AGARD

ADVISORY GROUP FOR AEROSPACE RESEARCH & DEVELOPMENT

7 RUE ANCELLE 92200 NEUILLY SUR SEINE FRANCE

Paper Reprinted from
AGARD Conference Proceedings 499

Machine Intelligence for Aerospace Electronic Systems

(L'Intelligence Artificielle dans les
Systèmes Electroniques Aérospatiaux)

NORTH ATLANTIC TREATY ORGANIZATION

A SYNERGISTIC APPROACH TO REASONING FOR AUTONOMOUS SATELLITES

Captain James M. Skinner
Phillips Laboratory
Kirtland AFB, NM 87117

Professor George F. Luger
University of New Mexico
Albuquerque, NM 87131

Summary

Based on our earlier research [1,2], we are convinced that the best way to approach problem-solving tasks is not through any single method of reasoning, but by a method that will allow several reasoning methods to be blended together. The thrust of this effort is to develop a synergistic approach to reasoning that will allow a system to rely on multiple reasoning methodologies, thus benefiting from the strengths of each of the reasoning methods, while minimizing their respective weaknesses. This paper discusses the steps we have taken towards developing such a system, which are: (1) the categorization of reasoning methods, (2) the selection of reasoning approaches to blend, (3) design of a framework to blend the systems, and (4) proposed tasks to investigate the result.

Human Reasoning

Research in Artificial Intelligence (AI) is sometimes partitioned into two approaches: a psychological approach which employs AI programs as useful tools for studying the mind, and an engineering approach which uses AI programs to solve problems, regardless of their similarities to human reasoning. While this effort is concerned with the latter approach, it is useful to examine the reasoning methods used by humans to guide us in modeling a reasoning approach for machines.

When humans reason they rely on many different reasoning techniques, including:
(1) relying on memory of past cases when solving problems which are similar to these past cases.
(2) using heuristics, or rules of thumb, when confronted with a familiar situation.
(3) following a procedure to solve a problem.
(4) developing a mental model of the problem, or referring to a schematic to diagnose problems in a complex system.
(5) comparing an unfamiliar problem to a past problem (and solution) from a different domain.
(6) using a formal logical method to prove a theorem is true.

The methods outlined above are by no means an exhaustive list of human reasoning methods. They do, however, represent a sample of methods that humans use. We might wish to ask if a computer could reason in a similar method.

First, we must address the question of whether a machine is capable of reasoning. While entire books have been written on related subjects [3,4,5], we will sidestep the issue by defining reasoning as "the drawing of inferences or conclusions from a set of facts or suppositions." When confined to this definition, few people will disagree that not only is it possible for computers to reason, but that many already possess this capability.

With this in mind, we can see that each of these reasoning methods have been automated to some degree as (1) case-based reasoning, (2) rule-based reasoning, (3) conventional reasoning, (4) model-based reasoning, (5) analogical reasoning, and (6) automated reasoning respectively. While it is doubtful that we will ever be able to automate the human reasoning process, a good facsimile might come from developing a system that is able to combine some or all of these reasoning techniques. Before we discuss how that might be done, it would be useful to develop a scheme to categorize the different methods of reasoning.

Several schemes are possible for classifying reasoning methodologies. Past approaches have included classification based on the degree of precision (e.g., sound versus fuzzy reasoning) or degree of generality (general purpose versus special purpose reasoning). The approach taken in this paper is a classification based on what we call the "depth" of reasoning.

Shallow Versus Deep Reasoning

Reasoning methods can be categorized as "shallow" or "deep," depending on the type of knowledge used in the system. An approach taken by Harmon (Figure 1) is to classify knowledge into three levels: (1) heuristic or shallow knowledge, (2) domain knowledge (includes procedural models) and (3) deep or theoretical knowledge [6].

Facts Heuristics	↑ Shallow Knowledge
Procedural Models and (Explicit) Domain Models	
Abstract Theories, Laws, and Principles	Deep Knowledge ↓

Figure 1. Three Levels of Knowledge [6].

