N92-23364

The Achievement of Spacecraft Autonomy Through The Thematic Application of Multiple Cooperating Intelligent Agents

p. 17

Philip J. Rossomando
General Electric Military and Data Systems Organization
(215) 531-5087

King of Prussia, Pennsylvania

KEYWORDS: Autonomy, Automation, Blackboard, Real-time Diagnosis, Expert Systems, Theme, Role Playing

1.0 ABSTRACT

This paper presents a description of UNICORN, a prototype system developed at General Electric for the purpose of investigating Artificial Intelligence (AI) concepts supporting spacecraft autonomy. With this objective, UNICORN employs thematic reasoning, of the type first described by Rodger Schank of Northwestern University, to allow the context-sensitive control of multiple intelligent agents within a blackboard-based environment (Schank, 1977). In its domain of application, UNICORN demonstrates the ability to reason teleologically with focused knowledge. Also presented within the following sections are some of the lessons learned as a result of this effort. These lessons apply to any effort wherein system level autonomy is the objective.

2.0 INTRODUCTION

A space-based system is composed of many subsystems whose associated performance can each contribute significantly to the success or failure of a mission. Each of these subsystems has its own changing needs and possibly conflicting requirements, which must be reconciled to maintain overall spacecraft health and operability. To address these issues, the solution space can be partitioned into multiple abstraction levels along both physical and functional boundaries. This partitive approach to spacecraft autonomy can be complemented through the context sensitive application of both conventional and advanced AI techniques within a hierarchical distributed control structure of the type currently found mostly within research institutions. The selective application of independent intelligent agents brings significantly more applicable knowledge to bear on a problem than is possible through the utilization of either conventional expert system or procedural methodologies alone.

C-2

For the past several years, GE in King of Prussia has been exploring AI problem solving paradigms that it feels will ultimately lend themselves to autonomous spacecraft operation. The first of these, rule/experiential and case–based reasoning, while allowing diagnostic knowledge to be expressed explicitly in the form of if–then rules and structured objects, falls prey to boundary conditions and is limited by the need for the *a priori* definition of faulted models and past cases. On the other hand, model directed reasoners of the type originally proposed by Johan de Kleer and Randal Davis are extremely CPU intensive ((deKleer, 1987), (Davis, 1985)). This fact limits their applicability within real–time environments of the type within which an autonomous spacecraft is expected to operate. To achieve autonomous operation, what seems necessary is a problem solving paradigm that allows the combining of the benefits of these approaches while at the same time minimizing their inherent weakness.

To attain the hybrid operation alluded to requires the ability to choose between knowledge sources employing both deep and shallow reasoning, based upon the current operational context of the space platform. This context need not necessarily be derived from the physical environment alone, but may arise from the goals and expectations identified to the spacecraft before and during operation. The blackboard control structure first introduced within the HEARSAY II environment appears to allow for this cooperative application of diverse knowledge sources (Erman, 1974). The twist however, introduced at GE, is not to opportunistically apply knowledge blindly but rather to do so in a focused manner that takes into account the spacecraft's context and the competing goals and demands of each of the subsystems of which the spacecraft is composed. This capability requires the utilization of intelligent agenda schedulers that understand these underlying needs while at the same time possessing a teleological comprehension of mission objectives within the constraints imposed by their assigned roles. This fact is significant because the goal of UNICORN is not just automation of space system functionality, but rather, is true autonomy.

While both autonomy and automation share attributes, and while both imply a reduction in human physical work load, autonomy also *implies* an *understanding of purpose*. This fact can be utilized to achieve a significantly greater reduction and/or elimination of the requirement for human cognitive activities related to spacecraft administration. Only through a deep understanding of mission objectives provided through the utilization of AI technologies supportive of autonomy, such as those herein documented, can we expect to produce self–directed spacecraft. By understanding why it was created, the autonomous system can be placed in a better position to prioritize its activities, utilize scarce resources, and generate expectations so as to achieve complex objectives. The question thus becomes how to develop a system wherein an understanding of both mission and ability can be utilized to guide performance in a highly unpredictable and constantly changing environment.

2.1 Background

The following sections provide the background needed in order to comprehend the difficulty of the spacecraft autonomy problem and why Thematic Reasoning within a blackboard–based control structure was chosen to address this issue.

2.1.1 Development History

GE began its investigation of autonomy by attempting to identify those spacecraft–related tasks where autonomy seemed most applicable. As indicated in Figure 1, spacecraft functionality can be divided into three areas:

1. Mission Management
2. Health & Maintenance
3. Payload Operations

Mission management relates to those tasks necessary to ensure that the payload can perform its assigned task. Such activities as on–board orbit maintenance and resource management fall under this heading. Under normal circumstances, an unmanned spacecraft's *Prime Directive* is to achieve as many payload mission objectives as possible. It is the duty of the Mission Manager to see to it that the overall spacecraft functions smoothly in addressing this directive. A payload is the package carried by the BUS that performs the task(s) for which the spacecraft was constructed.[1] Payload Operations determines what and when payload related activities are to be performed and makes demands of spacecraft resources based on these objectives.

