An integrated reasoner for diagnosis in satellite control

James M. Skinner

Air Force Office of Scientific Research
SNL-1600/P.O. Box 5800
Albuquerque, NM 87185
jmskinn@cs.sandia.gov

George F. Luger

Department of Computer Science
University of New Mexico
Albuquerque, NM 87131
luger@unmvax.unm.edu

ABSTRACT

An earlier paper[16] identified satellite control as one of ten space applications suitable for artificial intelligence techniques. In [Skinner & Luger 91], we outlined an approach to reasoning that integrates diverse reasoning paradigms for problem solving and discussed how such an approach could reduce the human intervention required for satellite control. Since that time, we have developed a prototype for the reaction wheel assembly of the Hubble Space Telescope to test our theories. This paper discusses (1) the domain of satellite control in general and the reaction wheel assembly in particular, (2) the four reasoning methodologies selected for integration in a reasoning system, (3) the integrating architecture, (4) a description of a sample operation, and (5) our conclusions.

1. THE DOMAIN: SATELLITE CONTROL

The domain for our research is satellite autonomy and control. Satellite control is a complicated, tedious, and labor intensive process. According to a 1989 GAO study, a staff of over 4,000 government and contract personnel is required to operate the Air Force Satellite Control Network consisting of fixed ground-based tracking stations, central control facilities, and communication links.[5] 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. However, the number of controllers supporting the network will at best remain constant while the level of expertise decreases due to retirements.[16]

Staffing requirements for satellite control can be reduced through satellite autonomy and improved control consoles. A 1986 Jet Propulsion Laboratory study identified several satellite subsystems well-suited for autonomous operations including: (1) guidance, navigation, and control, (2) power systems, (3) thermal control, and (4) payload management.[10] Improved control consoles can increase the power of the tools available to the satellite controller during normal operations as well as assist in the diagnosis of anomalies. This reduces the need to increase the number of controllers and enables operators to control different families of satellites without requiring extensive (and unrealistic) levels of training.

To examine the potential for automating the control and diagnosis of a space vehicle subsystem, we considered the Hubble Space Telescope (HST) Reaction Wheel Assembly (RWA). The function of the HST RWA is to point the Space Telescope at the proper area of the sky and keep the telescope locked onto its target. The RWA functions according to the principle of conservation of angular momentum. When the telescope is stationary, the reaction wheel moves at a small speed to counteract the torque caused by Earth's gravitational field. To move the telescope, the speed of the reaction wheel is increased, causing the telescope to spin in the opposite direction. When the telescope nears its proper orientation, the spin is reversed and the telescope slows down. There are four reaction wheels aboard HST, and the sum of the torque forces generated by these wheels enables the telescope to rotate about an arbitrary axis.[11]

Figure 1 is a cross-sectional view of the RWA. The outside shell consists of a metal casing, which mounts directly to the telescope bay walls. Inside the casing is a hollow aluminum wheel with a steel rim, mounted on a rotating shaft. The shaft is connected to a motor at the top of the assembly. The Rotor Control Electronics (RCE) and Power Control Electronics (PCE) supply power and control signals to the motor. Each end of the wheel shaft is supported by a bearing. The top bearing is called the motor-bearing, and the bottom bearing is called the RCE-bearing. Near each of the heat-generating components is a temperature sensor used to monitor the device's functioning.[11]

2. INTEGRATING REASONING PARADIGMS

Earlier research[15,17] strongly suggested that the best approach to many diagnostic problem-solving tasks may not be through any single method of reasoning, but rather by blending several reasoning methods together. In Section 3 we will introduce an architecture that integrates diverse reasoning methodologies in a manner that allows the system to benefit from the

