnoses written by the expert system and all valuable Badger data. We recommend that 13 months of alarm data be kept on line allowing annual summaries. If alarm processing provides a 75 percent compression rate, then the DBMS will contain about 300,000 records (3,000 alarms per day times 25 percent times 400 days). DBMS users will be able to easily manipulate data in the fully implemented CAP system. An off-the-shelf DBMS based on proven technology will be used in the prototype system.

4. SOFTWARE SPECIFICATIONS

4.1 EXPERT SYSTEM SOFTWARE SPECIFICATIONS

This section analyzes the MSC prototype problem characteristics to identify general specifications for the expert system software. The specifications indicate the expert system architecture to be used and the underlying internals of the software that would be desirable. Attempts are made when appropriate to weigh alternative specifications.

4.1.1 Selection of the Expert System Architecture

There are a number of expert system architectures that have performed well for certain problem domains. Expert system architectures differ in how (1) input data are obtained, (2) knowledge is represented, and (3) solution spaces are searched. For example, rule-based systems represent knowledge in "if-then" statements and use either forward chaining or backward chaining inference. Forward chaining inference performs well when the problem is data driven. On the other hand, backward chaining inference is useful for problems where there are a small number of solutions to choose from. Other architectures center around object descriptions rather than rules. For example, semantic nets or frame-based systems perform well for problem domains concerned with modeling and design (Kunz et al., 1987).

Matching the problem of alarms processing to the set of possible architectures has focused on three areas: (1) the expert system response time desired; (2) the characteristics of the input data; and (3) the expert system solutions desired. As explained below, a <u>forward chaining architecture</u> has been chosen for the prototype expert system.

The most important performance characteristics of the CAP expert system are that it is expected to run continuously and reliably respond to alarms in near real-time. Quick, reliable response to critical alarms is essential when the expert system is operating as an operator assistant. Also, fast processing is required to ensure that all alarms are processed and that the CAP system does not fail due to exhaustion of a computer resource (e.g, memory). Beyond the prototype, system requirements will continue to demand fast processing of large amounts of input data. Therefore, this system-design criterion is so pervasive that it affects all elements of CAP system design. A forward-chaining architecture, using a rule-based knowledge representation, if properly designed and implemented, can deliver the processing speed requirements of the prototype better than the backward chaining and object-oriented alternatives.

The characteristics of the input data match well with requirements and capabilities of a well-tuned forward chaining system. Kline and Dolins (1985, 1986) list three important characteristics: (1) The data input to a forward chaining expert system must be reliable and complete; (2) additional data requirements beyond data that drive the ES must be minimal; and (3) data be available in a very timely manner. With respect to these criteria, discussions with Bonneville staff indicate that alarm data are reliable and complete (i.e., the alarms from the Microwave Monitor and the Badger alarms systems, see Sections 2 and 3). In addition, it is envisioned that only occasionally will the expert system require data from the DBMS or operators, and, of course, the alarm systems should provide alarm data on a near-instantaneous basis.

For the CAP prototype, a number of solution characteristics have been identified that match very well with the capabilities of a forward chaining architecture. First, it is evident from talking with MCS experts that the expertise that must be encoded for the CAP prototype is best expressed in an "if-then" rule format. Second, the alarm analysis does not involve extensive modeling of the structure and function of the microwave network, of component parts of the network, or of the alarms, which rules out an object-oriented approach. Third, the forward chaining architecture works well for problems where all of the solutions can be encoded in advance and where the solutions do not require deep reasoning. That is, the solutions do not require system knowledge about the underlying principles of microwave network operations. Finally, it appears that the solutions will not require the software to perform complex looping or branching, or analysis of interacting goals.

Backward chaining inference may, however, also be needed for some of the CAP expert system processes. Examples include ES interaction with the DBMS and selecting one from a known set of options (e.g., the alarm path has been narrowed to a small number of stations). Backward chaining is initiated as a goal that the forward chaining system is to pursue. Note that this is quite different from a true backward chaining inference engine that performs a depth-first search of the search space, usually in search of data that supports a solution from a small set of options.

There are at least six areas of concern surrounding the analysis of the problem characteristics and the recommendation to use a forward chaining production system:

1) The level of deep reasoning needed. The prototype system will operate as a situation (or sequence of situations) recognizer. which is a shallow reasoning application. However, it may become necessary to add rules that pertain to how pieces of equipment operate or interact. For example, an object-oriented approach may eventually be needed for extensive modeling of system structure and function.
2) The representation and propagation of uncertainty required. Uncertainty management is an important function within expert systems and many approaches have been proposed (Tonn and Uppuluri 1986). A problem may arise in mapping the application into the best uncertainty system.
3) The instantiation of rules that suggest mutually exclusive, disjoint sets of conclusions. For example, the expert system may report network problems that cannot happen at the same time. Deep reasoning or operator input may be needed to resolve these problems.
4) The complexity of the temporal reasoning and consistency maintenance required. Temporal reasoning may be required to identify complex sequences of events or to make predictions about the expectancy of events. Consistency maintenance will be required when conclusions are invalidated by new inputs and to remove from the system old and useless facts. Both of these are subjects of state-of-the-art AI research.