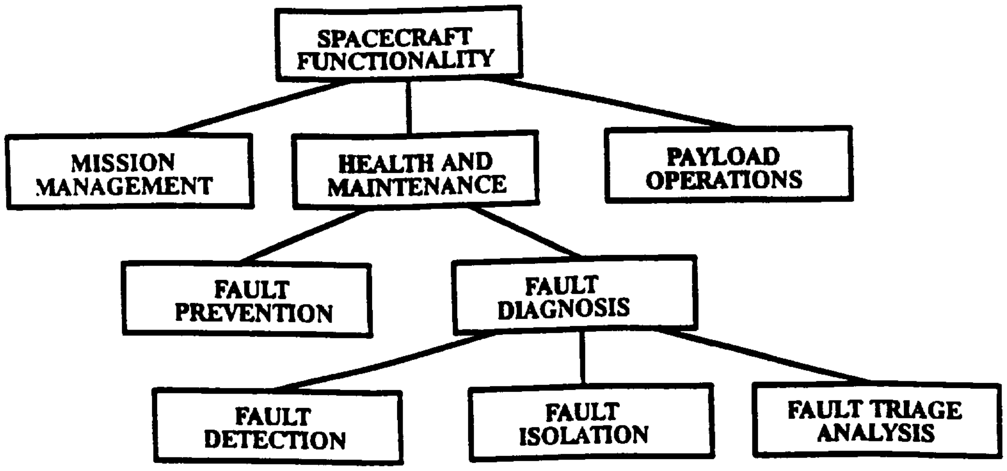

Figure 1. A Taxonomy Of Spacecraft Functionality

1. We use the terms BUS and spacecraft interchangeable within this paper.

Assuming an active payload, the demands placed on the BUS by the Mission Manager as a result of payload objectives can be expected to change with time. These demands are addressed by spacecraft BUS dedicated subsystems such as:

1. Power System
2. TT&C System
3. Thermal System
4. ACS

For example, the Electrical Power and Distribution System(EPDS) is expected to provide electrical power to the payload and the BUS's subsystems, while the Attitude Control System (ACS) is expected to maintain the spacecraft's stability. The Health and Maintenance function is responsible for ensuring that when resource/functionality demands are made of these subsystems they can be adequately addressed. This is done by:

1. Monitoring BUS related–subsystem operation relative to contextual expectations, thereby detecting possible fault–related symptoms.

2. Identifying trends that may imply future failures for which early corrective action can be taken.

3. Performing fault isolation and triage analysis.[2]

Because of a variety of factors, these are not easy tasks to make autonomous.

One significant reason for this difficulty is that the BUS subsystems are complex devices with many interacting subsystems of their own. Each of these is in turn composed of many parts whose behavior is critical to proper spacecraft operation. Faults within any of these components can cause adverse symptomatic behavior in the other components both within and external to the faulted subsystem. In addition, the behavior of spacecraft components can change without the presence of a fault anywhere within the spacecraft. The cause of an anomalous component behavior may be due to some change in the external spacecraft context. As indicated in Figure 2, in the course of its lifetime, an earth–orbiting satellite will change position relative to the earth, the moon, and the sun many times. Depending on that position, the behavior of critical satellite components can be expected to vary.

2. The UNICORN system performs triage analysis after it isolates the cause of a failure. In the case of a failed sensor, the identifying agent may choose to remove it from the scan list. If a backup system exists or a work around is possible, this may be provided as a recommended correction. On the other hand, there may exist no possible corrective action, and the spacecraft will then attempt to make do as best it can.

The electrical output of a solar array for example, can drastically change when the satellite is eclipsed by the earth or moon, without even a fault in that component. In the process of investigating spacecraft contexts, it was determined that these need not be just physical but may also be temporal (Allen, 1984). Component aging can also have a significant effect on behavior expectations. While these context changes can be predicted, others cannot and may appear as something other than what they actually are. For example, a nuclear burst exhibits many of the same characteristics as a solar flare. Another significant factor that further complicates the Health and Maintenance function is that even after a fault is corrected, spacecraft behavior does not instantly return to normal. During this period, if constraints are not relaxed, symptomatic behavior can be expected to continue even in the absence of a new fault.

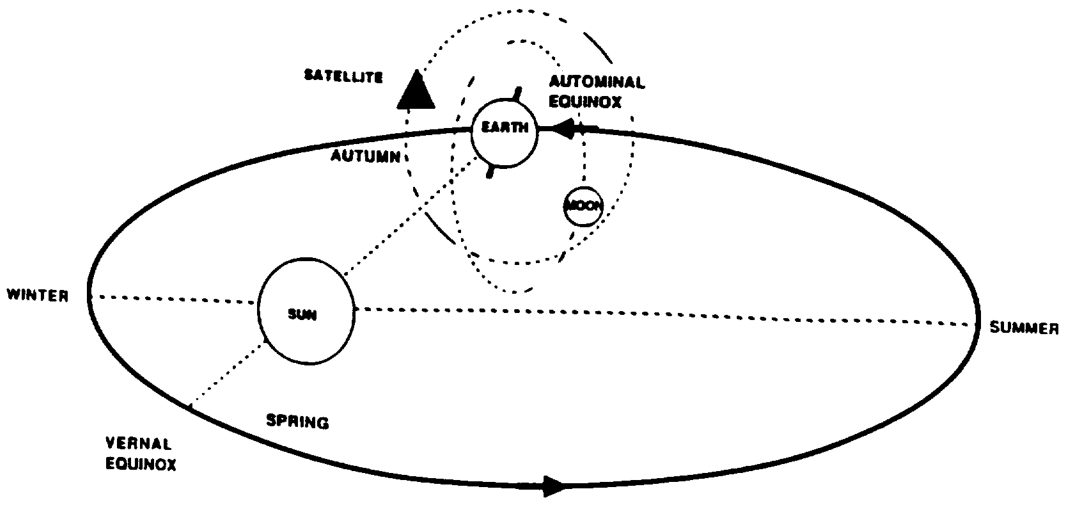

Figure 2. A Spacecraft's Context Is Dynamic

In 1986, in an attempt to automate the spacecraft Health and Maintenance function, GE utilized KEE on a Symbolics 3640 computer to develop a rule–based expert system, applying W. Clancy's classification strategy for performing diagnosis on a portion of the DSCS ACS ((Clancey, 1984), (Bell, 1986)).[3] While successful in its application domain, it soon became evident that enough rules could never be identified so as to ensure the diagnostic autonomy of an ACS let alone that of an entire spacecraft. In addition, it was discovered that if a fault exhibited symptoms slightly different from those expected, the rule–based system could not identify its cause.[4] Thus, the utilization of model–directed reasoning with constraint propagation and constraint relaxation was decided upon. This seemed to be an ideal approach as GE has a wealth of first–principle satellite knowledge readily available; by reasoning from first principles, faulted models are no longer necessary. Model–directed reasoning also seemed appropriate, as it appeared the only way to explicitly represent the peripheral paths of causal interaction introduced by the environment and adjacent, but not connected,