To illustrate this knowledge classification scheme, Harmon considers how a non-technical person would react if their television set did not work. First, he might check to see if it was plugged in.

If so, he might shake the cord and switch all of the knobs on and off several times. If all of this failed, the person would decide he needed to take the TV to the shop or purchase another one. This person has only Level 1 knowledge of television sets. It consists of a few heuristics that he applies to televisions, toasters, and other electrical devices.

If the person takes the TV to a repair shop, he expects the repair person to have much more knowledge about televisions. The repair person has Level 2 knowledge of television sets; she probably does not have an advanced degree in electronics or understand the physics of television, but she has domain or procedural knowledge of televisions and TV repair. The domain model includes many heuristics about how a malfunction in one component might induce symptoms elsewhere; the procedural model provides her with a step-by-step approach to troubleshooting the TV. The repair person can rely on both models to guide her in the systematic diagnosis of the TV set, and in selecting an appropriate repair strategy.

If you wanted to design a new television, you could probably find someone in an engineering department of a university who understands how televisions worked. This person has Level 3 knowledge of television sets. He probably can't fix your television set, but he has the underlying or deep knowledge of physics and electronics that would enable him to create a device that would function as a TV.

We can extend this knowledge classification scheme to reasoning systems. As might be expected, traditional expert systems (i.e., rule-based system) are associated with shallow knowledge; they rely on heuristic knowledge to solve a problem. Appropriately, we will refer to these systems as shallow reasoning systems. Conventional systems, which normally encode procedural knowledge, will be considered as the midpoint between shallow and deep reasoning systems. Automated reasoning systems (i.e., theorem provers) are associated with deep knowledge, and will be called deep reasoning systems.

When extending this classification scheme to reasoning systems in general a problem arises; theories can be encoded in rules, and heuristics can be represented as models. To account for this, we will modify the definition slightly.

To classify a reasoning system as shallow or deep, we will consider whether the system uses an implicit or explicit model of the domain when reasoning. The domain model is considered implicit if there is no distinction between different types of knowledge. Rule-based systems are an example; they use heuristic knowledge arising from many different sources (e.g., empirical knowledge, structural knowledge, causal knowledge) and will often combine different types of knowledge into a single rule without distinction. A system using an implicit domain model will be considered a shallow reasoning system.

The domain model is considered explicit if a distinction is made between the different sources of knowledge. The types of knowledge and the role that each plays in the problem-solving process is made clear. As an example, a system that models the subcomponents of an electronic component and describes the relationship between the components is an example of a system employing an explicit domain model, and would be considered a deep reasoning system.

While there is a tendency to attach a bad connotation to the term shallow, this is not justified with reasoning systems (consider that MYCIN, a very successful expert system used to diagnose meningitis infections, contained only shallow knowledge). Each form of reasoning has strengths and weaknesses. The application should dictate which form of reasoning is used.

Applications that require the system to capture experiential knowledge are well suited for shallow reasoning systems. This type of knowledge is easily encoded into empirical associations. Consider developing an expert system to capture the knowledge of a retiring technician. The technician's knowledge has been gathered over the years from many different sources and is best represented as an implicit model.

However, the lack of explicit representation of more fundamental knowledge can cause some serious problems. These problems include: (1) rapid degradation outside the narrow domain of expertise of an expert system, possibly leading to incomplete or incorrect conclusions; and (2) the inability to transfer knowledge to other tasks. This inability stems from the fact that heuristics are usually task specific; rules written for the diagnosis of a component will not normally be useful in an expert system developed to design the component, even though the underlying mechanism and physical principles are the same for both tasks [7].

Diagnosis of man-made devices is a good application area for a deep reasoning system. Especially if the device is a complex state of the art component with little or no expertise available. The explicit model of the deep reasoning system allows the system to respond to situations that could not have been predicted. But development of the explicit domain model requires the fundamental principles of the domain to be well understood. This makes deep reasoning inappropriate for areas which are not well understood. In medicine, for example, most of the knowledge used for diagnosing and treating diseases is empirical, not based on a model of the relevant biological and chemical mechanisms.