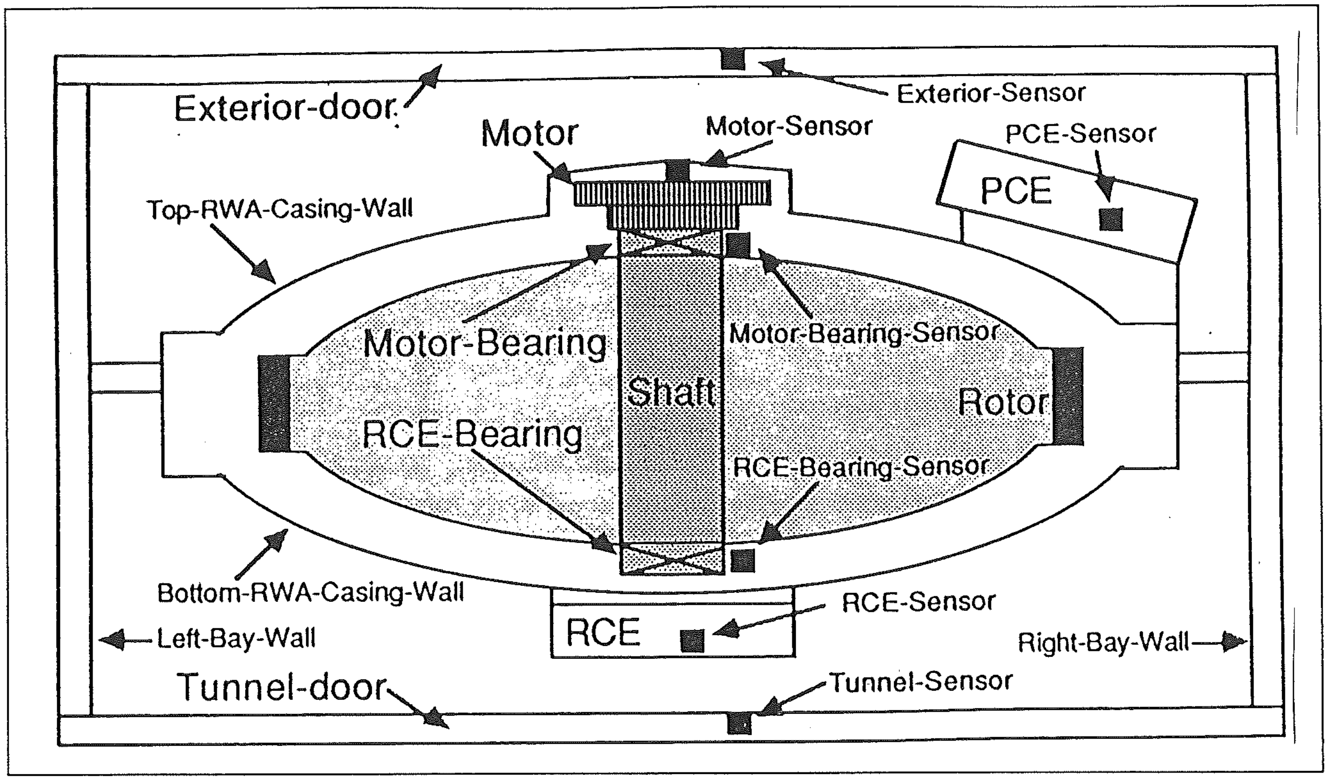

Figure 1. RWA Cross-sectional View.[13]

strengths of each, while minimizing their respective weaknesses. We selected four complementary reasoning methodologies for integration based on their potential utility in problem solving: conventional (or procedural), rule-based, model-based, and case-based reasoning. We present these paradigms next with examples of how the each paradigm applies to the HST RWA.

2.1 Conventional reasoning (CR)

Although conventional systems are not normally associated with reasoning, there are techniques which allow them to integrate reasoning with search and enable them to perform in an intelligent manner. Conventional and AI systems differ in their methods of representation and control. Conventional systems use data to represent information about the domain and use equations to perform mathematical computations. AI systems use strings of characters (known as symbols) to represent problem concepts and to apply various strategies to manipulate these concepts.[20]

