NASA Contractor Report 185263

Autonomous Power Expert System

Mark J. Ringer and Todd M. Quinn
Sverdrup Technology, Inc.
Lewis Research Center Group
Brook Park, Ohio

July 1990

Prepared for
Lewis Research Center
Under Contract NAS3-25266

National Aeronautics and
Space Administration

(NASA-CR-185263) AUTONOMOUS POWER EXPERT SYSTEM Final Report (Sverdrup Technology) 8 p
CSCL 21H
N90-25187
Unclas
G3/20 0293310

E-5609

Autonomous Power Expert System

Mark J. Ringer and Todd M. Quinn
Sverdrup Technology, Inc.
Lewis Research Center Group
Brook Park, Ohio 44142

ABSTRACT

The goal of the Autonomous Power System (APS) program is to develop and apply intelligent problem solving and control technologies to the Space Station Freedom Electrical Power System (SSF/EPS). The objectives of the program are to establish artificial intelligence/expert system technology paths, to create knowledge-based tools with advanced human-operator interfaces, and to integrate and interface knowledge-based and conventional control schemes [1]. This program is being developed at the NASA Lewis Research Center in Cleveland, Ohio.

The APS Brassboard represents a subset of a 20KHz Space Station Power Management And Distribution (PMAD) testbed. A distributed control scheme is used to manage multiple levels of computers and switchgear. The brassboard is comprised of a set of intelligent switchgear used to effectively switch power from the sources to the loads.

The Autonomous Power EXpert System (APEX) portion of the APS program integrates a knowledge-based fault diagnostic system, a power resource scheduler, and an interface to the APS Brassboard. The system includes knowledge bases for system diagnostics, fault detection and isolation, and recommended actions.

The scheduler autonomously assigns start times to the attached loads based on temporal and power constraints. The scheduler is able to work in a near real-time environment for both scheduling and dynamic replanning.

INTRODUCTION

Since the Space Station will be continuously operational with humans on-board, its power system must be able to work continuously without a major interuption of power. Many ground based operators controlling the Space Station's power system will be necessary in order to accomplish this task. This will be a very cumbersome and expensive function, therefore the control should be performed on-board in order to minimize the cost and increase reliability. This on-board control strategy should be used for both fault diagnosis and recovery along with planning and scheduling. The APS project incorporates automated fault detection and recovery with autonomous planning and scheduling into a hardware power system brassboard. In concert with the APS program, a comprehensive automation program is being developed at Lewis for the Space Station Freedom Electrical Power System [2][3].

The APS Brassboard is monitored by the APEX system software. The brassboard consists of two power sources, two embedded control computers, and a set of intelligent switchgear. Simple resistive loads are controlled through the switchgear by two single-board computers with embedded Ada software. The switchgear is able to configure into various arrangements using multiple channels in order to efficiently and effectively distribute power from the sources to the loads. Various hardware faults can be induced into the system and in turn be diagnosed by the expert system.

The APEX system software is designed to emulate human expert thought in order to assess the state of the APS Brassboard. Fault diagnosis and recovery are performed by APEX using the information gathered on the state of the system. The APEX system consists of a rule-based fault diagnostic expert system with interfaces to both a load scheduler and the APS Brassboard (Hardware). The fault diagnostic system includes the ability for detection, isolation, justification, and recommended action for both conventional and incipient faults. Load Configuration and state information is obtained from the remote load planner/scheduler. Dynamic replanning is also performed if a new load set or hardware structure is encountered in the event of an anomoly.

The APS Scheduler will autonomously schedule or reschedule the start time of each load on the APS Brassboard in a near-real time environment. The scheduler follows assigned time and power constraints to produce a schedule designed to optimize power use. Dynamic replanning autonomously reschedules and re-optimizes the load set in the case of an anomoly. The schedule generated also provides a timetable for the APEX system to follow.