3. KEE is a trademark of IntelliCorp.
4. Even the loss of one sensor can completely blind a rule–based diagnostic expert system and cause it to arrive at erroneous conclusions.

subsystems ((Abbott, 1990), (Davis, 1985)). To investigate the accuracy of these predictions, in 1987 a model directed system was developed, also in KEE, to diagnose faults within an ACS. This system worked well but brought to light problems that originally were not apparent (Rossomando, 1988). For example, back propagation is not possible when the component's transfer function has no inverse, thus making other strategies necessary if fault isolation is to continue. However, because of the basic success of this effort, it was decided to tackle the system illustrated Figure 3.

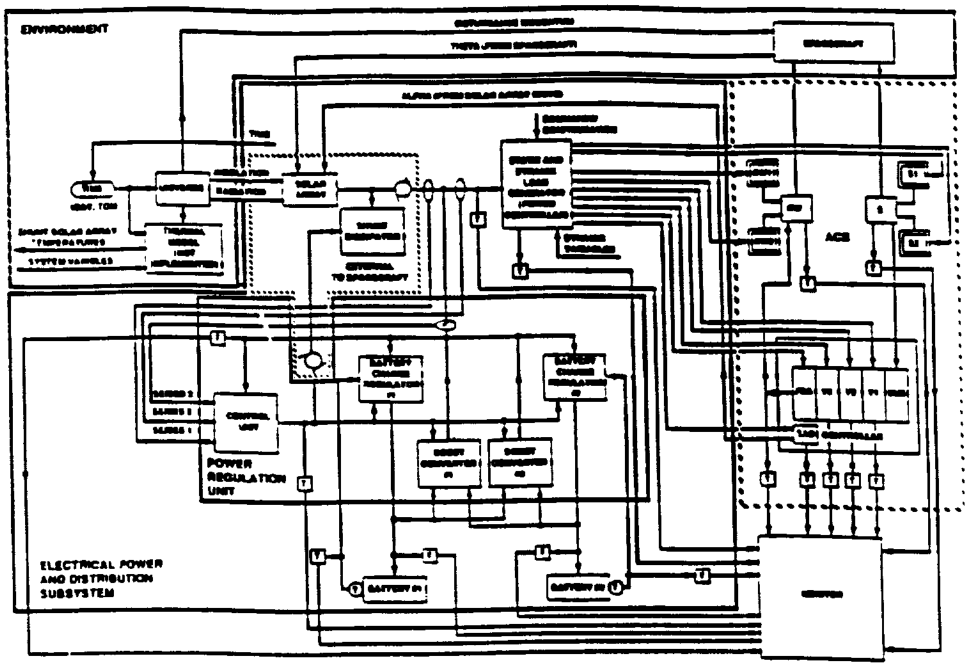

Figure 3. The UNICORN Spacecraft and Environment Model

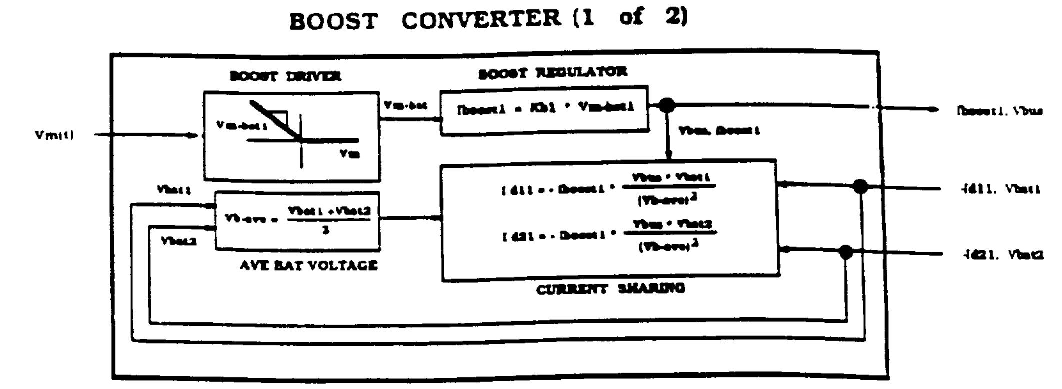

Figure 4. The Boost Converter

In the process of isolating faults within this more complex spacecraft, it was observed that constraint propagation can be very CPU intensive.[5] In an effort to address this problem, the spacecraft was divided into three separate but related causal models:

1. The Power System.
2. The ACS.
3. The External Environment.

In addition, it was observed that certain fault types have signatures that make them ideal fault model candidates. As a result, it was further decided that some combination of deep and shallow reasoning would be most appropriate in addressing real–time spacecraft Health and Maintenance issues. The question now was how to best combine these techniques in an environment that would allow reasoning across subsystem boundaries while also supporting detection, resolution, and mission demands not associated with fault isolation. Thus was born the UNICORN effort. Before describing that effort, however, the following sections introduce some of the non–diagnostic technologies on which UNICORN is based.[6]