While reasoning methodologies do not normally rely purely on shallow or deep reasoning techniques, we propose that they can be broadly categorized as one or the other. Returning to the three levels of knowledge of Figure 1, we claim the reasoning methodologies can be placed in correspondence with the levels of knowledge as shown in Figure 2. In this paper we will discuss the three shallow reasoning methodologies (case-based, rule-based, and conventional) and one of the deep reasoning methodologies (model-based).

Knowledge		Reasoning System
Facts	Shallow	Case-Based
Heuristics		Rule-Based
Procedural Models		Conventional
(Explicit) Domain Models		Model-Based
Abstract Theories, Laws, and Principles	Deep	Analogical
		Automated

Figure 2. The Depth of Selected Reasoning Systems.

Case-Based Reasoning

The mounting evidence that human experts rely heavily on memory of past cases when solving problems has led to an increase in the research of case- based reasoning (CBR) [8]. In CBR, past cases are used to solve a new problem case. This can increase both the quality and the efficiency of the reasoning by deriving shortcuts and anticipating problems in new situations based on past experience with similar cases.

There are two main types of CBR: classification and problem-solving. Classification CBR argues that a new situation should or should not be treated like a past one based on similarities or differences with the past case. Problem-solving CBR formulates a solution suited to the new case by modifying or adapting past solutions. Classification CBR is usually used for strategic planning or legal reasoning while problem solving CBR is typically used for design or diagnosis.

Proponents of CBR claim several advantages. First, the shortcuts in reasoning and the capability of avoiding past errors enhance performance. Second, no causal model or deep knowledge of the structure is necessary, although their existence will improve performance. A third advantage is the scalability of CBR. While there is a bottleneck in choosing the base cases to reason with, researchers have managed this problem by indexing the cases. Researchers believe that indexing will allow them to scale CBR up to the point where they can tackle real problems in real time. Finally, knowledge acquisition in case-based reasoning is much easier than for other reasoning methods. This is because much of the knowledge required for CBR is in the form of cases. Furthermore, many domains have existing case bases (e.g., medicine, law) that could be used as a seed.

Primary concerns with CBR are: (1) organizing cases in memory, (2) selecting the relevant cases, and (3) modifying existing cases to fit new problems. All of these areas are current topics of research.

Case-based reasoning is suitable for domains involving classification (medical, legal, planning) and problem solving (design, mathematics, diagnosis). CBR is best suited to domains in which many training cases are available, and where it is difficult to specify appropriate behavior using abstract rules.

Rule-Based Reasoning

Rules are the most commonly used knowledge representation technique in artificial intelligence. In a rule-based expert system, the domain knowledge is represented as sets of rules that are checked against a collection of facts or knowledge about the current situation. The rules are expressed as IF-THEN statements. When the If portion of the rule (the premise) is satisfied by the facts, the action specified by the THEN portion (the conclusion) is performed. This action may result in the addition of new facts, continuing the cycle until the goal is achieved.

The strength of rule-based systems lies in the simplicity of their construction and maintenance. Rules can be easily constructed because experts tend to express most of their problem-solving techniques in terms of situation-action rules which can be readily coded. Maintenance of the system is somewhat simplified because each rule approximates an independent chunk of knowledge, so that existing knowledge can be refined and new knowledge can be added in a modular fashion.

The weakness of rule-based system is their lack of robustness. This arises from the fact that only an implicit model exists, not an explicit one. The information contained in the model is a collection of empirical associations drawn from an expert. It is impossible for the system to respond to a situation unforeseen by the expert (or not coded by the knowledge engineer).

Rule-based systems have been successful in a wide variety of applications including diagnosis, configuration, and control. The classic example of a rule-based system is MYCIN, an expert system designed to solve the problem of diagnosing and recommending treatment for meningitis and bacteremia. While tools exist capable of unambiguously diagnosing meningitis, these tools require on the order of 48 hours to return a diagnosis. Unfortunately, treatment for meningitis patients must begin immediately. The goal of MYCIN was to emulate the doctor's expertise of forming a diagnosis that covers the actual infecting organisms based on initial symptoms and test results [2].