BRASSBOARD

System Overview

The APS Brassboard is based on an earlier design of the SSF 20KHZ PMAD Testbed. The SSF Testbed has subsequently converted to a DC system [4]. The PMAD Testbed is a distributed power system containing three subsystem control computers: the Power Source Controller (PSC), Main Bus Controller (MBC), and Power Disribution

Controller (PDC) with a Power Management Controller (PMC) performing the executive overview of the system as shown in Figure 1. The PSC controls the solar arrays while the MBC controls the distribution of power from the source to the PDC's. The PDC controls the low-level switching of power to the many associated loads. The APS Brassboard consists of Power Distribution Unit A, a PMC, and a PDC.

The Testbed and brassboard construction is based on a few simple concepts that make for an effective bridge towards design of large space power systems. Intelligence should be embedded as far down as possible: for example, switches can instantaniously control their state if an over-current fault is sensed. Distributed control incorporates many of the low level functions at the component level. This leaves the upper level computers free to make overall decisions about the state of the system and reduces communciation and data bus loading. The power distribution architecture is designed such that multiple paths exist between the various loads and sources. This allows for multiple reconfiguration schemes when recovering from a fault.

Simple resisitive loads (such as lights) are used as well as a variable resistance load bank. The variable resistance load bank is used to introduce incipient faults by slowly changing current levels. Hard faults are wired into the system with switches to control the insertion of the faults.

The APS Brassboard architecture (shown in Figure 2) consists of three RBIs and three RPCs with two step down transfomers. The RBIs operate at 440V while the RPCs operate at 220V. Power is available from two sources simulating the two inputs from the ring bus in Figure 1.

Component Description

The Remote Bus Isolators (RBIs) and Remote Power Controllers (RPCs) are both intelligent switches: the difference being that the RBIs have a solid state and a relay switch while the RPCs have only a solid state switch. The switchgear includes integrated 8 bit A to D data acquisition boards to measure current, voltage, power factor, and various measures of the state of the switch. Embedded hardware/software within each switch controls the trip point, communications, and state of the switch. The switches can, based on embedded software, be set to automatically trip (turn off) at a specified current level. The switches are then able to turn on again when the command is given.

The RPCs can switch off in a matter of a microseconds while the RBIs can switch off in the millisecond range. The RBIs are slower because of the relay switch. In a distributed system such as the brassboard, it is necessary to design the lowest level switches with the fastest trip times. If a fault occurs at the RPC level, it must be sensed and the switch must trip before the higher level RBIs sense the fault. In this way, low level faults do not cascade up the system.

The PMC and The PDC are Intel 8086-based single board computers with the controlling code written in the ADA language [5]. This software was written for the SSF Testbed and has been through some minor modifications in order to run on the APS Brassboard. The PMC, PDC, and APEX computer are interfaced via an Ethernet link. A MIL-STD 1553 bus is used at the lowest level to communicate between the PDC and the switchgear.

APEX

Overview

For very large space based power systems, human monitoring and control will be very difficult and costly because of the complexity of the system. An autonomous power system will be more reliable and less costly in the long run. The APEX system takes the place of a human expert in the control areas that are either extremely repetitious, require a long period of thought on the part of the expert, or when the human expertise has become unavailable.

APEX detects faults by comparing expected electrical and state parameters computed from knowledge of the system configuration and schedule information to the values obtained from the actual APS Brassboard. If no deviations from the expected operating state exist, APEX will again request data from the hardware, and re-initiate fault detection with the new data. If an anomoly is discovered in the system data, APEX will inform the user that a fault has been detected.

Once a fault has been detected, APEX can then be asked to isolate the probable cause. To reach a conclusion on the fault origin, APEX accesses information contained within its knowledge bases. The probable cause of the fault is displayed to the user along with a justification of the reasoning leading to the conclusion. A recommended action is given, based on both system and scheduler information, to reconfigure the system into the best post-fault configuration.