Conventional systems rely on fixed algorithms to solve a problem while AI programs use heuristics. Heuristics are generally true and applicable in many situations but are not guaranteed to be universally valid. Thus, conventional programs are designed to produce the correct answer (if derivable) every time, while AI systems will occasionally produce incorrect answers. At first this would seem to give conventional programs a distinct advantage over expert systems. However, conventional programs for performing complex tasks will also make mistakes; if the expert provided the wrong rules, there is no reason to believe she would provide the correct algorithms. But the mistakes will be difficult to detect and remedy since the program's strategies, heuristics, and basic assumptions will not be explicitly stated in the program code.[20]

The primary advantage of conventional reasoning is its simplicity. If an algorithm that can be computed in non-polynomial time exists to solve the problem, a conventional approach is the most straightforward method of solving the problem. In addition, proven verification and validation techniques exist for testing conventional code.

Unfortunately, algorithms do not exist for all problems. In many cases, conventional programs lack the power required to solve the problem. Conventional programs cannot easily incorporate heuristics nor make inferences about a domain. Conventional systems also suffer from poor explanation capabilities. Numerical simulation describes behavior as a set of successive variable values but cannot provide causal explanations of why things behave as they do.[9]

Conventional programs are well-suited for repetitive processes in which the same actions are performed on a stream of data. AI programs are more suitable for making inferences about a domain. Consequently, conventional programs are an effective method of manipulating large databases, whereas AI programs are effective in the manipulation of large knowledge bases.[20]

For the conventional reasoner in the HST RWA prototype, the knowledge required is the set of control algorithms that are currently used onboard the vehicle. For purposes of the prototype, the CR module implements a simulator for the attitude control system that enables the satellite to maintain its correct position and attitude. The simulator allows the user to change the attitude of the satellite relative to the Earth or to change the path of the satellite around the Earth. The simulator fires the thrusters as necessary to achieve the new position and reflects the changes through graphics on the screen.

The CR module also provides an interface to allow the user to post messages concerning scheduled maintenance and to induce a fault in any of the components of the HST. The CR module provides access to the graphics facilities which allow the user to view the objects of the system, examine their slots, and follow links from one object to another.

2.2 Rule-based reasoning (RBR)

Rules are the most commonly used knowledge representation technique in AI. They provide a formal method of representing recommendations, directives, strategies, or heuristics.[12] A rule-based expert system consists of the knowledge base, the working memory, and the inference engine.

The knowledge base holds a set of production rules that embodies the expertise of the domain. Each rule is a condition-action pair that defines a single chunk of problem-solving knowledge. The condition part of the rule is a pattern that determines when that rule may be applied to a problem instance. The action part defines the associated problem-solving step.

The working memory contains a description of the current state of the world in a reasoning process. This description is a pattern that is matched against the condition part of a production to select appropriate problem-solving actions. When the condition element of a rule is matched by the contents of working memory, the action associated with that condition may then be performed.

The inference engine performs the function of the recognize-act control cycle. The recognize-act cycle matches the patterns in working memory against the conditions of the production rules, producing a subset of productions called the conflict set. One of the productions in the conflict set is then selected (through a process known as conflict resolution) and fired (the action of the rule is performed), changing the contents of working memory. The control cycle then repeats with the modified working memory. The process terminates when no rule conditions are matched by the contents of working memory.

The strength of rule-based systems lies in the simplicity of their design and maintenance. Rules can be easily constructed because experts tend to express most of their problem-solving techniques in terms of situation-action pairs 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 primary weakness of rule-based expert systems is their lack of robustness. This arises from the fact that only an implicit model of the domain 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 expert systems have proven successful in a number of important applications such as diagnosis, configuration, and process control. They are best suited for problems that are constrained and for which expertise exists. Good candidates for a rule-based approach are problems which can be best understood as consisting of many independent states as opposed to being understood through a concise, unified theory; problems with simple control flow; and problems for which the associated knowledge can be separated from the way in which it is to be used.[3]

The rules for the RWA are a set of diagnostic associations relating the readings of the temperature sensors to the possibility of faults in the bearings or electronics. An example of one such rule concerning the RCE bearing is:

IF Temperature of RCE-Bearing-Sensor is High, and
Temperature of RCE-Sensor is OK, and
Temperature of Tunnel-Sensor is OK
THEN Set Malfunction of RCE-Bearing to True.