Conventional Reasoning

Contrary to popular belief, AI advocates do not propose that all problems should be solved with AI tools. Conventional programming methods will suffice in many instances. Table 1 lists the differences between conventional systems and AI systems.

Table 1. Comparison of Conventional and AI Systems [9].

Conventional	AI System
Representation and use of data	Representation and use of knowledge
Algorithmic	Heuristic
Repetitive Process	Inferential process
Effective manipulation of large data bases	Effective manipulation of knowledge bases

Many problems are best solved by conventional programming [9]. For example, problems which have tractable mathematical solutions such as solving differential equations with numerical analysis techniques are not appropriate for AI, while problems requiring algebraic simplification lend themselves quite readily to symbolic reasoning. Problems that can be solved with algorithms (formal procedures that guarantee the correct solution every time) are better left to conventional systems. For example, it is more cost effective to sort lists with a conventional system than with an AI program.

Model-Based Reasoning

The first of the deep reasoning systems to be discussed is model-based reasoning. Model-based reasoning uses an explicit model of a system to describe the components of a physical system, the connections between the components, and the behavior of each of the components [10]. The system is able to reason about physical laws which apply to the system, and the effect the laws may have on the system. This form of reasoning allows for a more robust response to a previously unencountered set of circumstances.

Model-based systems commonly use causal reasoning, reasoning from first principles, and reasoning from the principle of locality in solving a problem. Causal reasoning relies on knowledge concerning how the behavior (or misbehavior) of one component affects the behavior of another component. Reasoning from first principles relies on the laws of physics or mathematics to predict or explain behavior of a system. The principle of locality considers how components are connected (mechanically, electrically, physically) in determining how behavior of one component can be influenced by another component.

The strength of model-based systems lies in their fundamental knowledge of the domain. This allows the system to reason about situations previously unencountered, and for which the system has not been explicitly programmed. The type of knowledge representation also allows the knowledge to be transferred to other tasks; that is, a model-based system originally developed for diagnostics would be of significant value when developing a design expert system for the same domain whereas knowledge from a diagnostic rule-based system is of questionable value.

The weaknesses of a model-based system become apparent only when it is a pure model-based system; that is, when no heuristics are used. Diagnostic searches through components of systems based on the principle of locality make sense only if all components are equally fallible and equally accessible. Otherwise, rules are essential. In fact, many cases can be shown where model-based reasoning requires more time than rule-based systems. For this reason, most model-based systems will include at least a few rules to improve performance.

In a subsequent section we argue for blending these four reasoning methods to produce a synergistic effect. We claim a modified blackboard is a suitable framework for merging the individual reasoning methodologies. The traditional blackboard architecture is discussed next to provide a foundation for these arguments.

Blackboards

The blackboard architecture was developed for a speech understanding system developed in the Seventies known as Heresay-II [11]. Since its development, the blackboard model has been proven a robust model of problem solving on applications from ocean surveillance (HASP) to air traffic monitoring (TRICERO).

The traditional blackboard model contains three major components as shown in Figure 3 [12]:

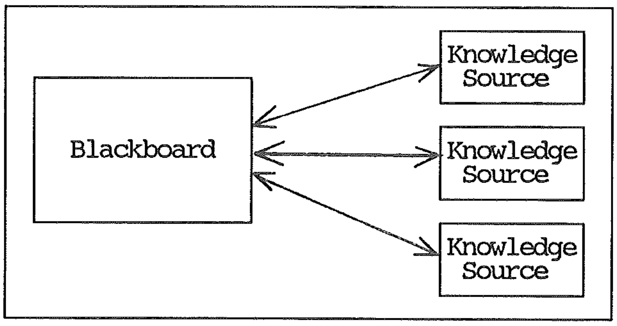

Figure 3. The Traditional Blackboard Architecture [12].

(1) The knowledge sources - The knowledge needed to solve the problem is partitioned into knowledge sources, which are kept separate and independent.
(2) The blackboard data structure - The problem solving state data are kept in a global data store, the blackboard.