Expert System Implementation

APEX consists of a knowledge base, a data base, an inference engine, and various support/interface software. The knowledge base is composed of a representation of the reasoning patterns which have been found useful to the human expert during his efforts at problem solving. The data base is the basic working area where storage and calculations of numerical data occurs. The inference engine is the reasoning mechanism which draws conclusions from information stored within the knowledge base. Conventional software is also necessary to provide the user with an interactive interface and to obtain data from the hardware and scheduler.

Representation of knowledge within the APEX knowledge base consists mainly of frames, semantic triples and production rules. Frames are structures which describe objects or classes of objects and the relationships between objects. Objects are composed of slots which specify the different attributes belonging to each object. Individual slots of an object can contain either declarative or procedural information. Declarative information expresses facts about the object, whereas procedural information is in the form of a program or a set of procedural steps. Frames themselves are considered to be declarative information. Within APEX, declarative information is also represented by semantic triples which state information in the form of object/attribute/value (i.e. attribute of object = value). Production rules are 'If-Then' statements which infer either declarative or procedural facts when the conditional facts contained in the premises of the rule are found to be true. Again the facts are represented as a semantic triple (declarative) or a program (procedural).

APEX employs an inference engine contained in the

Knowledge Engineering Environment (KEE) expert system shell. The inference engine is the heart of the expert system, determining how knowledge is represented and processed. By operating on rules within the knowledge base, the inference engine can reason and make inferences about the state of the power system. The ways the inference engine processes the rules and data are commonly refered to as forward and backward chaining. Forward chaining (also known as data driven) works from the given data to a conclusion. Backward chaining (also known as goal driven) works from a particular goal and tries to either confirm or refute its truth. In the APEX system, fault detection is implemented with forward chaining and fault isolation is accomplished with backward chaining.

The data base contains a historical record of data acquired from the switching devices in the power distribution system. Storage and manipulation of the historical data is accomplished with conventional techniques and does not require the use of the inference engine. This historical record is kept in order to detect incipient faults.

Fault Detection

Three discrete types of fault can be detected: inconsistency, active, and incipient. Inconsistency faults occur when two or more data values give conflicting results about the state of the power system. Active faults are detected when measured and expected values conflict within limits of a specified tolerance. Multiple active faults can also be detected. Incipient faults are detected by monitoring a history of data values that identify trends toward tolerance limits. Trends are detected by statistical correlation and regression analysis of the historical data. Once a fault is detected, domain specific troubleshooting knowledge is referred to and backward chaining is initiated to isolate faults.

Probable causes for faults are identified by the fault isolation phase. In some cases, more than one probable cause is displayed, these are displayed in order of highest to lowest probability. Justification is obtained from a trace back of the rule firings. These rule firings, written in an expert system shell language, are then translated into a natural language form, giving an explanation of the reasoning process leading to the probable cause conclusions.

A recommended action feature suggests what should be done to correct the fault. The APEX system considers information such as the severity of the fault and priority of the affected loads in recommending an action that should be taken to correct, bypass, or temporarily tolerate the fault.

Interfaces

The user interface enables APEX to communicate with the operator through color graphic display screens and menu selections. The graphic screens can show information to the user at different levels of detail. Using the menu options, the user can select the level of information to be displayed, ask for justification of a particular conclusion, and request any available recommended action to correct an isolated fault. The APEX system is fully mouse activated for quick and easy operation.

A data simulator is included in the APEX system which can be used in place of the APS Brassboard when the software is being tested. For software verification, domain experts set up fault scenarios within the simulator and review the expert systems diagnosis of the faults and recommended actions. A scheduler interface is also included and will be discussed in the next section.

The Testbed data acquisition interface requests pertinent parametric data values from the PDC control computer and asserts new values received into the knowledge base. For incipient fault detection, the data is stored in a First In First Out (FIFO) table that contains the last 200 values for each analog test point on the brassboard.