5) The level and type of operator interfaces needed for the real-time expert system. The CAP system design minimizes the interface between the operator and the expert system by using the capabilities of the DBMS. The reason is to allow the expert system to process alarms without interruption or without having to freeze the system state. However, it may be found that expanded operator support is needed for microwave network problem diagnosis.
6) An increasing demand for processing speed and efficiency. The pressures on system performance can come from two directions: larger amounts of input data and larger rule sets. Input data can increase in two ways: larger input records and faster transmissions rates. Also, rule sets can grow in two aspects: number of rules and rule complexity. As a prototype matures or is extended into similar application areas, processing demands will grow and may exceed the capabilities of the software architecture, the software implementation, or the hardware. The next two subsections explore an algorithm and other programming features which attempt to optimize inference system performance. Also, Appendix A explores ways to improve production system performance through hardware enhancements.

4.1.2 The Rete Algorithm and High Performance

The previous subsection presented analysis which yielded the recommendation that a forward chaining inference architecture be used for the prototype system. The following subsections specify that the software

selected should use an algorithm called the Rete match algorithm which will match the alarm data to the rules very quickly. A review of the basics of forward chaining begins the subsection and an explanation of how the Rete algorithm optimizes the operations that the inference engine performs follows.

A production system (PS) includes an unordered set of rules, or productions, with a syntax such as

```
IF
  condition 1

  condition 2
     :      :

THEN
  action 1

  action 2
     :   :,
```

where the conditions are called the left-hand side (LHS) of the rule and the actions the right-hand side (RHS). The facts of the production system (e.g., alarms, physical network descriptions) are contained in working memory (WM) and the productions are kept in production memory (PM). The conditions of a rule express patterns and pattern restrictions which describe the working memory elements (WME) necessary for the rule to become instantiated and eligible to fire (i.e., execute the actions). The condition patterns may contain constants, variables, and restrictive tests such as relational comparisons and function calls.

Typically, a forward chaining production system operates in a cycle with the following steps:

1) Match. Compare each LHS with the set of WM elements to determine which rules are eligible to fire. The resulting set of rules is called the conflict set.

2) Conflict resolution. Using an established procedure, choose one rule from the conflict set.
3) Act. Execute the actions on the chosen rule's RHS.
4) Halt or continue. If HALT is an executed action, then stop. Otherwise, continue the cycle with step (1) until no more rules are eligible to fire.

RHS actions that occur often are MAKE and REMOVE, which operate on facts in WM. Also, facts can be established by a read operation. Thus, WM changes slowly from cycle-to-cycle and these changes affect the conflict set, causing rules to be added or removed.

When a WME meets the pattern specifications of a condition, they are said to match. A large percentage of PS processing is spent matching WMEs to productions. In simple implementations of the software, where each WME is compared to each production on each cycle, the processing cost may be prohibitively high, particularly if there are a large number of WMEs, productions, or both.

Software techniques aimed at improving the match process try to take advantage of one or both of the following PS characteristics:

1) From cycle-to-cycle, WM changes relatively slowly, and
2) Rules often have in common similar conditions and sets of conditions.

The techniques include one or more of the following approaches (McDermott et al., 1978):

1) Data structures to indicate (e.g., index) which conditions occur in productions.
2) Data structures to remember which WMEs match which conditions.

3) Data structures to indicate which conditions are combined within which productions.

The Rete match algorithm uses all three efficiency improvement techniques (Forgy 1982). The algorithm takes advantage of the persistence of WM across processing cycles by matching only the changed WMEs against the productions. This requires storing with each pattern the WMEs that it matches. The memory of these must be updated on each cycle when WMEs are added or deleted. This improvement avoids iteration over WM.

The Rete algorithm also avoids iteration over production memory by using a tree structured sorting network. A rule compiler builds a network of nodes, each of which tests for one pattern feature, either within a pattern or to join patterns. Each time a WME is added or deleted, a token is passed through the network and the memories of the nodes are updated appropriately. If a token reaches one of the network's terminal nodes, then a change to the conflict set is indicated and a rule and its instantiations are either added or removed.

4.1.3 Optional Software Specifications

Even though the Rete match algorithm is widely acknowledged as the fastest method to match facts to rules for forward chaining expert system applications (Forgy 1987), work continues on improvements aimed at increasing processing speed.* The first two improvements described below are domain independent while the rest covered in this subsection are suggested as being potentially valuable for alarm processing and similar applications.

*Appendix A contains a discussion on hardware improvements related to improving processing speed.

One improvement has been pioneered by IBM (Ennis et al., 1986). Similar to CAP, their application involves significant amounts of external data (i.e., more than one fact may become available asynchronously). Rather than perform one I/O operation on each ES processing cycle, they added to the PS processing cycle a step which performs all necessary input and output. Thus all available input data is incorporated on a single cycle and output messages are delivered without delay.

IBM also implemented a facility which enables a fact to become true at a specified time in the future (i.e., a timed-MAKE). This is useful for specifying an action to take place depending on the outcome of a predicted event. Such a facility, which requires a timer function and timer queue, is likely to be more efficient than dynamically building and removing customized rules (i.e., each such rule will establish a fact at an indicated point in time).

A mechanism which provides prioritized conflict resolution is needed for CAP. For example, a critical alarm should be handled before a non-critical alarm. Explicit support by the expert system software of setting rule priorities and the use of these during conflict resolution is preferable to using metarules. Explicit support provides processing efficiency and more simple rules.

CAP must run continuously. If the input alarm systems are quiet and no rules are firing (a standard system would HALT) and the DBMS is being used on the same processor, then an efficient facility would halt the ES until it is awakened by an interrupt (probably caused by an input alarm or the timer facility). A simple but less efficient approach to continuous PS operation is to design a rule (e.g., with the lowest priority) that will always fire.