Knowledge sources produce changes to the blackboard, which lead incrementally to a solution to the problem. Communication and interaction take place solely through the blackboard.
(3) Control- The knowledge sources respond opportunistically to changes in the blackboard. The structure of the control is left open. It can be in the knowledge sources, on the blackboard, or a separate module.

Although implementations vary, knowledge source activity is usually event driven. Each change to the blackboard constitutes an event that in the presence of specific other information on the blackboard can trigger one or more knowledge sources. The control mechanism selects a single knowledge source to execute its action on each problem-solving cycle. The control mechanism may use a variety of criteria such as the credibility of the knowledge source's triggering information, the reliability of the knowledge source, or the importance of the solution it would generate. When a knowledge source is triggered, it will typically produce new blackboard events. These events may in turn trigger other knowledge sources [11].

Blackboard systems construct solutions incrementally. On each problem-solving cycle a single knowledge source executes, generating or modifying a small number of solution elements. Some elements are assembled into growing partial solutions; others may be abandoned. Ideally, elements are eventually assembled into a complete solution.

As an illustration of how a blackboard uses cooperating knowledge sources, we can compare its operation to an aircraft accident investigation board. Whenever an Air Force aircraft crashes, an investigation is performed to determine the facts surrounding the case. After the investigation is complete, a board is held. The board consists of the board chairman and specialists who may be able to help determine the cause of the accident (e.g., pilot, navigator, aircraft mechanic). The chairman lists the facts on a blackboard in the front of the room. He then asks for comments from the group.

If someone in the room has information to add, they raise their hand. The chairman calls on one of the people with their hand raised, and that person adds new information (facts or hunches) to the board, then returns to their seat. Based on this new information more hands may go up, and some of the hands that were up may go down. The chairperson again selects someone. This person may add new facts or hunches, or may refute earlier hunches by other members of the board. The session continues until the cause of the accident is determined, or until the group can contribute no new information.

The parallel to a blackboard is obvious. The chairman is the control associated with the blackboard. The panel members are the knowledge sources, each with a particular domain. The blackboard in the front of the room is the blackboard structure of the architecture.

The Synergistic Reasoning Approach

Each of the reasoning methods previously discussed has associated strengths and weaknesses. Merging the methods in the proper fashion could create a system that would benefit from the strengths of each of the methodologies while minimizing their weaknesses. It is postulated that a such a blend would result in a synergistic effect, allowing a system to solve problems that could not be solved by any of the individual reasoning methodologies.

Such a synergistic blend is possible by making a fundamental modification to the blackboard problem-solving approach. The essence of the modification is to partition the system based on reasoning methodologies rather than knowledge modules. Indeed, it may be more appropriate to partition the system based on methods of reasoning. Consider a person taking a (closed book) test. All knowledge to be used during the test is self contained. There is no reason to believe that the knowledge is partitioned into different modules within the person. However, the person may use several different methods of reasoning about the problem. These methods of reasoning can be modeled as "reasoning modules."

This approach will produce a synergistic effect by allowing the modules to focus their individual strengths on a problem. By cooperating through the blackboard and posting partial solutions, a problem that could not be solved by any of the systems individually could be solved by the system as a whole. That is, one reasoning module could post a partial solution not obtainable by any of the other modules, and while it might not be able to generate the desired solution, one of the remaining modules, which was also unable to generate the desired solution from the original problem, might be able to do so based on this new result. A second synergistic effect is expected from the ability of one reasoning system to refute conclusions of another reasoning module. A diagram of the synergistic reasoning system is shown in Figure 4.

Satellite Autonomy

Satellite Autonomy is a suitable application for the proof of concept of the SRS. Satellite control is a complicated, tedious, and labor intensive process. According to a 1989 GAO study, over 4,000 government and contract staff are required to operate the Air Force Satellite Control Network consisting of fixed ground-based tracking stations, central control facilities, and communication links [13,14]. This network currently controls the operations of approximately 80 on-orbit satellites. Predictions are that 135 satellites will be on-orbit by the year 2000, and 150 will be on-orbit by 2015.