Referring to Figure 1, this rule states that if the sensor for RCE-bearing is abnormally high, and nearby sensor readings are normal, then there must be a malfunction within the RCE-Bearing.[11]

2.3 Model-based reasoning (MBR)

MBR is based on representations of the physical structure of the domain. This technique employs an explicit model of the domain that depicts knowledge of the structure, function, behavior, and common misbehavior of a system. MBR uses causal reasoning, reasoning from first principles, and reasoning from the principle of locality in solving problems about the domain.[3]

Structural knowledge describes objects in the domain and their physical configurations. In most programs, the structure of a device is represented in terms of its components and the physical connections among components. Behavioral knowledge includes the laws of physics and the functions of subcomponents and can be represented by causal links connecting events, a sequence of states connected by causal links, or as qualitative mathematical relations among variables pertaining to various aspects of the objects in the domain.[9] It is often useful to include information about common misbehavior; that is, how a component usually fails. For example, since a faulty four-bit adder will most likely have an output of all zeros or all ones, it is unusual for an adder with an output of 0101 to be defective.

Causal reasoning relies on knowledge concerning how the behavior (or misbehavior) of one component affects the behavior of another component. Reasoning from first principles emulates the ability of an experienced engineer or technician to troubleshoot an unfamiliar device by referring to schematic drawings and applying appropriate physical laws. The principle of locality considers how components are connected (mechanically, electrically, physically) to determine how behavior of one component can be influenced by another component.[3]

MBR is usually implemented in an object-oriented manner. The domain knowledge required is a list of components, their connectivity, and threshold values for their output to be considered good or bad. The components of the domain are represented as objects in the coding environment. Connections between the components are represented by links between the objects in the code. In diagnosing a system, the path of a fault can be traced back to its source by following the link from a component's bad input to the output of the component that supplies the faulty signal.

The use of an explicit model of the domain lessens the limitations associated with rule-based reasoning. Because the system has fundamental knowledge of the domain, it can reason about situations that have not been previously encountered and for which the system has not been explicitly programmed. In addition, the knowledge can be represented in a format independent of the specific application, allowing the knowledge to be transferred to other tasks in the same domain. For instance, a model-based system 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 to design. Finally, MBR systems can use the explicit model of the structure, function, and behavior of a domain to provide causal explanations.

MBR systems tend to be more CPU intensive than RBR systems. A pure model-based system does not take advantage of the experiential knowledge of experts in the domain. In addition, MBR does not consider information that, while unrelated to the structure of the domain, is valuable in reasoning about the system (for example, what test was in progress when an error message occurred). Finally, in using the principle of locality for diagnosis, pure model-based search does not take into consideration the likelihood of a component's being at fault, nor does it consider how difficult testing a component may be. Many MBR systems include at least a few heuristics to overcome these weaknesses.

Diagnosis of physical devices is a good application area for MBR, especially if the device is a complex state-of-the-art component with little or no human expertise available. The explicit model of the domain allows the system to respond to situations that could not have been predicted. But development of the explicit domain model requires that the fundamental principles of the domain be well understood. MBR is not always appropriate in medicine, for example, where most of the knowledge used for diagnosing and treating diseases is empirical, rather than based on a model of the relevant biological and chemical mechanisms. In theory, however, MBR is applicable to a wider range of domains, including natural and social sciences, as long as some theories exist about causal mechanisms in the domain, from which a model can be constructed to explain and predict behavior.[9]

The primary knowledge source for the models used in the prototype of the SRS was a set of papers written by researchers from NASA Ames and Stanford[7,11] that cover the structural and functional models for the HST RWA. Secondary sources were used to provide details for constructing models. The structure and function of additional components on which the RWA depends were taken from a satellite design manual.[21] In addition, we held knowledge engineering sessions with a satellite operator[6] and a satellite analyst[2] to determine how failures in the system may reveal themselves as symptoms. The resulting set of models is a composition of the knowledge from these sources.