2.1.2 The Blackboard Control Structure

A blackboard is much more than a shared global area.[7] It is that plus a whole collection of intelligent agents/actors and concepts that must work together to solve a complex problem. As indicated in Figure 5, a blackboard can be divided into multiple abstraction levels called panels. Each panel can be made to correspond to a different degree of solution abstraction. Associated with each panel are a number of *knowledge sources* whose purpose is to do certain specific sub–tasks within a much larger problem space. Each knowledge source in turn is an independent, self directed, intelligent agent composed of a number of condition and action parts. Knowledge source communication is strictly through the creation of *blackboard events*.[8] Some knowledge sources, such as fault isolators, symptom detectors, and triage analysts, are domain specific while others have generally more domain independent responsibilities. Two agents/actors in particular are worthy of note. These *control knowledge sources* are:

5. Within the model illustrated in Figure 3, as necessary, each identified component was itself modeled. For example, as illustrated in Figure 4, the Boost Converter within the Power System is itself made up of interacting components. At the lowest level, each component is associated with a constraint equation. This transfer function is used to provide the quantitative behavior of that component. It is the interaction of these component constraint equations that provides the overall subsystem behavior.
6. Because of space considerations, we will not detail UNICORN's model directed reasoning or classification problem solving techniques at this time. The reader is directed to the references documented at the end of this paper for a description of these technologies.
7. For an excellent general description of basic blackboard technology, references 7 and 9 are highly recommended by the author.
8. Within UNICORN, these events are produced through the use of object demons, and each blackboard panel is an object within an associative network. Panel objects are grouped together to form individual blackboards.

1. The *blackboard handler.*
2. The *agenda scheduler.*

The function of the blackboard handler is to initialize its assigned panel and to keep it clean and free of stale information during normal processing. Within UNICORN, this knowledge source is modeled as an expert system with its own *local context* and its own set of cleanup scripts. These scripts identify significant blackboard events and specify what must be done when they occur. Each blackboard handler is dedicated to a single panel, and so its *input* and *output levels* are identical. The activities of each blackboard handler as well as those of the other knowledge sources are controlled by an agenda scheduler/(blackboard monitor). Within UNICORN, there exists one agenda scheduler per blackboard.[9] The purpose of this knowledge source is to maintain the *focus of attention* of the other knowledge sources assigned to that blackboard. It performs this task by controlling the administration of the agendas within each blackboard panel. When a knowledge source detects an event or sequence of events about which it is interested and is in a position to act upon, it makes a bid to do so by placing a *Knowledge Source Activation Request (KSAR)* on the agenda associated with its assigned panel. This *blackboard object* describes what the bidding knowledge source intends to do if given control and provides an estimate of the affect that its application will have on the portion of the overall problem solution within its limited scope of control. In addition, also contained within the KSAR is an estimate of how much it will cost the system in terms of CPU utilization, solution time, and/or memory space if the associated knowledge source is allowed to execute. The agenda scheduler uses the information contained within the KSAR, along with its own knowledge, posted control directives from higher level monitors, and possible past experiences with the bidding knowledge source, to accept or reject the bid.[10]

If a bid is rejected by an agenda scheduler, the bidding knowledge source simply waits for another chance to bid again at some later time when its chances of success may be better. If, however, a bid is accepted, the knowledge source is given control and performs the action(s) described in the posted KSAR. The result of this action may be one or more blackboard events within that knowledge source's assigned panel or within another panel of its associated blackboard. The events so produced may cause yet other knowledge sources to become active. Two knowledge sources that cooperate in this manner are said to have *overlapping areas of interest.* For cooperative problem solving to work, each knowledge source must produce something that is of interest to another agent either within its own panel or within another panel in its assigned blackboard.

9. As indicated in Figure 5, UNICORN is composed of more than a single blackboard.
10. Each Agenda Scheduler within UNICORN is given a limited resource budget, which it can use to achieve assigned objectives. Any overdrafts must be covered from surpluses available through other agents.