CAP must remove old facts from WM that are no longer needed. This allows continuous and optimized operation. If all of the old facts are not removed as a set (as opposed to removing one fact per operating cycle), then a rule may inadvertantly fire (e.g., a rule's NOT-condition becomes true). It would be difficult for the knowledge programmer to anticipate such a situation and it becomes necessary to introduce additional control over the PS's processing. It is desirable that the ES have a facility that allows the identification and orderly removal of dependent facts.

Several Rete-based PSs report improvements to the organization of the Rete network. For example, node joins from the right (instead of just from the left) and special joins of groups of condition patterns have been explored (Schindler 1986). Again, the emphasis is on improving the PS matching process and therefore processing speed.

Section 5 of this report examines a number of expert system shells and programming languages that are based on the Rete match algorithm. Valuable evaluation criteria is provided by considering the extent to which the packages support or can be extended to support various operations needed for real-time applications.

4.1.4 Expert System Software Issues - Weighing the Pros and Cons

The previous discussions gave general specifications for the ES software: forward chaining inference based on the Rete algorithm. This subsection discusses several other important issues that arise while evaluating expert system software. Careful consideration of the issues is required because it is important to carefully match the software to the application. Software selection will affect what problems can or

cannot be solved, the run-time performance of the expert system, the productivity of the programmer, the cost of the software, and the flexibility to incorporate new knowledge representations and inference techniques as the project develops.

Most of the software analysis considered thus far was concerned with run-time performance and problem-solution approaches. There is another operation mode of the ES software that is just as important - the development environment. The development environment will largely determine the labor cost for programming each increment of the development cycle. This feature should receive emphasis during software evaluation because this project has the goal of delivering a fully implemented development environment (as opposed to a run-time module) capable of supporting three goals: additional development of the CAP, development of other ES applications, and introduction to and education about ES technology.

A productive development environment usually includes the following features. Graphics capability is used to provide multiple windows into the working memory, the conflict set, and the production memory. Also, a graph can quickly summarize the relationships between groups of related data. An interface for browsing allows dynamic adjustment of the windows and their contents. A step facility allows the system to be studied before and after one rule has fired. Integration of the code editor and the ability to perform incremental compilations allows quick testing of new rules. Another useful feature is the ability to dynamically add or remove facts and rules. These

features do add to the cost of the software and supporting hardware, but the savings in labor usually offsets the extra costs.

The issue of using LISP or C (for example) as the underlying ES programming language offers a confusing set of alternatives. Four observations are made. One, the compiled language (C) will fire rules faster than the interpreted language (LISP). Two, the use of a compiled language often requires that an external editor be used on the rules, eliminating the benefits of an interpreted development environment. This is because the modified code must first be compiled before the entire set of modules must be linked into an executable form. This development cycle is very time-consuming and often lacks the dynamic interfaces mentioned earlier. Three, for real-time applications, LISP has the distinct disadvantage of requiring garbage collection (i.e., reclamation of dynamically managed memory) which can cause long, unscheduled delays in processing. Four, it is likely that C will be easier than LISP to link to the DBMS.

4.2 DATABASE MANAGEMENT SYSTEM SOFTWARE SPECIFICATIONS

The DBMS component of CAP at least equals the ES component in functional importance. Section 2 describes the functional responsibility of the DBMS in general terms and also describes the integration of the DBMS with other system components. This section discusses certain issues that arise when evaluating DBMS software. The issues are prioritized in light of the CAP system requirements.

Appendix B contains a detailed evaluation checklist for DBMS software (Huntley 1986) and includes both general and technical considerations.

General considerations are: installation, documentation, system integration, system portability, and vendor stability. The technical considerations include: the relational operators, the data manipulation language (DML), report generation capabilities, application development, host language interface, size limitations and memory requirements, and logical and physical database organization.

The first DBMS requirement considered is the compatibility necessary for system integration. If the CAP prototype is implemented on a single computer, the DBMS and ES must share the same operating system. Also, the two systems must share an external language interface that will be used for the I/O traffic between the two systems.

From the point of view of the users of the CAP DBMS, the most important evaluation issues are related to system productivity. System productivity is related to system flexibility and "user friendliness" and considers the ease of system use, ease of system modification, and the productivity of application generation. These issues are addressed by emphasizing the following DBMS specifications: use of a relational-model based system (as opposed to a hierarchial- or network-based system), inclusion of the Structured Query Language (SQL) standard, and the availability of a fourth-generation language (4GL) for application generation. SQL is a procedural language for database access and is quickly becoming an industry standard. Thus time spent learning SQL, which is quite powerful, is a good investment. The 4GL's are touted as high-level languages for "specifying what to do, not how to do it" and are claimed to reduce the time and effort required to create applications built around a DBMS.

An attractive addition to SQL that some systems offer is called Query by Example (QBE). While SQL is statement oriented using Boolean operators, QBE involves indicating the query form under each field descriptor in a skeleton table. QBE is easy to use and requires less training.

There are a number of the DBMS considerations outlined in Appendix B that do not require emphasis because the environment of the CAP prototype DBMS is not a particularly demanding one. The volume of input data records will be manageable and the record formats will be relatively simple. There will be a minimal amount of user update of the data, and as single updates are not considered critical, features related to system integrity (e.g., recovery) are not emphasized. There will be a small number of users, at least concurrently, and there should be no security problems. Finally, distributed database capabilities are not being stressed at this time. All of these CAP DBMS characteristics simplify selection and implementation of the DBMS software.