2.4 Case-based reasoning (CBR)

CBR uses the results of past problem-solving episodes to solve new cases. This approach increases the quality and efficiency of the reasoning by deriving shortcuts and by avoiding past mistakes. A CBR system can 'learn' by adding newly solved cases to memory, thus allowing the system to continually improve its performance.

The two main types of CBR are precedent-based and problem-solving. Precedent-based CBR, similar to legal reasoning, argues that a new situation should or should not be treated like a past case based on similarities or differences between the cases. Problem-solving CBR, used for tasks such as design or planning, formulates a solution suited to the new case by modification of past solutions.

CBR systems comprise a case memory, indices into the memory, metrics for assessing similarity, and procedures for adapting past cases to the new situation. Descriptions of the operation of CBR systems differ, but most are described as having five steps:[14] (1) recall relevant cases from case memory, (2) select the most promising case(s) to reason with, (3) construct a solution for the new case, (4) test and criticize the proposed solution, and (5) update memory/adjust indexing mechanisms.

Proponents of CBR claim several advantages. The shortcuts in reasoning and the capability of avoiding past errors enhance performance. No causal model of the domain is necessary. Because much of the knowledge required for CBR is in the form of cases, knowledge acquisition is easier than for other reasoning methods. Not only is CBR a natural way of codifying a response to a problem, many domains (e.g., medicine, law) have an existing case base that can be used as a seed.

The difficulties of CBR are the complexity of handling a large number of cases, the implementation details of CBR theory, and a lack of robustness due to the absence of fundamental knowledge of the domain. Because CBR systems tend to require a large number of cases, the cases must be organized and indexed to minimize search and implement efficient retrieval. Implementation requires algorithms for selecting the 'best' match, modifying a past case to a new situation, anticipating problems, repairing faulty cases, and testing the results.

Most successes in CBR have been in domains involving classification (e.g., medical, legal) and problem solving (e.g., design, planning, diagnosis).[1] CBR is best suited to domains in which many training cases are available, and where it may be difficult to specify appropriate behavior using abstract rules.

Cases for the RWA were constructed based on consultations with a satellite analyst.[2] A sample case is shown below. The case number is universal time, representing the number of seconds since midnight, January 1, 1900 GMT. The case number is used as a unique identification number and a means by which the CBR module can reason temporally during the follow-up phase, searching recent actual cases for trends in diagnosis.

```
Case-Number: 2902248000    ;;; Dec 20 1991 1500
Symptoms: ((:weak-signal)
           (:calibrate-pointing :unsuccessful))
Suspect: none
Actions: ((:cr  :symptom-posted :weak-signal)
          (:em  :check-prior-messages :none)
          (:rbr :adjust-antenna :unsuccessful)
          (:cr  :calibrate-pointing :unsuccessful)
          (:mbr :diagnose-acs :acs-faulty))
Diagnosis: attitude-control-system
Result: successful
Source: actual
```

3. AN ARCHITECTURE FOR INTEGRATING REASONING PARADIGMS

As was stated earlier, the central motivation for this research was the expectation that synergism could result from blending multiple reasoning methodologies in a proper framework. We designed such a framework by making a fundamental modification to the traditional blackboard architecture.

3.1 The traditional blackboard architecture

The blackboard model (Figure 2) consists of three components: the blackboard data structure, one or more knowledge sources, and the control component.[4] The blackboard data structure is a global data store that holds the problem-solving state data.