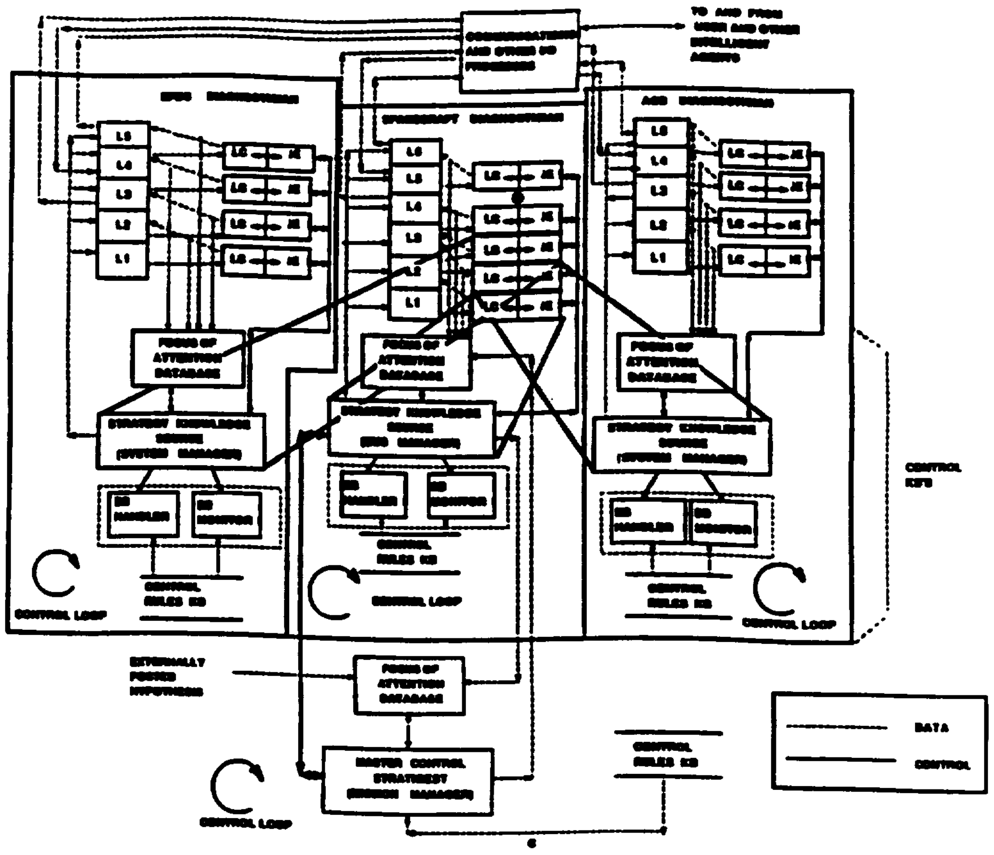

Figure 5. Some UNICORN Blackboards

2.1.3 Thematic Reasoning

Thematic reasoning was first proposed by Rodger Schank as a mechanism for story understanding. In this application domain, themes are utilized to generate expectations about the predicted behavior of an actor (Schank, 1977). Themes can produce these expectations because they can be made to contain background information upon which actor goal predictions can be made. A theme may be programmatically represented as a collection of related goals or even as a generator of these goals. Schank described three types of theme:

1. Role themes
2. Interpersonal themes
3. Life themes

Of these, the first two were extensively utilized within UNICORN. In a role theme, an actor's goals are determined by its role. Once a role is adopted, it sets up expectations about the goals and actions of the role player. Within a blackboard control structure, each knowledge source may be viewed as an actor playing a role. For example, within UNICORN the following actors are resident:[11]

11. These are just the agenda schedulers. There are many other actors, some of which will be introduced in the following sections.

1. ACS Manager
2. EPDS Manager
3. Payload Manager
4. Bus Manager
5. Mission Manager

These actors have specific control related roles, and when a significant event occurs it triggers each to act out that role. For example, the Payload Manager may pass (i.e., MTRANS) a request to the Mission Manager to perform an activity that may in turn result in a resource request of the Bus Manager. Likewise, a fault may be detected within the ACS that could cause the ACS Manager to inform the Bus Manager of a potential problem. The BUS Manager may in turn ask another actor to determine if related symptoms have been detected in the spacecraft subsystem to which it is assigned.

As illustrated in Figure 6, themes are important within UNICORN because they generate the goals that drive the different knowledge sources. These goals are indexed to produce the plans that describe how the identified goal is to be attained. UNICORN utilizes theme related scripts within role objects to determine what actions are required to attain a goal. Any goal may be associated with any number of strategies for its attainment. The strategy selected will depend upon the role being played, the current directive being followed, and the emotional and/or physical context of the actor. An actor may decide to suspend its role in certain circumstances.

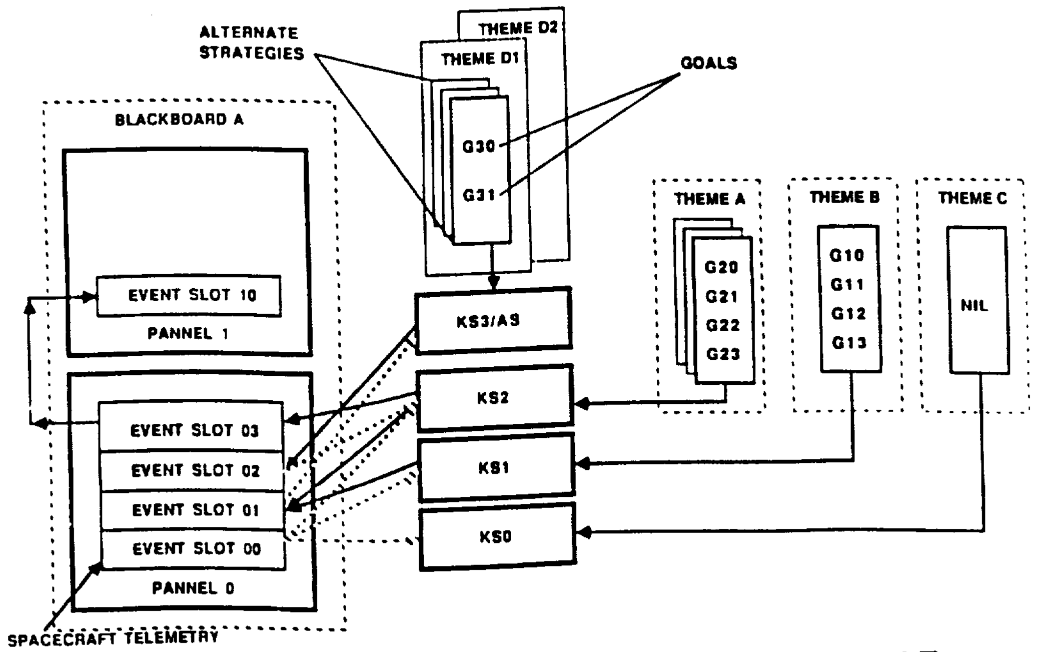

Figure 6. Themes Produce the Goals In Response to Blackboard Events

Role suspension can occur if an interpersonal theme gets invoked. An interpersonal theme consists of a set of test–action pairs where the test is defined as a blackboard event that is indicative of a physical event that threatens the functionality of the spacecraft. The action is the generation of one or more goals that will in turn produce associated plans that eventually cause a physical change to occur within the spacecraft. For example, an interpersonal theme like *Exhibit–Team–Spirit* can cause an agent like the ACS Manager to give up some or even all of its temporal resources in support of a teammate being threatened with the loss of its own subsystem's functionality. This would be done even though the role of the ACS Manager is to look after only ACS functionality. Thus Thematic Reasoning allows UNICORN to represent control knowledge explicitly, change task priorities, reallocate resources, select KSAR's for execution, and in general react to changes within its dynamic real–time environment.