5. EVALUATIONS OF SOFTWARE AND HARDWARE CONFIGURATIONS

This section uses the specifications developed in the previous section as criteria to evaluate software packages and software/hardware configurations for use in the prototype system. First addressed are potential software alternatives for the expert system module. Next, software for the database management system is considered. Finally, the expert system and DBMS recommendations are teamed with various hardware options and evaluated as complete configurations, from development to delivery.

Table 5.1 lists seven possible software alternatives for the expert system module. Each alternative either encompasses Rete-based forward chaining capability or allows a programmer to write his own Rete algorithm. The list was designed to include the most well-known and tested major software packages as well as software which requires extensive programming input. For example, ART represents the former whereas Common LISP represents the latter.

Tables in Appendix C contain descriptions of the attributes of the software alternatives. From these tables, the interested reader can learn that the major packages possess better developmental environments but cost more than the OPS and Common LISP alternatives. Information on processing speed, hardware environments, and the availability of source code is also given.

The general recommendation is that the expert system module of the prototype system be developed using a major software package. The

Table 5.1. Expert System Software Evaluation*

Software+	Class	Ranking
ART	Major package	1
Knowledge Craft	Major package	3
OPS5/Common LISP	Shell	5
OPS83	Shell	5
CLIPS	Shell	7
OPS5/DEC	Shell	8
Common LISP	Language	10

*1 is the highest rating, 10 the lowest.

+KEE, a major package marketed by IntelliCorp, is not rated here because it does not have Rete-based, forward chaining capabilities. Instead, it provides object-oriented programming capabilities.

major reason is that these packages provide a better development environment. Tens if not hundreds of man-years have gone into providing the user with special windowing capabilities, graphics, and other programming aids as well as into developing the basic artificial intelligence technologies such as forward chaining, knowledge representation, and the maintenance and control of facts in dymanic working memories. Given that the prototype project's foremost goal is to develop a proof-of-concept system, it is most prudent to build upon past efforts as much as possible. Then in the future, after the proof-of concept has been established, decisions about programming a streamlined system from scratch may make some sense.

At this time, the ART package appears more suited to the prototype's needs than Knowledge Craft. It's forward chaining rule syntax is very expressive and powerful and its inference engine is well integrated with other system components. Another reason is that ART's logical dependency scheme makes keeping track of the relationships between alarms and and conclusions a less difficult proposition. ART also has some limited backward chaining capabilities and a means to assign rule priorities. Lastly, ART is due to appear in a C language version, which would make it faster than the LISP version, easier to integrate with other software components, and more portable with respect to hardware. Review of the commercially available expert system products did not uncover any serious competitors to ART or Knowledge Craft nor indicate that any competitors would surface in the near term.

The review of DBMSs was comparatively uneventful. Appendix D contains four tables describing the attributes of various DBMSs. They were reviewed for being able to meet the prototypes's modest record processing requirements, host a small number of concurrent users, support the SQL framework, generate applications automatically, run on PCs, and provide a user-friendly interface. All the alternatives in Appendix D meet these criteria. Discussions with staff at ORNL who are DBMS experts further indicated that even more packages would meet the minimum criteria.

Our general recommendation is to simply find a satisfactory DBMS. First, Bonneville or the prototype system developer may have suitable DBMS software available already. These systems should be reviewed and possibly adopted because there would be a savings of cost and in-house experience would also be resident. If no in-house DBMS appears preferable, then price would be a major consideration. For example, INGRES has the lowest cost of the four DBMSs listed in Appendix D. The only constraints would be the system's ability to communicate with the expert system and, if located on the same computer, have the same operating system as the expert system software.

Table 5.2 presents four possible software/hardware configurations that encompass both the development and delivery environments. The number of configurations is small because (1) only ART is proposed as the expert system development software, (2) the choice of the DBMS is noncritical, (3) the expert system should be developed on an AI workstation (e.g., a Symbolics), and (4) the tendency is to propose that the delivery hardware be a general purpose workstation.

Table 5.2. Hardware/Software Configuration Evaluations

Expert System Software	Development Hardware	Database Management System Software	Development Hardware	Delivery Hardware	Ranking
ART	AI workstation	Reasonable commercial system	PC or Mini	General use workstation	Preferred
ART	General use workstation	Reasonable commercial system	PC or Mini	General use workstation	Satisfactory
ART	AI workstation	Reasonable commercial system	General use workstation	General use workstation	Satisfactory
ART	AI workstation	Reasonable commercial system	Mini/Main frame	AI workstation and mini/main frame	Possibility

Point (3) is made because the AI workstations coalesce with ART to provide the most powerful expert system development environment. Our own experience and experience reported in the literature suggest that programmer productivity increases by 2 to 4 factors from using Common LISP, on a VAX for example, to using a major package on an AI workstation.

Point (4) is made because discussions with Bonneville staff have indicated that the delivered system should not interfere in any way with existing dedicated Bonneville computer systems. The best way to synthesize the expert system with a DBMS would be on one general purpose workstation (e.g., a MicroVAX).

The preferred option is to develop the expert system on an AI workstation, develop the DBMS on handy equipment such as a PC or an accessible minicomputer, and deliver the entire system on a general purpose workstation. The expert system could be developed on a general purpose machine, but at a loss of development power, which may increase labor costs over the life of the project. It may be wise to develop the expert system on a machine such as the Symbolics because the general purpose machine may end up being too slow for both the expert system and DBMS. Thus, the fourth configuration is listed as a possibility, that the expert system and DBMS may reside on different systems in the delivered environment.