Currently, the operator reviews the justification and recommended actions and then manually performs the procedures to clear the faults. The next step in the APEX development effort is to communicate recommended actions directly to the subsystem controllers. This would provide for autonomous fault isolation and recovery with a human overseeing the process.

Hardware and software being used for the development of APEX are Texas Instruments Explorer II LX workstations, the Knowledge Engineering Environment (KEE) expert system development shell and common Lisp (List Processing language).

SCHEDULER

Overview

The power availability on Space Station differs from a terrestrial utility because power is in very limited supply. This forces users on Space Station to request a specific amount of power and include constraints which must be met such as temporal considerations and consumables availability. On a ground based system, a switch can just be turned on at anytime in order to receive power. The scheduler must decide which loads receive power and when based on an overall optimizing strategy.

Scheduling loads on Space Station is a very complex process, with thousands of loads, hundreds of resources, and even more load constraints. The scheduling problem will be an extremely difficult and labor intensive task. Scheduling for the Voyager II encounter of Uranus took work-decades of effort [6]. Space Station will be orders of magnitude more complex and will be continously operational. This complexity is compounded by the fact that loads can be scrubbed, fail, or the scope of their work can be changed on short notice. These two facts emphasize the need for an on-board autonomous and adaptive scheduling system.

Scheduling the entire Space Station mission would include scheduling millions of loads over the entire 30 year lifetime. Even if all these loads and available resources were known ahead of time (which they aren't) it would be an impossible task. In order to solve any complex scheduling problem, it must be subdivided into smaller pieces to create a number of smaller feasible problems. The APS scheduler breaks the problem down into 8-hour planning horizons with a minimum event time interval of 5 minutes. Provisions for loads that end after the planning horizon is over are made by allowing loads to continue into the next planning horizon, and loads in the current schedule can continue from the previous horizon.

Figure 3 shows the output of the APS Scheduler. The upper portion displays the power used and available, dark areas represent unused power. On the bottom portion, loads are represented as bars, the arrows show loads continuing from the past or continuing into the next planning horizon, and brackets represent time constraints.

In most spacecraft design problems, the demand for power largely exceeds supply or in scheduling terms is oversubscribed. An excess demand means that power must be given out to the users (loads) in the most efficient manner. This high demand for power actually makes the optimizing process easier because a larger number of feasible schedules exist. This process of scheduling loads can be accomplished by either numerical algorithms or heuristic based shuffling strategies which optimize power use, the number of constraints violated, or some other form of priority based goodness factor. The APEX scheduler uses a numerical search algorithm with a few heuristic rules and optimizes total power use.

Algorithm

Although the overall scheduling problem on Space Station is very complex, the APS scheduling problem is a significantly simpler problem. Because of the relative simplicity of the APS scheduler, an optimum search based solution was chosen. This was the most efficient both in time to design and build as well as execution time for this specific application. This algorithm performs a tree search over portions of the solution space in order to find an "optimum" solution. The true optimum could be found for each case if the entire solution space was searched, but this is not necessary in order to find a reasonably good schedule in a short time. In reality, only an extremely small fraction of the solution space is searched in order to find a solution using in excess of 99% of the available power. Since limits are placed on time allowed to find a solution, all uses of the word optimum will mean the solution found in this limited amount of time. Heuristic based rules are also used to limit the space searched as well as well as place the loads in optimum positions. This tree search is also aided by a branch and bound algorithm which limits the space searched and stops the algorithm from searching through many infeasible schedules.

The algorithm allows for loads with varying power profiles. Temporal constraints can also be specified in order to force the start or end times of certain loads into a specified time period.

The algorithm first performs a depth-first search in order to place as many loads as possible into the schedule in the shortest amount of time. From this original schedule, small perturbations are made in order to find a schedule with more power use. As the scheduling algorithm searches for an optimum, it generates many schedules along the way. Each schedule found that is better than the previous optimum is saved (as the new optimum) and displayed graphically on the screen. In this way, the user can see the progression from the original schedule as the algorithm constantly improves the total power use of the schedule. This scheduling strategy is known as an anytime schedule, because at anytime the scheduler can output a good schedule [7].