3.0 SYSTEM FUNCTIONAL DESCRIPTION

The core of the UNICORN system was meant to be placed eventually either on board some future spacecraft or within a ground station wherein it would act as a performance analyst's advisor. Although designed to eventually control all spacecraft functionality, only those object level knowledge sources involved either in diagnosis or related mission management activities have been completed as of this writing. UNICORN's basic external architecture is illustrated in Figure 7. Within this Symbolics–based system, spacecraft and environment behavior is produced through the utilization of quantitative simulators. The environment simulator produces output that is utilized by the spacecraft simulator to produce subsystem behaviors. These are presented to the blackboard process as clock pulses, simulated telemetry packages, and contexts that are unpacked and utilized to produce blackboard events and to drive the diagnostic models. As indicated above, the blackboard paradigm has been utilized to coordinate the activities of a number of knowledge sources and to realize the best problem solutions traded against costs, such as time required to solve a problem and CPU utilization.

In addition, the control structure developed can handle probable cost solution value changes that may result due to temporal aging, physical re–configuration and critical emergency situation changes. To perform these tasks, UNICORN utilized a blackboard structure that was modeled as a conceptual taxonomy in which responsibility is divided along the diagnostic and managerial lines illustrated within Figure 8.

The best way to understand how the blackboard paradigm is utilized within UNICORN is to consider the real–world environment of which it is a model. Within this environment, console operators monitor incoming spacecraft telemetry, scanning certain key observational items for signs of trouble. If these are detected, other normally not actively monitored telemetry points are more closely examined. If a true anomaly is identified, management may request subsystem specialists to further isolate the anomalous behavior to one or more specific subsystems. For critical situations, more than a single expert may be utilized. Each of these individuals may apply a different problem solving technique to arrive at a conclusion. If multiple subsystems are involved, other experts who understand BUS and payload interactions at a much deeper level than do the individual subsystem experts may be consulted. Once a fault is isolated, corrective actions can only be taken after considering possible inter–system ramifications. Thus while recommendations may be made at lower levels, the final decision as to the corrective action to pursue is made higher in the hierarchy, after review of all options.[12]

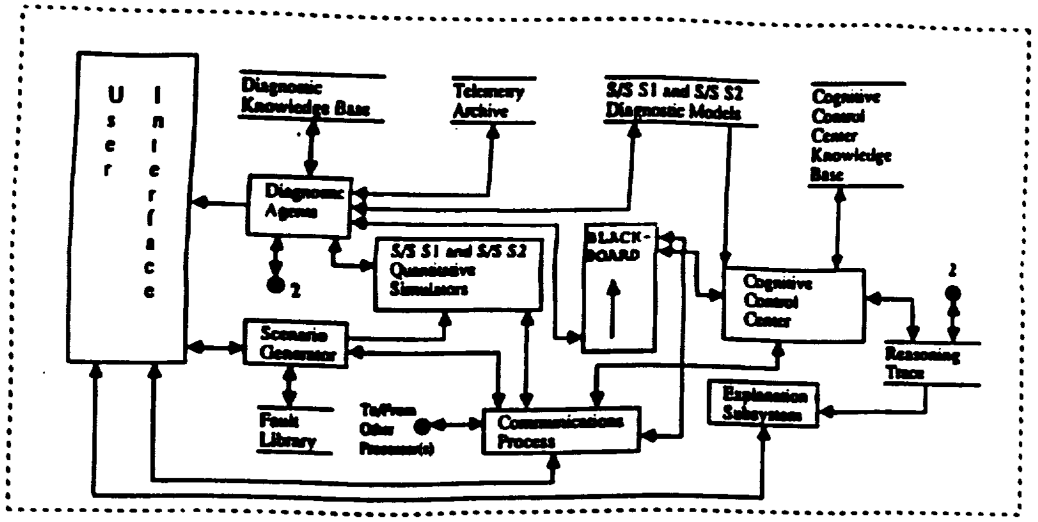

Figure 7. A High Level View Of UNICORN Component Parts

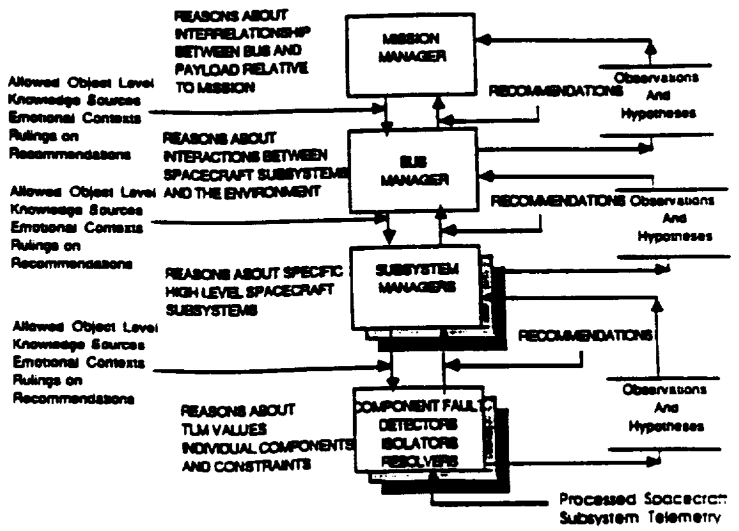
REASONS ABOUT INTERRELATIONSHIP BETWEEN BUS AND PAYLOAD RELATIVE TO MISSION
MISSION MANAGER
Allowed Object Level Knowledge Sources Emotional Contexts Rulings on Recommendations
RECOMMENDATIONS
Observations And Hypotheses
REASONS ABOUT INTERACTIONS BETWEEN SPACECRAFT SUBSYSTEMS AND THE ENVIRONMENT
BUS MANAGER
Allowed Object Level Knowledge Sources Emotional Contexts Rulings on Recommendations
RECOMMENDATIONS
Observations And Hypotheses
REASONS ABOUT SPECIFIC HIGH LEVEL SPACECRAFT SUBSYSTEMS
SUBSYSTEM MANAGERS
Allowed Object Level Knowledge Sources Emotional Contexts Rulings on Recommendations
RECOMMENDATIONS
Observations And Hypotheses
REASONS ABOUT TLM VALUES INDIVIDUAL COMPONENTS AND CONSTRAINTS
COMPONENT FAULT DETECTORS ISOLATORS RESOLVERS
Processed Spacecraft Subsystem Telemetry