If we had to begin the project today, given the above evaluations, we would recommend that the expert system module be developed using ART on a Symbolics AI workstation that could be networked to a PC where a satisfactory DBMS would reside. Every attempt would be made to deliver the system on a MicroVAX II using ART's soon to be available C version and the proper version of the DBMS.

6. RECOMMENDATIONS AND PROTOTYPE DEVELOPMENT

Artificial intelligence is an important new research and development area within the power systems community. Expert systems that aid in power system planning, design, operation, maintenance, education and training have the potential to significantly improve power system management and reliability. Literature review and problem analysis indicates that the development of an expert system to process Bonneville's Microwave Communication System alarms is eminently feasible. The alarms processing problem matches well with guidelines proposed by well-known expert system developers and is given a high priority in a study conducted by the Electric Power Research Institute.

Success of the prototype expert system is based on properly scoping its application. It is recommended that the prototype system focus on one of seven MCS networks, and address only a subset of all possible alarms and alarm systems. The expert system alarm processor will be linked directly to incoming alarms and directly to a DBMS. The DBMS would archive important alarm data and supply the expert system with data upon request. Operators and Power System Control personnel will have access to the DBMS.

The report also delves into the details of expert system and DBMS software. The critical choice is the expert system development software, which was determined to require a Rete based forward chaining algorithm and a powerful development environment. The DBMS should support the SQL standard, have automatic application generation capabilities, and be user

friendly. It is recommended that a proven major expert system building software package be acquired, with ART developed by Inference Corporation of Los Angeles as the preferred package.

Since many DBMSs meet the requirements of the prototype project, Bonneville is advised to first look in-house for an available package. It is recommended that the expert system be developed on an AI workstation, the DBMS be developed on an available PC or other computer, and that the entire system be delivered on a general purpose workstation.

This section concludes with a few recommendations about how to procede on the prototype project. According to our experience and other experiences documented in related literature, one year should be allowed to develop the prototype system. The first six to eight weeks after familarity with the ES package has been established will entail an intensive effort to build what is known as a rapid prototype expert system. This system will contain maybe 10-20 rules, at most, but will clearly indicate the appropriateness of the software and accompanying inference engine. This exercise will also indicate the ability of the expert to contribute to the project.

The first six months or so should be directly focused on bringing up the expert system software. Problems with expert system facilities for continuous, real-time operation should be resolved in this time frame. The DBMS should be integrated in this time frame only if DBMS data are required by the expert system. The entire package should be ready for Bonneville testing at around ten months. This allows two months to modify the expert system, transfer software, and install equipment on-site at Bonneville.

With respect to the expenditure of manpower resources, expert system building teams are typically small. It is recommended that the team be limited to five individuals, including a microwave communications expert. One and one-half FTEs should be sufficient to cover a project manager, an AI theory expert, a knowledge engineer, and a programmer. Time commitments of the expert may run around 10 to 20% over the year, with most of the effort coming at the beginning.

REFERENCES

Amelink, H., A. M. Forte, and R. P. Guberman, 1986. "Dispatcher Alarm and Message Processing," IEEE Transactions on Power Systems, 1(3), 188-194.

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

Buchanan, B., 1986. "Expert Systems: Working Systems and the Research Literature," Expert Systems 3(1): 32-53.

Buchanan, B. and E. Shortliffe, 1984. Rule-Based Expert Systems: The MYCIN Experiments at the Stanford Heuristic Programming Project, Addison-Wesley, Reading, Massachusetts.

Deering, M. F., 1984. "Hardware and Software Architectures for Efficient AI," in Proceedings of the National Conference on Artificial Intelligence, 73-78.

Dress, W. B., 1986. "A Real-Time Expert System Language for Engineering Applications," in IEEE Proceedings of the Eighteenth Southeastern Symposium on System Theory, 290-293.

Electric Power Research Institute, 1985. Functional Specifications for AI Software Tools for Electric Power Applications, NP-4141, Palo Alto, California.

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

Ennis, R. L., et al., 1986. "A Continuous Real-Time Expert System for Computer Operations," IBM Journal of Research Developments, 30(1), 14-28.

Feigenbaum, E., 1979. "History of Artificial Intelligence Research, 1956-61," IJCAI-79, 1103-1105.

Feigenbaum, E. and P. McCorduck, 1983. The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World, Addison-Wesley, Reading, Massachusetts.

Forgy, C. L., 1982. "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem," Artificial Intelligence, 19(1), 17-37.

Forgy, C. L., 1984. "Initial Assessment of Architectures for Production Systems," in Proceedings of the National Conference on Artificial Intelligence, 116-120.

Forgy, C. L., 1987. "Rete: A Fast Match Algorithm," AI Expert, 2(1), 34-40.

Fujiwara, R., T. Sakaguchi, Y. Kohno, and H. Suzuki, 1985. "An Intelligent Load Flow Engine for Power System Planning," IEEE-PICA, 236-241.

Harmon, P. and D. King, 1985. Expert Systems, Artificial Intelligence in Business, John Wiley & Sons, New York.

Huntley, A. F., 1986. Computer Software Evaluation Methodology and Data Base Management System Selection, ORNL-6260, Oak Ridge National Laboratory, Oak Ridge, Tennessee.

Kline, P. J. and S. B. Dolins, 1985. "Choosing Architectures for Expert Systems," Texas Instruments Inc., Dallas, Texas.

Kline, P. J. and S. B. Dolins, 1986. "Problem Features That Influence the Design of Expert Systems," AAAI-86, 956-962.

Koch, C., B. Isle, and A. Butler, 1987. "Intelligent User Interface for Expert Systems Applied to Power Plant Maintenance and TroubleShooting," IEEE/PES, Winter Meeting, New Orleans, Louisiana.