The algorithm and user interfaces were written in the C language. The scheduling engine is based on an algorithm developed by DiFilippo [8]. The scheduler can be run as an autonomous server to the APEX computer, or can be run in a stand alone mode in order to explore the workings of the scheduler. The scheduler is fully mouse controlled with an interactive editor to edit the power profiles of each load in the stand alone mode. The scheduling system is run on a stand-alone 386-based PC and linked via ethernet to the APEX computer. This architecture relieves the APEX computer from any worries about scheduling and makes for a transparent scheduling interface.

Implementation

In case of a fault, the scheduler must be able to dynamically replan the remainder of the horizon in order to optimize to the new conditions caused by the change in state of the system. This is very important for fault recovery. The distribution system and/or load states can change after the fault, therefore the new configuration and/or loads must be reconfigured into an optimum condition. An example (refer to Figure 2) would be if load 1 is on, power is flowing through both RBI 3/1 and RPC 3/4 which are on. If RBI 3/1 fails, the system would reconfigure by making sure that RBI 3/1 is off then turning on RBI 3/3 and RBI 3/2 in order to receive power from the other power bus. This reconfiguration decision is developed by both APEX and the scheduler.

The scheduler is meant to work in a near real-time environment therefore, limits must be placed on the amount of time the scheduler has to work. When APEX sends a request to reschedule, it also sends a maximum amount of time the scheduler has to work. This is usually on the order of 5-10 seconds. If the scheduler comes to an optimum before this time, it will send back the completed schedule.

One problem in scheduling for real-time systems that have fault conditions is the fact that the switchgear can trip (turn off) very quickly and leave the scheduler to clean up the mess. The switchgear must turn off in a matter of microseconds in order to protect the system from overcurrent conditions which could permanently damage the system. This is quite different than most thoughts that a scheduler has ultimate control over all events.

CONCLUSION

Because of the complexity and critical nature of the Space Station Power System, autonomous control would provide a more reliable and less costly system. The Autonomous Power System project serves as a bridge towards accomplishing this goal. APEX is able to diagnose and give recommended actions for faults on the APS Brassboard. This is soon to be augmented by implementation of a closed loop control strategy, where APEX actually reconfigures the system autonomously when an anomoly is detected.

The APS Scheduler is able to autonomously assign starting times to loads in a near-real time environment based on time and power constraints. It also works in concert with APEX to configure the system and replan when a fault is detected.

The APS Brassboard is a simple representation of a power distribution system and future enhancements to increase system complexity are planned. A more complex architecture would allow for a more realistic system as well as more fault and reconfiguration scenarios.

ACKNOWLEDGEMENTS

The authors would like to thank Gene Liberman for his work on Ada software, Walt Krawczonek and Tony Merolla for their work on the brassboard hardware. The expert

system was modelled after knowledge gained from Walt Krawczonek our power system "expert". Jim Kish provided exceptional guidance both technically and managerially.

This work was performed under contract NAS3-25266 with Jim Kish as Coordinator.

REFERENCES

[1] J.L. Walters, T.M. Quinn, et al, "Autonomous Power Expert System", in proc. 1990 Goddard Conference on Space Apllications of Artificial Intelligence.

[2] J.L. Dolce, J.A. Kish, and P.A. Mellor, "Automated Electric Power Management and Control for Space Station Freedom", in Proc. 25th Intersociety Energy Conversion Engineering Conference, AIChe 1990.

[3] P.A. Mellor, J.L. Dolce, and J.C. Krupp, "Electrical Power Scheduling: a Distributed Problem-Solving Approach for Space Station Freedom", in Proc. 25th IECEC 1990.