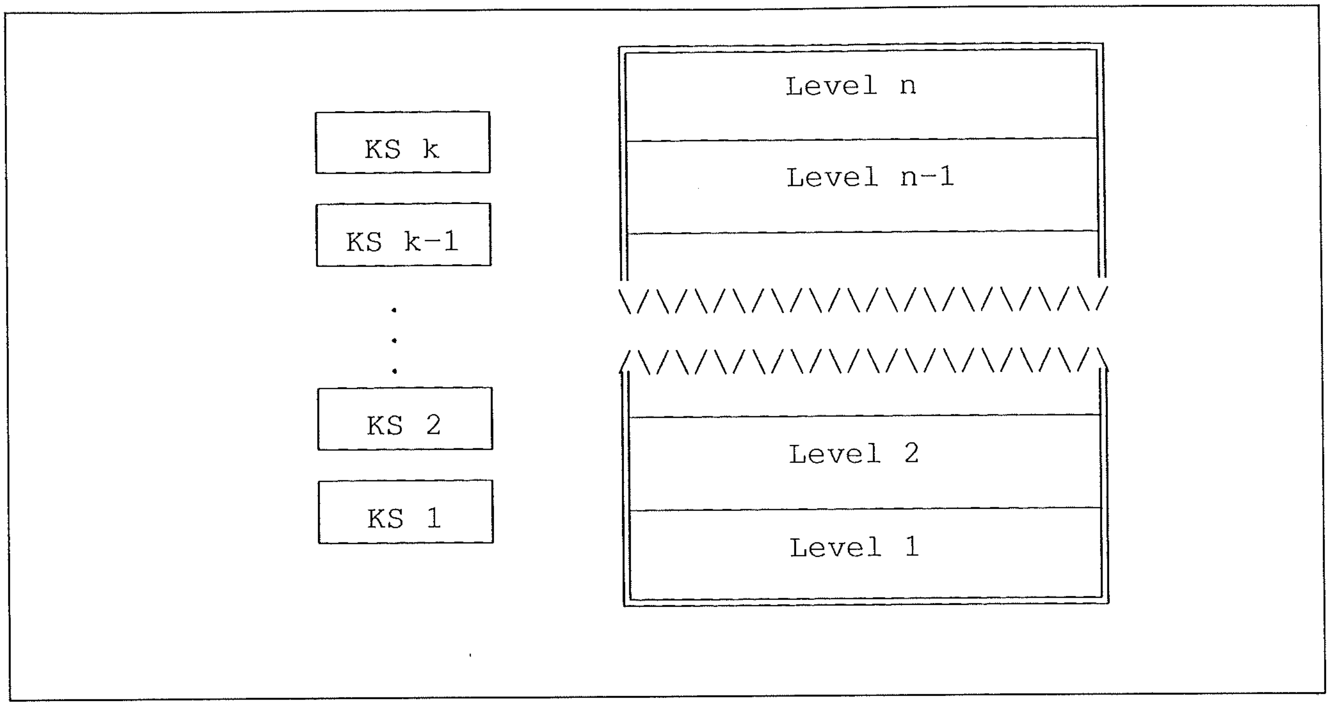

Figure 2. Traditional Blackboard Architecture.

The blackboard can be partitioned into distinct information levels to allow information about the problem to be represented at different levels of detail. For example, the first blackboard, Hearsay-II, used the levels to represent phrases, words, syllables, segments, and other building blocks of the spoken language. All communication and interaction of the system takes place solely through the blackboard.[4]

The domain knowledge needed to solve the problem is partitioned into independent knowledge sources. Knowledge sources produce changes to the blackboard by transforming information on one level to information on the same or other levels using algorithmic or heuristic rules. Typically, the knowledge is stored as precondition-action pairs.

Control can reside in the knowledge sources, on the blackboard, in a separate module, or in some combination. Although implementations vary, operation of the blackboard usually begins with a change to the blackboard caused by the posting of the initial problem statement. Each change to the blackboard constitutes an event that in the presence of 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. When a knowledge source is triggered, it will typically produce new blackboard events. These events may in turn trigger other knowledge sources.[8]

A common analogy used to describe the behavior of the blackboard model is that of a group of people trying to assemble a jigsaw puzzle.[4] Imagine a room with a large sticky blackboard and a group of people each holding pieces of a jigsaw puzzle. Volunteers first put their most promising pieces on the board. Each member of the group looks at his or her pieces to see whether any fit with the pieces already on the board. If so, those with appropriate pieces update the evolving solution. The new updates cause other pieces to fall into place, and other people add their pieces. The whole puzzle can be solved in complete silence - there is no need for direct communication between the individuals. The apparent cooperative behavior is mediated by the state of the solution on the blackboard. The solution is built incrementally and opportunistically.