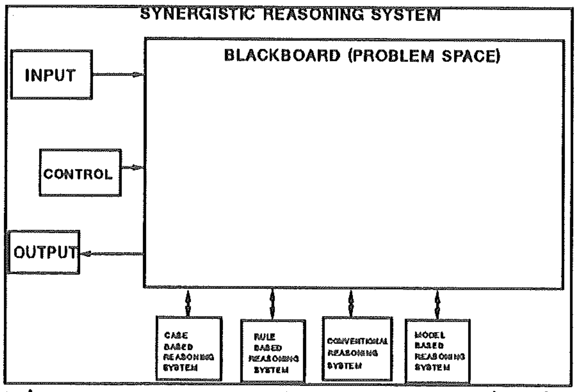

Figure 4. The Synergistic Reasoning System

However, the number of controllers supporting the network is likely to remain constant while the level of expertise decreases due to retirements [15].

As early as 1985, AI technology was identified as capable of supporting high levels of satellite autonomy. A Jet Propulsion Laboratory study identified specific applications that could benefit from one or more AI techniques [16]. As a method of meeting the goals of satellite autonomy, it is possible to develop autonomous subsystems for the satellite. Autonomy for each subsystem would be valuable in its own right, and could serve as components of a fully autonomous satellite. Subsystems on-board that were identified as well-suited for autonomous operations include: (1) guidance, navigation, and control, (2) power systems, (3) thermal control, and (4) payload management.

The same 1985 report defined eleven levels of satellite autonomy which are shown in Appendix A. Current satellites operate at about Level 3. The goal of an application associated with the synergistic reasoning system would be to provide a satellite with the salient characteristics associated with Level 5 operation, specifically, autonomous fault tolerance for operations in the presence of faults specified a priori. This capability will employ spare system resources, if available, or will maximize mission performance based upon available capability and/or available expendables without ground intervention.

Satellite autonomy is well suited as a domain for testing the Synergistic Reasoning System. Current control of satellites is performed through the sole use of conventional programs and frequent human intervention. A significant increase in effectiveness in satellite operations is expected from implementation of a synergistic reasoning system since performance improvements are documented for rule-based and model-based systems on similar domains. While there is less experience with case-based systems, the unpredictable behavior of spacecraft often leads to problems which are currently solved by relying on past cases.

Figure 5 is a diagram of a Synergistic Reasoning System for satellite autonomy. Note that human intervention has been modeled as a separate reasoning module. The blackboard lists the satellite subsystems. The individual expert systems are shown from left to right according to the depth of their reasoning methodology. An implicit control line runs from the control module to each of the components of the system. A proof of concept in this domain would be limited to a single subsystem, such as the fault diagnosis and recovery of the autonomous navigation system.

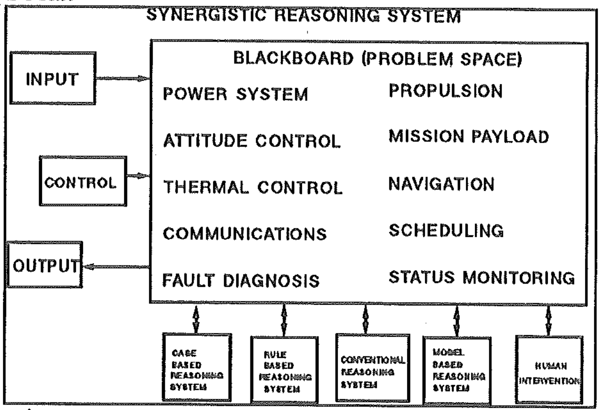

Figure 5. SRS for Satellite Autonomy

Satellite Autonomy Sample Operation

As an example of how the reasoning system might produce a synergistic effect from the proposed architecture consider the following scenario:

I. During normal operation, the conventional program performs routine operations.

II. The signal from the ground station falls below a predetermined level.

III. A message is posted to the blackboard by the conventional system that a problem exists.

"Signal weak from ground station"

IV. The executive module determines that the problem is diagnostic in nature and does not require immediate human intervention. The rule-base reasoner is activated.