Figure 8. One Way In Which UNICORN's Blackboard Structure May Be Viewed

To support the task flow outlined above, UNICORN's blackboards are arranged hierarchically, as illustrated within Figure 9. At the lowest leaf nodes of this taxonomy are those blackboards associated with the BUS subsystems mentioned in section 2.1.1. Each of these blackboards has been divided into three levels of abstraction corresponding to the different phases of diagnosis (i.e., detec-

12. Actually, the level to which a fault is propagated depends on the estimated impact on mission success. Higher level agents are only made aware of problems if they exceed the scope of control of a lower level agent.

tion, isolation, and resolution/prognosis). Within panel 0 of these blackboards are three symptom detectors. The first is a limit checker, the second is a trend analyzer, and the third uses constraint propagation to identify symptomatic behavior.[13] The first and second utilize fixed context specific limits to screen incoming telemetry. The second requires a number of telemetry frames to detect a symptom while the first needs only one. Both, however, are not usable immediately after a failure, unless constraint relaxation is evoked. The third can be used even after a failure but is CPU intensive and is normally allowed to execute only in situations where the other two can not operate.

Panel 1 of the above-mentioned blackboards has currently assigned to it two fault isolators. The first employs experiential rules to isolate a set of *a priori* defined faults, while the second employs model directed reasoning for fault isolation. The first is not always guaranteed to find the cause of an anomalous behavior, but when it does, it does so relatively quickly. The model directed analyst is more likely to always isolate a fault within its causal model but is CPU intensive and may take a considerable time to isolate certain faults within its domain of application.

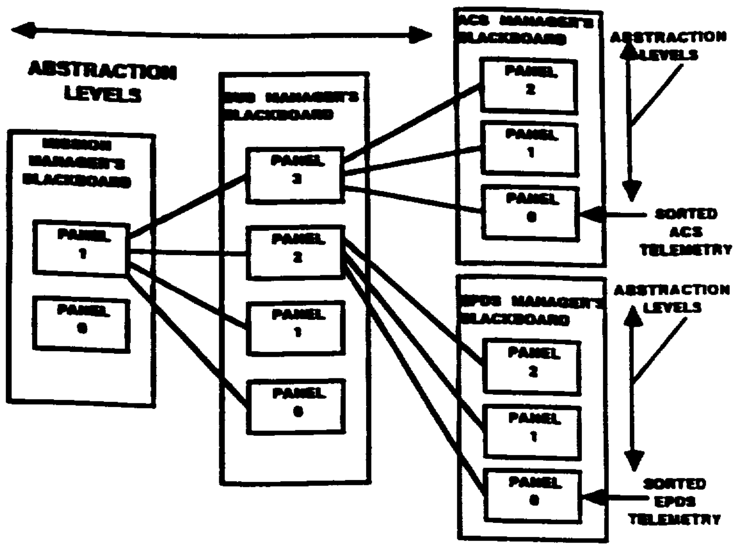

Figure 9. A Taxonomy of UNICORN Blackboards

Panel 2 contains a causal analyst and another agent which performs prognosis. The causal analyst, examines the causal network created within the first two panels and attempts to identify the cause of the fault.[14] All agents within a BUS subsystem associated panel adopt a closed world assumption

13. In addition to the knowledge sources mentioned, each panel is associated with the control level agents described in section 2.1.2.

14. The output of each diagnostic agent is a linked list of hypothesis elements which are arranged within a causal network. This makes the fault hypothesis explicit and readable by the other diagnostic elements. When multiple agents arrive at the same diagnostic conclusion, a numeric confidence level related to that hypothesis is incremented. If it turns out that an agent's hypothesized faults are correct, that agent has its reliability incremented. If it turns out to be wrong, the agent's reliability is decremented.

and assume the fault, if it exists, resides within its own domain of application. The output of this panel consists of one or more recommendations that are passed up to the BUS Manager for review and acceptance or rejection.

Above the BUS subsystem–related blackboards, within UNICORN's blackboard taxonomy, is the BUS Manager's Blackboard. Associated with this blackboard are an number of knowledge sources whose purpose is to reason about faults that seem to cross subsystem boundaries. At this level, the physical environment of the spacecraft is examined as the possible cause of an anomalous behavior. This blackboard is controlled directly by the BUS Manager, whose purpose is to ensure the smooth working of all BUS subsystems. The BUS Manger performs its task by utilizing Thematic Reasoning to determine what goals are necessary and and how they should be pursued. By utilizing a strategy appropriate to the urgency of the situation as seen by itself, its subordinates, and its superior, the BUS Manger can in effect be given an understanding of the importance of its decisions. This is very significant in an environment wherein there may not always be a single answer to a question. Once the BUS Manager, in association with its Health and Maintenance team, has reached a conclusion about what recovery actions are necessary to address a diagnostic situation, it reports them to the Mission Manager.[15]

It is the Mission Manager that is actually given an understanding of mission objectives. This is done by identifying to it which events indicate successful mission accomplishment and which indicate a threat to that objective. The Mission Manager controls the highest level blackboard within the UNICORN blackboard taxonomy. From this vantage point, it receives inputs from, and makes demands, of both the BUS and Payload Managers relative to the performance of their assigned spacecraft systems. Depending upon the nature of the reports received, the Mission Manger may change emotional context and in turn make decisions that result in orders or directives to its subordinates. The Mission Manager may approve or reject a subordinate's recommendations based on its wider understanding of the overall spacecraft operation.