Kunz, D., B. Tonn, and G. Hosack, 1987. "Knowledge-Based Systems and Large Public Organizations: An Application to Navy Budget Analysis," ORNL-6360, Oak Ridge National Laboratory, Oak Ridge, Tennessee.

Liu, C. and K. TomSovic, 1985. "An Expert System Assisting Decision-Making of Reactive Power/Voltage Control," IEEE-PICA, 242-248.

McDermott, J., A. Newell, and J. Moore, 1978. "The Efficiency of Certain Production System Implementations," Pattern-Directed Inference Systems, D. A. Waterman and F. Hayes-Roth, editors, Academic Press, Inc., New York.

Newell, A. and H. Simon, 1972. Human Problem Solving, Prentice-Hall, Englewood Cliffs, New Jersey.

Schindler, M., 1986. "Tool Picks Up Pace of Expert Systems," Electronic Design.

Schor, M. I., et al., 1986. "Advances In Rete Pattern Matching," AAAI-86, 226-232.

Siemens, R., M. Golden, and J. Ferguson, 1986. "StarPlan II: Evolution of an Expert System," AAAI-86, 844-850.

Stefik, N., et al., 1982. "The Organization of Expert Systems, A Tutorial," Artificial Intelligence, 18, 135-173.

Talukdar, S., E. Cardozo, and T. Perry, 1985. "The Operator's Assistant - An Intelligent, Expandable Program for Power System Trouble Analysis," IEEE-PICA, 255-260.

Tonn, B. and R. Uppuluri, 1986. "Uncertainty Paradigms Used in Expert Systems," Proceeding of the Colloquia of the International Symposium on Methodologies for Intelligent Systems, Knoxville, Tennessee, 59-68.

Waterman, D., 1985. A Guide to Expert Systems, Addison-Wesley, Reading, Massachusetts.

Waterman, D., 1986. "How Expert Systems Differ from Conventional Programs," Expert Systems 3(1), 16-21.

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

APPENDIX A. PRODUCTION SYSTEM SPEED IMPROVEMENTS THROUGH ADVANCED HARDWARE

At some point in the not-so-distant future, significant ES processing speed improvements through software enhancement may not be possible. Therefore, although this project will likely be implemented on conventional hardware, it is prudent to begin investigations of specialized hardware support for expert systems operations (Forgy et al., 1984; Derring 1984).

One promising area of research for PS efficiency improvement is parallel processing. Three types of Rete algorithm parallelism are possible: production-level, condition-level, and action-level. Of the three, production-level parallelism is the most promising. Here, the productions of the system are divided among several processors which can perform the match process in parallel. In the extreme case, each processor is responsible for only one production. The changes to working memory would have to be communicated to all of the processors but the amount of communcations should be small, allowing for various system architectures to be considered (e.g., nonshared memory). The limiting factor to the amount of speed-up possible is the processor which has the longest task. On each cycle, the faster processors would have to wait for the slowest to finish.

Other speed-up opportunities exist through the use of specialized hardware. Forgy concludes that because complex instruction sets and addressing modes are typically not needed for forward chaining PSs, a reduced instruction set architecture is desirable simply to execute instructions in fewer machine cycles. Similar in concept is work at

the Oak Ridge National Laboratory on a FORTH-based version of OPS5. An implementation has demonstrated significant speed-up over traditional OPS5 software (Dress 1986). The next step is to match the software to a specialized FORTH processor where each machine instruction is a FORTH instruction.

APPENDIX B. DATABASE MANAGEMENT SYSTEMS EVALUATION CHECKLIST

I. General Considerations

A. Installation

1. Procedures fully documented
2. Installation options modifiable without reinstallation

B. Documentation

1. Comprehensive and well organized
2. Standardized notation
3. Leveled documentation
   a. Executive summary
   b. User's guide
   c. Technical reference
4. Documentation contains many examples, with actual code listings
5. Extensive indices
6. Quick reference card
7. On-line tutorials
8. Context-sensitive help
9. Error message clarity

C. System Integration

1. Internal consistency of the complete system
2. Consistent command syntax and availability between modules
3. Logic of data manipulation language (DML) commands
4. User specified DML program loops for record at a time processing
5. Integrity constraints used by all modules
6. Efficiency of computer hardware/software usage
7. Logical, standardized use of keyboard function keys
8. Operating system commands available within the DBMS
9. Able to run other software from within DBMS
10. Hardware/operating system imposed limitations

D. System Portability

1. Successful hardware implementations
2. Adherence to industry standards

E. Vendor and Product Stability

1. Software development history
2. Vendor financial stability
3. Error-free software releases

4. Frequency of system releases, extent of modifications, user impacts
5. Discontinued support of system modules
6. Vendor commitment to customer support
7. Technical assistance available by telephone
8. Variety of training classes

II. Technical Considerations

A. Relational Operators

1. Support the eight relational operators
   a. select, project, join, divide, union, intersection, difference, and Cartesian product
2. Restrictions and limitation on use of the operators
3. Support comparisons based upon equality, inequality, a range of values, or multiple discrete values
4. Semantic override of operator restrictions available

B. Data Manipulation Language

1. Compilation of DML statements
2. Mathematical and statistical functions
3. Date manipulation
4. Data field comparison based on character substring
5. Data field comparison using wildcard characters
6. Editing capabilities for both data field values and DML statements
7. DML uses logical command syntax
8. Full DML available for use from high-level language applications programs, with no change in command syntax
9. Multifield sort of selected records, ascending or descending order
10. Count of records retrieved available
11. Dynamic table creation allows iterative narrowing of record selection criteria without new search through entire database
12. Can store query templates