V. The rule-base tries a series of data driven quick fixes.

IF <Signal Weak>
THEN Increase Gain.

IF <Gain Increased> and
<Signal Weak>
THEN Calibrate Pointing.

IF <Gain Increased> and
<Pointing Calibrated> and
<Signal Weak>
THEN Check Components <Sensor 1>.

VI. The executive module acknowledges the inability of the rule-base module to correct the problem. The model-based reasoner is activated.

VII. A model of the Sensor system is constructed:

Component1: Antenna
Component2: Amplifier
Component3: Phase Shifter
Component4: Power Supply
Component5: Ground System Output

VIII. The suspected components are posted on the blackboard. At this time the rule-based module is used again to determine the order of search. The rationale is that the blackboard may contain additional information on the current situation which will modify the corrective actions of the system. For example, if multiple subsystems were experiencing difficulty, there would be reason to suspect the power supply. In this instance, the rule-based reasoner adds no additional information and the model-based reasoner is reactivated.

IX. The model-based reasoner then examines each of the components and determines that all components are sound. The results are posted to the Blackboard.

Signal Weak From Ground Station.
Rule-base Quick Fix Tactics Unsuccessful.
Components of Communication System Verified Sound.

X. The executive module activates the case-based reasoner to determine if a similar event has previously occurred. The case-based system finds a match with a previous event:

Problem:
Ground Station Reports Weak Signal.
Rule-base Quick Fix Tactics Unsuccessful.
Components of Communication System Verified Sound.

Repair:
Fault in Attitude Control System.

XI. **Check Attitude Control System** is posted to the Blackboard. The model-based system builds the following components:

Component1: Attitude Control Electronics
Component2: Earth Sensor
Component3: Sun Sensor Assembly
Component4: Rate Gyro
Component5: Reaction Wheel
Component6: Solar Array Switch

XII. Components of the model are diagnosed and it is determined that the rotation speed of one of the four reaction wheels is low. The faulty rotation wheel is turned off and the blackboard is sent the message:

Fault detected in momentum wheel #2.

XIII. The Rule-based system would use data driven tactics in an attempt to solve the problem, such as recycling power to the wheel. If these attempts are unsuccessful, a message stating that it will be necessary to develop new algorithms to work around the fault would be sent to the mission controller at the ground station (modeled as "Human Intervention").

Future Improvements

Two extensions to the Synergistic Reasoning System are proposed: (1) the addition of more reasoning modules and (2) a feedback system to allow controlled modification of the modules based on past sessions. This extended system is shown in Figure 6.

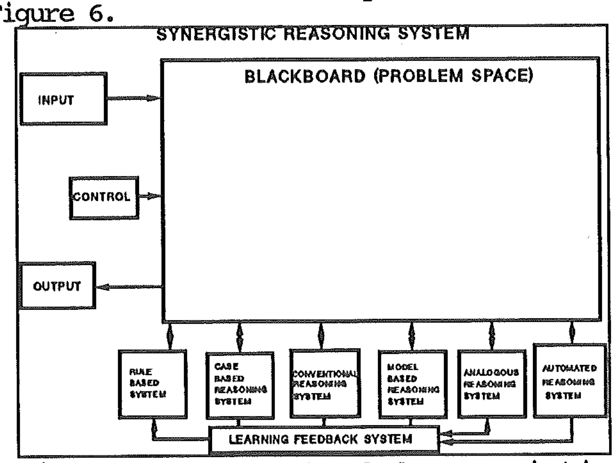

Figure 6. The Extended Synergistic Reasoning System.

The four reasoning methodologies used in this research can cover most common problem instances. To be successful, however, they require that either rules, past cases, algorithms, or models exist for the problem domain. To be successful in domains for which such information is not available more powerful reasoning methodologies are required. Two such reasoning methods are analogical reasoning and automated reasoning. While it is not expected that analogical and automated reasoning systems would be used as often as the other forms of reasoning, it is conceivable that they could prove valuable. For example, they could provide considerable value in deep space missions in which previously unencountered situations occur, and human intervention is hampered by the amount of time required by long distance communications.