[4] R.F. Beach, G.L. Kimnach, T.A. Jett, L.M. Trash, "Evaluation of Power Control Concepts using the PMAD Systems Testbed.", Proc. 25th IECEC 1990.

[5] T. Wright, M. Mackin, and D. Gantose, "Development of Ada Language Control Software for the NASA Power Management and Distribution Test Bed", Proc. 25th IECEC 1990.

[6] E. Biefeld, and L. Cooper, "Scheduling with Chronology-Directed Search", AIAA Computers in Aerospace VII Conference and Exhibit, Oct. 3-5, 1989.

[7] M. Zweben, M. Deale, and M. Eskey, "Anytime Rescheduling", NASA Ames Artificial Intelligence Branch Technical Report.

[8] D.M. DiFilippo, "Space Power System Scheduling", M.S. Thesis, Case Western Reserve University, January 1989.

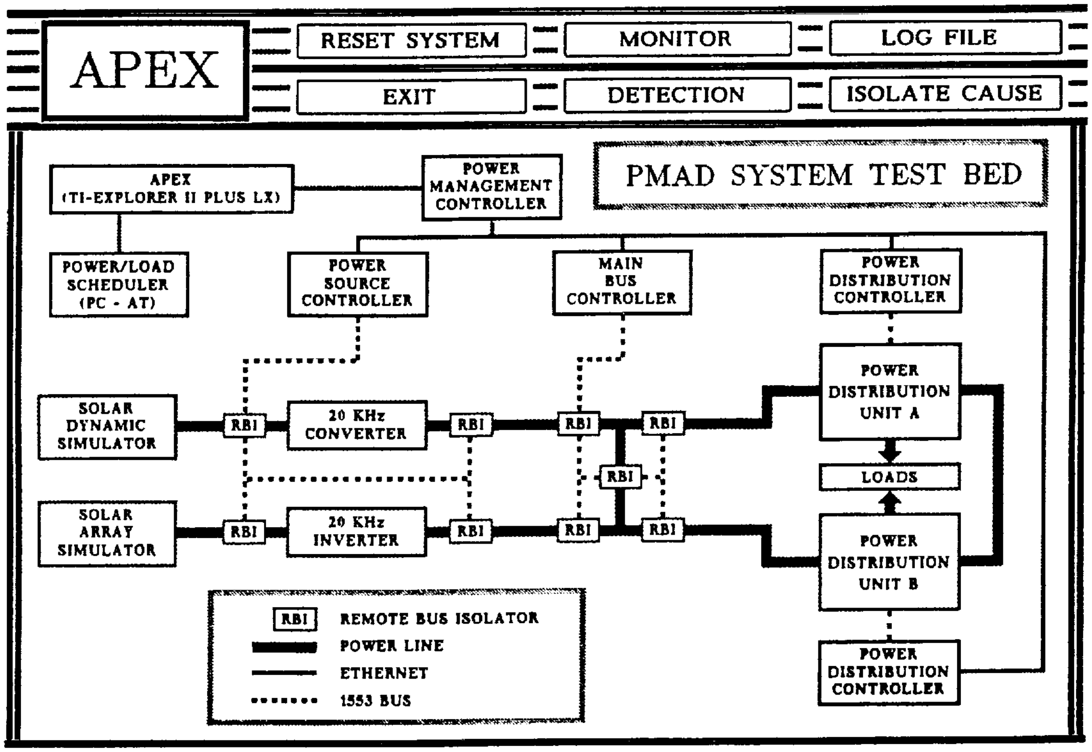

Figure 1. Space Station Testbed Configuration

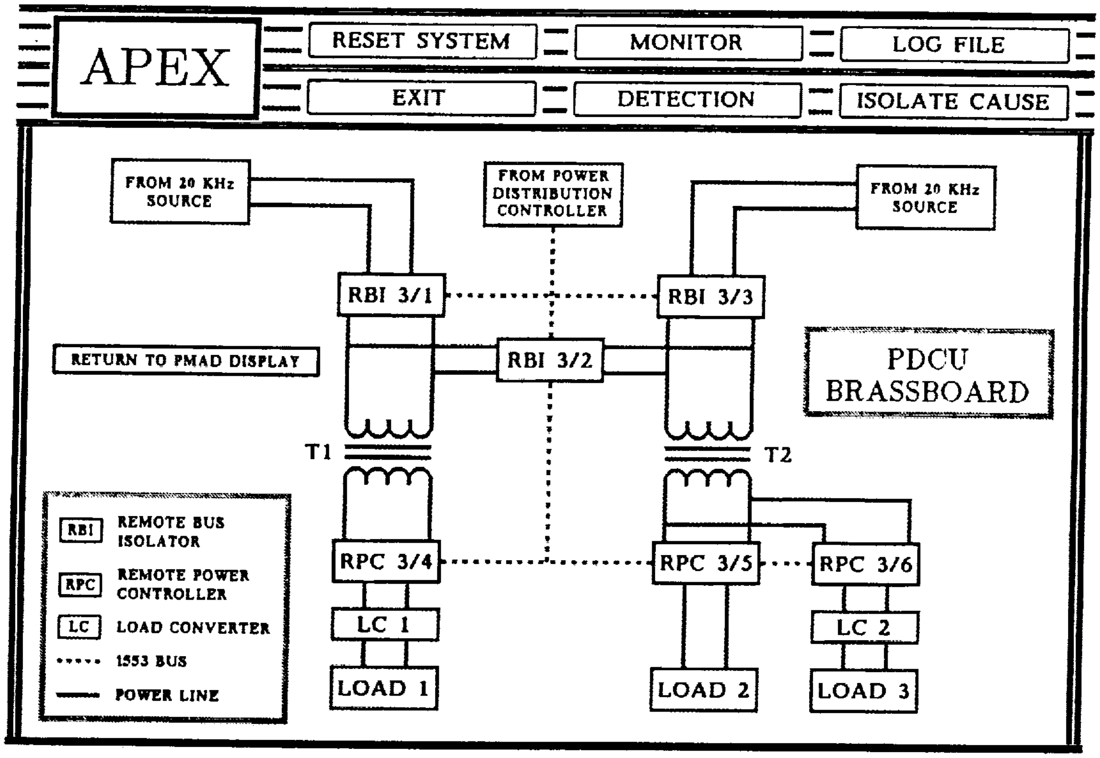

Figure 2. Autonomous Power System Brassboard Configuration

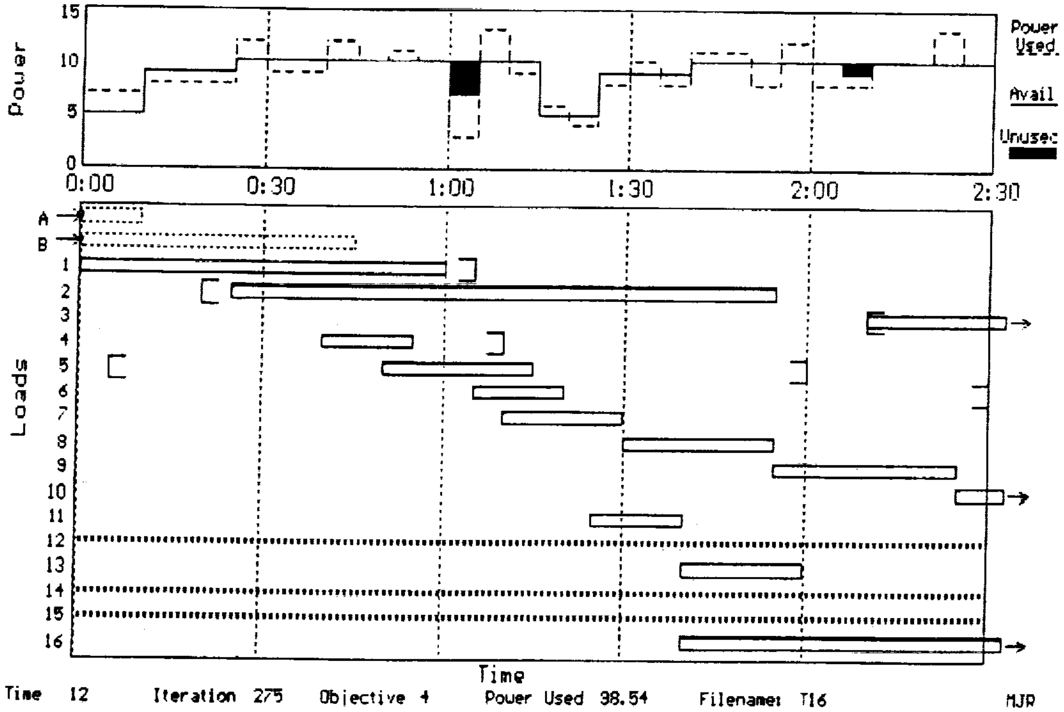

Figure 3. Scheduler Output

NASA
National Aeronautics and
Space Administration

Report Documentation Page

1. Report No.
NASA CR-185263

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Autonomous Power Expert System

5. Report Date
July 1990

6. Performing Organization Code

7. Author(s)
Mark J. Ringer and Todd M. Quinn

8. Performing Organization Report No.