C. Report Generation Capabilities

1. Modifiable display format, without additional data retrieval

   a. Report titles
   b. Multi-line customized column headers
   c. Column width changes

d. Break points
e. Reformat data field values
f. Perform computations on selected records

D. Application Development Subsystem

1. Support for multi-screen applications
2. Data field validation
3. Programmable context specific help and error messages
4. Single screen display of data from multiple tables
5. Single screen display of multiple data records

E. Host Language Interface

1. Languages for which interfaces are available
2. Full DML available from host language, no change in command syntax
3. Graphics package interface

F. Database Size Limitations and Memory Requirements

1. Maximum data field length
2. Maximum data record length
3. Number of fields allowed per table
4. Number of records per table
5. Number of tables per database
6. Memory requirements (single and multiple users)
7. Database storage efficiency
8. Index and data table compression usage

G. Database Loading, and Import/Export Facilities

1. Facilities for bulk loading of partial or complete data records
2. Database export of standard format files available, for partial or complete databases
3. Archiving of records to new file, or append to existing file

APPENDIX C. EXPERT SYSTEM SOFTWARE DESCRIPTIONS

OPS5, Digital Equipment Corporation

Expert System Software Attributes

Runtime performance	Fair
Development environment	
editor, incremental compile	Limited
editor, facts and rules	Limited
windows	No
step, trace, debug	Yes
statistics	Yes
Flex. of knowledge representation	Limited, facts
Flex. of inference techniques	Limited, forward chaining
Flex. of control techniques	Limited, two types of conflict resolution, no priorities
Hardware	VAX
Operating system	VMS
Underlying language	BLISS-32
External language	Any, using Common Calling Standard
Cost*	
initial	$8,800
Maintenance	3,000
Source code available	No

*List prices are shown. Actual cost may be significantly less due to GSA price schedules, multiple copy price schedule, and competitive bidding.

OPS5, Common LISP

Expert System Software Attributes

Runtime performance..... Poor

Development environment
- editor, incremental compile..... Limited
- editor, facts and rules..... Limited
- windows..... No
- step, trace, debug..... Yes
- statistics..... Yes

Flex. of knowledge representation... Limited, facts

Flex. of inference techniques..... Limited, two types of conflict resolution, no priorities

Flex. of control techniques..... Limited, but source can be modified

Hardware..... Any

Operating system..... Any

Underlying language..... LISP

External language..... LISP

Cost
- initial $0
- maintenance 0

Source code available ..... Yes

OPS83

Expert System Software Attributes

Runtime performance..... Good

Development environment
  editor, incremental compile..... No
  editor, facts and rules..... No
  windows..... No
  step, trace, debug..... Yes
  statistics..... Yes

Flex. of knowledge representation... Limited, facts

Flex. of inference techniques..... Limited, forward chaining

Flex. of control techniques..... Yes, full control of conflict resolution

Hardware..... VAX, PC, ...

Operating system..... VMS, MS/DOS, ...

Underlying language..... C

External language..... Any

Cost*
  initial..... $9,000 (VAX), $2000 (PC)
  maintenance 1,000

Source code available..... No

*List prices are shown. Actual cost may be significantly less due to GSA price schedules, multiple copy price schedule, and competitive bidding.

CLIPS

Expert System Software Attributes

Runtime performance..... Fair

Development environment
  editor, incremental compile..... No
  editor, facts and rules..... No
  windows..... No
  step, trace, debug..... No
  statistics..... No

Flex. of knowledge representation... Limited, facts

Flex. of inference techniques..... Limited, forward chaining

Flex. of control techniques..... No

Hardware..... Any

Operating system..... Any

Underlying language..... C

External language..... C

Cost
  initial..... $200
  maintenance ..... -

Source code available..... Yes

KNOWLDEGE CRAFT

Expert System Software Attributes

Runtime performance.....	Fair
Development environment	
editor, incremental compile.....	Yes
editor, facts and rules.....	Yes
windows.....	Yes
step, trace, debug.....	Yes
statistics.....	Yes
Flex. of knowledge representation...	Yes, facts, schemata
Flex. of inference techniques.....	Yes, forward chaining and backward chaining
Flex. of control techniques.....	Yes
Hardware.....	Symbolics, VAX,...
Operating system.....	Symbolics, VMS,...
Underlying language.....	LISP
External language.....	LISP
Cost*	
initial.....	$60,000
maintenance	4,000
Source code available.....	No

*List prices are shown. Actual cost may be significantly less due to GSA price schedules, multiple copy price schedule, and competitive bidding.

ART

Expert System Software Attributes

Runtime performance..... Fair

Development environment
- editor, incremental compile..... Yes
- editor, facts and rules..... Yes
- windows..... Yes
- step, trace, debug..... Yes
- statistics..... Yes

Flex. of knowledge representation... Yes, frames, inheritance, temporal reasoning

Flex. of inference techniques..... Yes, forward and limited backward chaining

Flex. of control techniques..... Some, rule priorities

Hardware..... Symbolics, VAX*

Operating system..... Symbolics, VMS*

Underlying language..... LISP

External language..... LISP

Cost**
- initial..... $60,000
- maintenance 7,800

Source code available..... No

*A C-based version of ART has been released but not evaluated. The C language replaces LISP yet the development environment is unchanged. The C version should be faster and easier to deliver.