3.2 The Synergistic Reasoning System

We designed an alternative problem-solving approach based on the blackboard problem-solving model which we call the Synergistic Reasoning System (SRS). The difference between SRS and the traditional blackboard model can be understood by contrasting analogies. An analogy for SRS is a person taking a closed-book test. All of the knowledge to be used during the test is self contained. However, the person is likely to use several different methods of reasoning while taking the test including relying on past experiences, employing heuristics, following procedures, or developing a mental model of a problem.

Implementing this approach requires a fundamental modification to the blackboard model. Rather than partitioning the domain knowledge functionally for the problem, the synergistic approach partitions the problem-solving approach in terms of reasoning methodologies. Instead of the traditional knowledge sources, the system has reasoning modules. Specifically, the synergistic reasoning system is equipped with four reasoning modules: case-based reasoning, rule-based reasoning, conventional reasoning, and model-based reasoning. The system dynamically switches between the reasoning modules as necessary to solve the problem. This approach produces a synergistic effect through *cooperation*, *confirmation*, *refutation*, and *follow-up*. These aspects of synergism will be demonstrated in Section 4. They are described in detail, along with the SRS control algorithm, in [Skinner & Luger 92] and [Skinner 92].

4. Sample Operation of SRS

A prototype of SRS for the RWA was implemented as a modified blackboard architecture with a hierarchical blackboard database, four reasoning modules (i.e., CBR, RBR, CR, and MBR), and an Executive (control) Module (EM). The blackboard database has one root blackboard and four blackboards as interior nodes (one each for the individual reasoning modules). The root blackboard has seven spaces: Status, Symptoms, Suspects, Actions, Diagnosis, Recommendations, and Goals. The interior blackboards each have a single space to record local information.

We developed a scenario to test the operation of SRS in which the onboard sensors detected a weak signal from the ground station. The response of the system is useful in depicting the four aspects of synergistic behavior. The state of the blackboard at various points is shown in the figures. An explanation of the events that led to these states follows.

4.1 Diagnosis of the RWA

The scenario begins during normal operations, with the CR module active. When the signal strength falls below a predetermined level, the CR module posts the symptom on the blackboard and surrenders control. The creation of a symptom causes the status to change to a fault condition which, in turn, triggers the EM.

After determining the symptom is not due to scheduled maintenance, the EM posts a goal to diagnose the symptom. The RBR responds using a set of rules that it has concerning the antenna adjustment which allow it to increase the gain by ten percent or to calibrate pointing. The RBR recommends an increase in gain which boosts the signal and alleviates the problem. The diagnosis is low-gain and (it would seem) the diagnostic session is complete. The posting of the diagnosis causes the EM to add three goals: confirm the diagnosis, refute the diagnosis, and follow-up on the diagnosis. This is SRS's method for error checking and increasing confidence in the conclusion. The state of the blackboard at this point is shown in Figure 3.

The CBR confirms the actions taken are the correct response for the given symptom by finding a past case which resulted in success. Next, the CBR attempts refutation, but cannot find any cases in which this tactic was unsuccessful. During follow-up, the CBR discovers that the gain has been increased twice in the last three hours. This trend, seen as indicative of a deeper problem, is posted as a new symptom and diagnosis is continued.

The RBR recommends calibrating the pointing of the antenna, but the CR module reports that calibration failed. This is added as a new symptom and causes the Tracking, Telemetry, and Control (TT&C) subsystem to be suspected. The state of the blackboard at this point is shown in Figure 4.

The MBR module constructs a model of the TT&C subsystem and checks each point, but no fault is found. At this point, the EM has no specific recommendations and must rely on the predetermined guidelines. The CBR module is used to search for past cases, retrieving a case in which the antenna could not be calibrated due to a fault in the attitude control system (ACS).