None (E-5609)

10. Work Unit No.
590-12-31

9. Performing Organization Name and Address
Sverdrup Technology, Inc.
Lewis Research Center Group
2001 Aerospace Parkway
Brook Park, Ohio 44142

11. Contract or Grant No.
NAS3-25266

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

13. Type of Report and Period Covered
Contractor Report
Final

14. Sponsoring Agency Code

15. Supplementary Notes

Project Manager, James A. Kish, Power Technology Division, NASA Lewis Research Center. Prepared for the 25th Intersociety Energy Conversion Engineering Conference cosponsored by the AIChE, ANS, SAE, ACS, AIAA, ASME, and IEEE, Reno, Nevada, August 12-17, 1990.

16. Abstract

The goal of the Autonomous Power System (APS) program is to develop and apply intelligent problem solving and control technologies to the Space Station Freedom Electrical Power System (SSF/EPS). The objectives of the program are to establish artificial intelligence/expert system technology paths, to create knowledge-based tools with advanced human-operator interfaces, and to integrate and interface knowledge-based and conventional control schemes "Autonomous Power Expert System" by J.L. Walters, T.M. Quinn, et al. (in proc. 1990 Goddard Conference on Space Applications of Artificial Intelligence). This program is being developed at the NASA Lewis Research Center in Cleveland, Ohio. The APS Brassboard represents a subset of a 20 KHz Space Station Power Management And Distribution (PMAD) testbed. A distributed control scheme is used to manage multiple levels of computers and switchgear. The brassboard is comprised of a set of intelligent switchgear used to effectively switch power from the sources to the loads. The Autonomous Power EXpert System (APEX) portion of the APS program integrates a knowledge-based fault diagnostic system, a power resource scheduler, and an interface to the APS Brassboard. The system includes knowledge bases for system diagnostics, fault detection and isolation, and recommended actions. The scheduler autonomously assigns start times to the attached loads based on temporal and power constraints. The scheduler is able to work in a near real-time environment for both scheduling and dynamic replanning.

17. Key Words (Suggested by Author(s))
Power system automation
Artificial intelligence
Space power systems

18. Distribution Statement
Unclassified – Unlimited
Subject Category 20

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price*
Unclassified	Unclassified	8	A02

NASA FORM 1626 OCT 86

*For sale by the National Technical Information Service, Springfield, Virginia 22161