**List prices are shown. Actual cost may be significantly less due to GSA price schedules, multiple copy price schedule, and competitive bidding.

APPENDIX D. DATABASE MANAGEMENT SYSTEM SOFTWARE DESCRIPTIONS

ORACLE

DBMS Software Attributes

Operating Systems (MicroVAX)	
VMS	Yes
ULTRIX	Beta
PC version	Yes
Interface Languages	
C	Yes
Other	Yes
4-GL Features	
Interactive forms designer	Yes
Interactive dictionary designer	Yes
Report writer	Yes
Graphics	Yes
Statistical package	No
Other Features	
Online help	Yes
Menu-driven	No
Maximum Number Of	
Records	Unlimited
Fields per record	254
Index keys per file	Unlimited
Field types	5
Date	Yes
Date-time	Yes
Characters per field	240
Cost (MicroVAX)*	
SQL	$11,000
4-GL	4,000
Language interface	600
Maintenance	3,200

*List prices are shown. Actual cost may be significantly less due to GSA price schedules, multiple copy price schedule, and competitive bidding.

INFORMIX

DBMS Software Attributes

Operating Systems (MicroVAX)	
VMS	Yes
ULTRIX	Yes
PC version	Yes
Interface Languages	
C	TBA
Other	No
4-GL Features	
Interactive forms designer	Yes
Interactive dictionary designer	No
Report writer	Yes
Graphics	No
Statistical package	No
Other Features	
Online help	Yes
Menu-driven	Yes
Maximum Number Of	
Records	Unlimited
Fields per record	Unlimited
Index keys per file	Unlimited
Field types	8
Date	Yes
Date-time	No
Characters per field	Unlimited
Cost (MicroVAX)*	
SQL	$3,750
4-GL	included
Language interface	2,750
Maintenance	1,150

*List prices are shown. Actual cost may be significantly less due to GSA price schedules, multiple copy price schedule, and competitive bidding.

ACCELL

DBMS Software Attributes

Operating Systems (MicroVAX)	
VMS	No
ULTRIX	Yes
PC version	Yes
Interface Languages	
C	Yes
Other	Yes
4-GL Features	
Interactive forms designer	Yes
Interactive dictionary designer	Yes
Report writer	Yes
Graphics	No
Statistical package	No
Other Features	
Online help	
Menu-driven	
Maximum Number Of	
Records	2 billion
Fields per record	256
Index keys per file	8
Field types	7
Date	Yes
Date-time	Yes
Characters per field	256
Cost (MicroVAX)*	
SQL	$9,500
4-GL	included
Language interface	included
Maintenance	2,000

*List prices are shown. Actual cost may be significantly less due to GSA price schedules, multiple copy price schedule, and competitive bidding.

INGRES

DBMS Software Attributes

Operating Systems (MicroVAX)
- VMS..... Yes
- ULTRIX..... Yes

PC version..... TBA

Interface Languages
- C..... Yes
- Other..... Yes

4-GL Features
- Interactive forms designer..... Yes
- Interactive dictionary designer... Yes
- Report writer..... Yes
- Graphics..... Yes
- Statistical package..... No

Other Features
- Online help..... Yes
- Menu-driven..... Yes

Maximum Number Of
- Records..... Unlimited
- Fields per record..... 127
- Index keys per file..... Unlimited
- Field types..... 12
  - Date..... Yes
  - Date-time..... Yes
- Characters per field..... 2,000

Cost (MicroVAX)*
- SQL..... $5,000
- 4-GL..... -
- Language interface..... included
- Maintenance..... 2,500; 1,250 2nd year

*List prices are shown. Actual cost may be significantly less due to GSA price schedules, multiple copy price schedule, and competitive bidding.

INTERNAL DISTRIBUTION

1. L. F. Arrowood
2. R. S. Carlsmith
3. K. R. Carr
4. M. L. Emrich
5. D. M. Flanagan
6. W. Fulkerson
7. P. A. Gnadt
8. R. T. Goeltz
9. J. L. Hardee
10. M. L. Hilliard
11. E. L. Hillsman
12. E. Hirst
13. R. B. Honea
14. H. Hwang
15. M. A. Kuliasha
16. G. E. Liepins
17. A. S. Loebl
18. F. C. Maienschein
19. B. W. McConnell
20. M. S. Phifer
21. S. L. Purucker
22. J. H. Reed
23. D. T. Rizy
24. R. A. Stevens
25. B. E. Tonn
26. E. W. Whitfield
27. T. J. Wilbanks
28. Document Reference Section
29. Central Research Library
30-31. Laboratory Records Department
32. Laboratory Records Department - RC
33. ORNL Patent Section

EXTERNAL DISTRIBUTION

34. Office of the Assistant Manager for Energy Research and Development, DOE-ORO
35-65. Technical Information Center, DOE, P. O. Box 62, Oak Ridge, TN 37831
66. J. G. Carbonell, Associate Professor of Computer Science, Carnegie-Mellon University, Pittsburg, PA 15213
67. F. R. Kalhammer, Vice President, Electric Power Research Institute, P. O. Box 10412, Palo Alto, CA 94303
68. R. E. Kasperson, Professor, Government and Geography, Graduate School of Geography, Clark University, Worcester, MA 01610
69. R. L. Perrine, Professor, Engineering and Applied Sciences, Civil Engineering Department, Engineering I, Room 2066, University of California, Los Angeles, CA 90024
70-540. Decision Systems Research Section External Mailing List and Extra Copies to M. S. Phifer, 4500N, 207