Expert Systems Relations in Space Applications

Michael Brady
Cognitive Systems Lab
University of Alabama in Huntsville
Huntsville, AL 35899

Abstract

Independently designed expert systems that operate in common environment will almost certainly share some resources and data. They will also be connected in a network of interdependencies, each one reliant upon the others. Measures must be taken, therefore, to insure that these expert systems can communicate with one another.

This paper addresses the problem of expert systems relations as they pertain to space applications. First, these systems will be categorized and the relationships between them will be analysed. Then, an expert systems cooperation paradigm will be proposed. This paradigm will address various types of communication and coordination issues in an attempt to create a general model applicable in a variety of situations.

Introduction: The Common Goal

The interactions between expert systems that one might expect to find on a modern spacecraft are likely to be as complicated as the systems themselves. These systems must interact, however, in order to achieve their common goal of a successful mission. Like any team, these systems should each do what they do best, while assisting their teammates as much as possible.

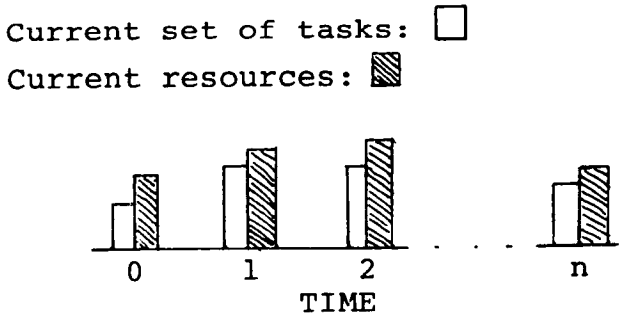

fig. 1 Mission Model

A simple model of a mission is depicted in fig. 1. The terms "task" and "resource" are used in the most general sense. Example tasks include life support systems, experiments, and fixing a

broken camera. Example resources include oxygen, electrical power, and tools. The common goal of the expert systems, then, is to cooperate in order to insure that at any given point in time the spacecraft's available resources either meet or exceed the requirements of the current set of tasks.

The Expert Systems Cooperation Paradigm

Ten categories of expert systems are listed in [1]. This list can be condensed into four classes which one would be likely to find on a spacecraft:

Diagnosis	--	Monitoring and interpreting sensory data. Possibly making predictions based on that data, ultimately trying to identify any problems that might affect the resource pool.
Scheduling	--	Developing a plan whereby all tasks are completed taking into consideration resource constraints.
Repair	--	Helping provide for resource repair either directly or in the form of advice.
Control	--	Controlling mission tasks in accordance with the schedule.

Given these four classes, the author proposes a paradigm for expert systems cooperation depicted in fig. 2. The "Physical System" may be the spacecraft as a whole or any subsystem therein. The paradigm is, therefore, meant to be valid in many different applications of varying scope. It is also possible to ignore any components of the paradigm that are not needed. If, for example, some subsystem does not have a repair expert system, that module and its associated data (the Symptoms and Repair Reports) can be ignored.

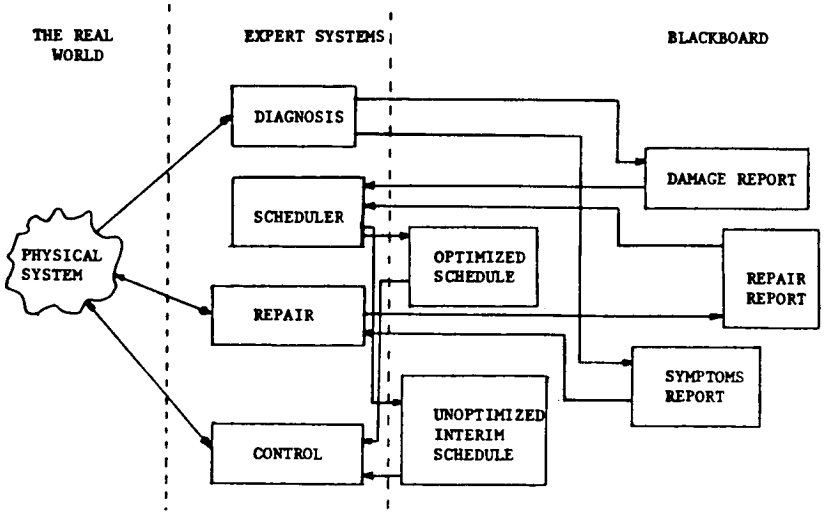

fig. 2 The Expert Systems Cooperation Paradigm

The paradigm utilizes a blackboard architecture, as described in [2], to facilitate expert systems cooperation. According to this scheme, each system monitors the blackboard, looking for something it can contribute. For example, when a Symptoms Report is written by the Diagnosis system, the Repair system will become active. Conversely, whenever the Symptoms Report is empty, the Repair module is inactive because there is nothing it can contribute. It is important to note that all communication between expert systems is through the blackboard.

In a typical case, considering the complete paradigm, both the Diagnosis and Control modules are active at the outset of the mission. At this point all the data objects except the Optimized Schedule are empty. When the Diagnosis system observes or predicts some abnormal change in the resource pool, it writes a Damage Report for the Scheduler and a Symptoms Report for the Repair system.

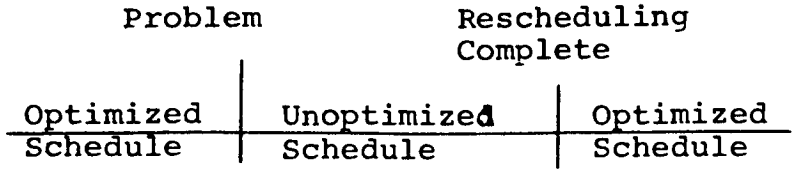

fig. 3 Unoptimized Interim Schedule

At this point there is a problem, however. The Scheduler must revise the optimized schedule, taking into consideration the resource changes, but what about the tasks that must continue during the rescheduling process? Moreover, the Scheduler cannot operate without a definite start time and a definite set of tasks to consider. Clearly, there must be an Unoptimized Interim Schedule for use in the time between the problem diagnosis and rescheduling completion (see fig. 3). The time of this interim should be an upperbound of the runtime of the Scheduler. The Unoptimized Interim Schedule should include as many tasks of the highest priority as can be accommodated. The Control system, then, will use the Unoptimized Interim Schedule whenever one exists. This process is the same when the Repair system, via a Repair Report, signals the Scheduler that the Resource problem has been fixed.

Conclusions

This paradigm will handle multiple resource faults, but includes no methodology for the isolation of or recovery from catastrophic failures. This sort of problem should be handled by the individual expert systems, and is independent of the paradigm in question. The paradigm will, however, help system designers plan the input to and output from their systems. It will also, hopefully, provide a useful framework for the development of the system or subsystem as a whole.

References

1. Nii, H. P., "Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures," "AI Magazine," Summer, 1986, pp. 38-52

2. Waterman, D. A., "What Expert Systems Have Been Used For," A Guide to Expert Systems, 1986, pp. 32-48