4.0 LESSONS LEARNED

Many lessons have been learned as a result of this effort the following is just a partial list of some of the most interesting ones relative to UNICORN's utilization of the blackboard control paradigm:

1. A blackboard control structure does indeed allow multiple knowledge sources with different problem solving paradigms to work together harmoniously.

15. The BUS Manager is given a time budget by the Mission Manger to arrive at a diagnostic conclusion. It distributes this budget among its subordinate agents. If time appears to be running out, the BUS Manager can halt further investigation and report the current best hypothesis to the Mission Manager. This suggested action will be associated with some degree of confidence. The Mission Manager can than decide to go with the recommendation or else ask for further analysis, if further resources are still available.

2. Local contexts make sense within a blackboard environment, as they provide a knowledge source's own working area for solution derivation.

3. Solutions with higher confidence factors can be produced through cooperative problem solving.

4. A blackboard environment does indeed support distributed knowledge source development.

5. It is vital that the system be hosted on a CPU with sufficient space and horsepower to allow it to operate in a real–time environment.

6. Avoid toy problems, they produce toy solutions.

7. The selection of system goals and task priorities must be allowed to change based on the situation at hand and cannot remain static.

8. Autonomy within a complex system requires the distribution of functionality among multiple agents, each with a world view and scope of control limited to its assigned role.

9. Blackboard construction from scratch is difficult and is no longer recommended.

5.0 FUTURE DIRECTIONS

There are a number of enhancements that would greatly improve the current UNICORN system. The first is to move the system from its current Symbolics environment to a SPARC workstation or equivalent processor. In addition, it is recommended that the KEE–based blackboard architecture be replaced with one based on something like GBB by Blackboard Technology Group, Inc. These modifications should allow the system to run much faster. Second, it is recommended that a case–based reasoner be investigated for addition within the fault isolation panels. These systems could work with the model directed reasoners already in place to acquire new cases automatically. Third, it is recommended that a knowledge source that employs neural network technology be added to each symptom detection panel within UNICORN's blackboards. These knowledge sources could potentially replace the existing limit checking symptom detectors. The possibility of using the current themes contained within UNICORN as a source of diagnostic explanation should be investigated. If themes are good for understanding stories, then it would seem logical that they should also be usable in explaining diagnostic reasoning. Lastly, it might be interesting to go a little further and replace some of the pre–programmed script–based behavior currently employed within UNICORN with dynamically–derived plans produced in an attempt to address an event–triggered objective. Alas, all of this however is for another day.

6.0 CONCLUSIONS

This paper has described one approach to the achievement of system level autonomy. This approach advocates partitive analysis and the distribution of functionality between many intelligent and simi–independent agents, each such agent having a limited world view, but having some understanding of the value of its contributation. In the process of constructing the blackboard control structure, it was discovered that this task was every bit as difficult as constructing the individual diagnostic agents. To shorten the development effort, consideration should be given to the utilization of a commercial blackboard building environment. It was discovered that the concept of Thematic Reasoning allowed for the envisionment of diagnostic agent behavior by making explicit the goals associated with each agent. This explicit representation of functionality helped in the knowledge source construction task.

In the process of constructing this system, many interesting concepts were explored, and much knowledge was gained that will have definite applicability to future efforts in the area of autonomous systems. Much work still remains to be done, however, and many questions as yet remain unanswered.

7.0 ACKNOWLEDGEMENTS

The work documented within this paper was supported with IR&D funding from GE ASTRO Space Division from 1986 through 1990. The author wishes to take this opportunity to thank Robert Schule, Jim White, Chris Sterritt, and Ben Bell for their valuable assistance in helping to bring the future of spacecraft autonomy just a little bit closer to reality.

REFERENCES

1. Abbott K., *Robust Fault Diagnosis Of Physical Systems in Operation*, Phd Dissertation, Rutgers, The State University, 1990.

2. Allen J., *Towards a General Theory of Action and Time*. Artificial Intelligence, 23, 1984.

3. Bell B., Gerner M., and Sterritt C., *DSCS ACS Diagnostic System*, Technical Information Series, No. 87SDS003, GE Astro–Space Division, December 1986.

4. Clancey W., *Classification Problem Solving*. In National Conference on Artificial Intelligence, 1984.

5. Davis R., *Diagnostic Reasoning Based on Structure and Behavior*. In Daniel G. Bobrow, editor, Qualitative Reasoning about Physical Systems, The MIT Press, 1985.

6. de Kleer J. and Williams B., *Diagnosis of Multiple Faults*, Artificial Intelligence, 32, 1987.

7. Engelmore R., Morgan T., *Blackboard Systems*, Addison Wesley Publishers, 1988.

8. Erman L., *An Environment And System For Machine Understanding of Connected Speech*, Phd Dissertation, Stanford University, 1974.

9. Nii H. P., Anton J. J., *Signal–to–Symbol Translation: HASP/SIAP Case Study*, The AI Magazine, Spring 1982.

10. Rossomando P., Sterritt C., Johnson D., *Model Based Diagnosis In An Analog Spacecraft Domain*, Technical Information Series, No. 87SDS043, Astro–Space Division, 05 January 1988.

11. Rossomando P., Bell B., Sterritt C., *UNICORN: Spacecraft Diagnosis in a Distributed Problem Solving Domain*, Technical Information Series, No. 88SDSD002, Astro–Space Division 1989.

12. Schank R., Abelson R., *Scripts, Plans, Goals, and Understanding*, Lawrence Eribaum Associates, Inc. Publishers, 1977.