The second extension is to modify the traditional architecture to allow the controlled modification of one reasoning module based on results from one or more of the other reasoning modules. This violates one of the defining features of the blackboard architecture which is that the knowledge sources are totally independent of each other, and cannot influence each other directly [11].

The controlled modification would be in the form of a feedback system that allow the contents of a module to be modified based on the results of previous sessions that involved other reasoning modules. For example, the system would allow a solution resulting from use of the model-based system to be added as a rule to the rule-based system. In addition, the results of any session (including unsuccessful sessions which required human intervention) would be stored as a new case in the case-based reasoning system.

This feedback system could also be used to improve the performance of the analogical reasoning system. When analogical reasoning is employed, the result is positive mappings (those attributes that are confirmed to correspond between source and target), negative mappings (those attributes that are confirmed to not correspond between target and source), and neutral mappings (those attributes whose correspondence has yet to be confirmed). The feedback system could postulate a theorem that a neutral attribute has a positive (or negative) mapping. It could then use the automated reasoner to prove or disprove the theorem, leading to an increase in confidence in, or denial of, our analogy.

Bibliography

1. Skinner, James M. A Diagnostic System Blending Deep and Shallow Reasoning, Proceedings of the Ninth International Workshop on Expert Systems and Their Applications. Avignon, France, 1989.

2. Luger, George and William A. Stubblefield. Artificial Intelligence and the Design of Expert Systems. Benjamin/Cummings. Redwood City, CA. 1989.

3. Searle, John. Minds, Brains, and Science. Harvard University Press, Cambridge, Mass, 1984.

4. Torrance, Steve. The Mind and the Machine - Philosophical Aspects of Artificial Intelligence. John Wiley & Sons, Chichester, 1984.

5. Weizenbaum, J. Computer Power and Human Reason San Francisco: Freeman, 1976.

6. Harmon, Paul and Brian Sawyer. Creating Expert Systems for business and Industry. John Wiley & Sons, Inc. 1990.

7. Iwasaki, Yumi. The Handbook of Artificial Intelligence Volume IV. Edited by Avron Barr et al. Addison-Wesley Publishing Company, INC. Reading, Mass. 1989.

8. DARPA. "Case-Based Reasoning from DARPA: Machine Learning Program Plan",Proceedings: Case-Based Reasoning Workshop. Morgan Kauffman Publishers, San Mateo Ca 1989.

9. Waterman, Donald, A Guide to Expert Systems. Reading MA: Addison Wesley Publishing Company, 1986.

10. Davis, Randall, et al., "Diagnosis Based on Description of Structure and Function" Proceedings of the National Conference on Artificial Intelligence, pp. 137-142, 1982.

11. Hayes-Roth, Barbara. "Blackboard Systems," The Encyclopedia of Artificial Intelligence. Edited by Shapiro. 1989.

12. Nii, Penny. The Handbook of Artificial Intelligence Volume IV. Edited by Avron Barr et al. Addison-Wesley Publishing Company, INC. Reading, Mass. 1989.

13. GAO Report to the Chairman, Subcommittee on Defense, Committee on Appropriations, House of Representatives, Military Space Operations - Shuttle and Satellite Computer Systems Do Not Meet Performance Objectives. August 1988.

14. GAO Report to the Chairman, Subcommittee on Defense, Committee on Appropriations, House of Representatives, Military Space Operations - Operational Problems Continue With the Satellite Control Computer System. August 1988.

15. Skinner, James M. and Richard Higgins Jr. Employing Artificial Intelligence to Meet Space Requirements, Proceedings of the Twenty-seventh Space Congress. Canaveral Council of Technical Societies. Cocoa Beach, Florida, 1990.

16. Jet Propulsion Laboratory. Autonomous Spacecraft Program - Technology Development Tasks for Spacecraft High-Level Autonomy. Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, 1985.