The MBR builds a model of the ACS and isolates the fault to the RWA. It cannot, however, find any malfunction in the components of the RWA model. The RBR module uses experiential knowledge to determine the faulty behavior is due to a high ambient temperature in the bay and recommends opening a louver to the outside to allow heat to dissipate; closed louver is posted as the diagnosis. Once again, confirmation, refutation, and follow-up are posted as goals.

No confirmation is found, but the CR refutes the diagnosis - according to its data the louver is open. The EM relies on the MBR to resolve the contradiction. In diagnosing a model of the thermal control system (which contains the louver), the MBR determines the input to the louver motor is good, but the louver is closed. The motor is determined to be bad and a backup motor is employed. The final state of the blackboard is shown in Figure 5. The results are stored by the CBR module, the blackboard is scrubbed, and control is returned to the CR module.

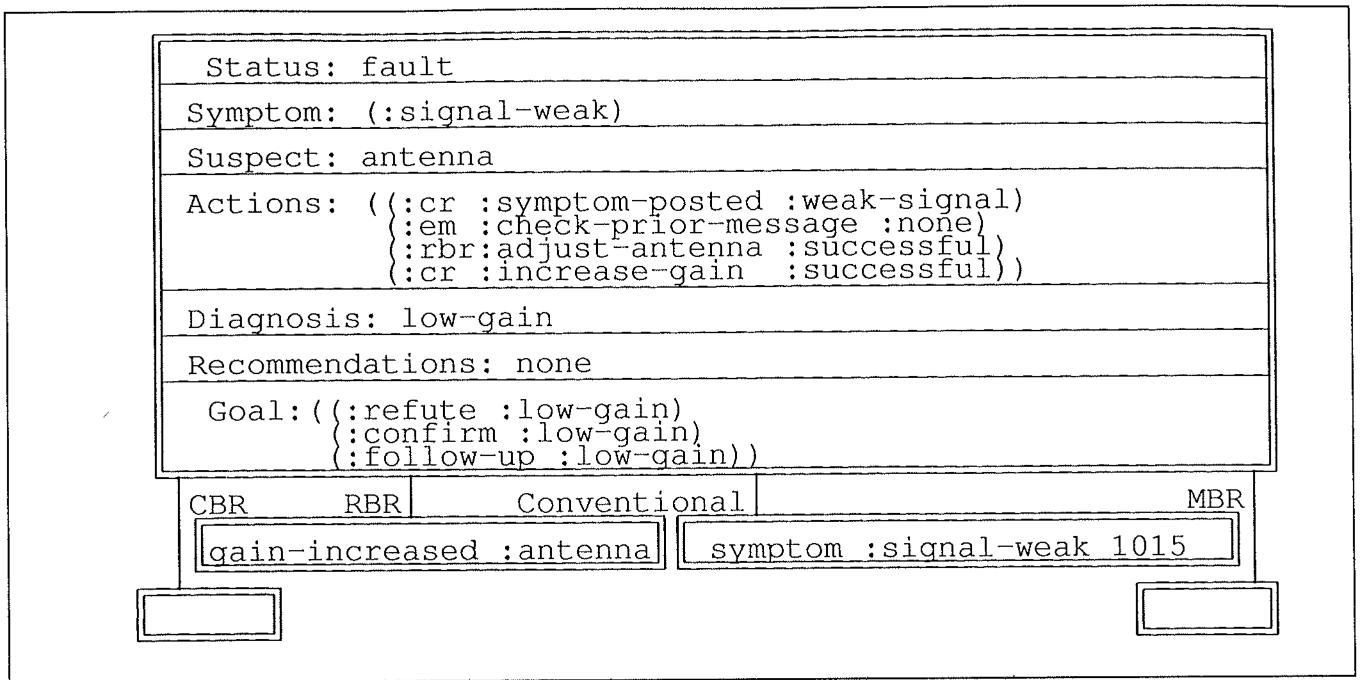

Figure 3. **Sample Operation.** The Upper portion shows the contents of the seven major spaces of the top-level blackboard. The lower four boxes reveal the contents of the individual modules.

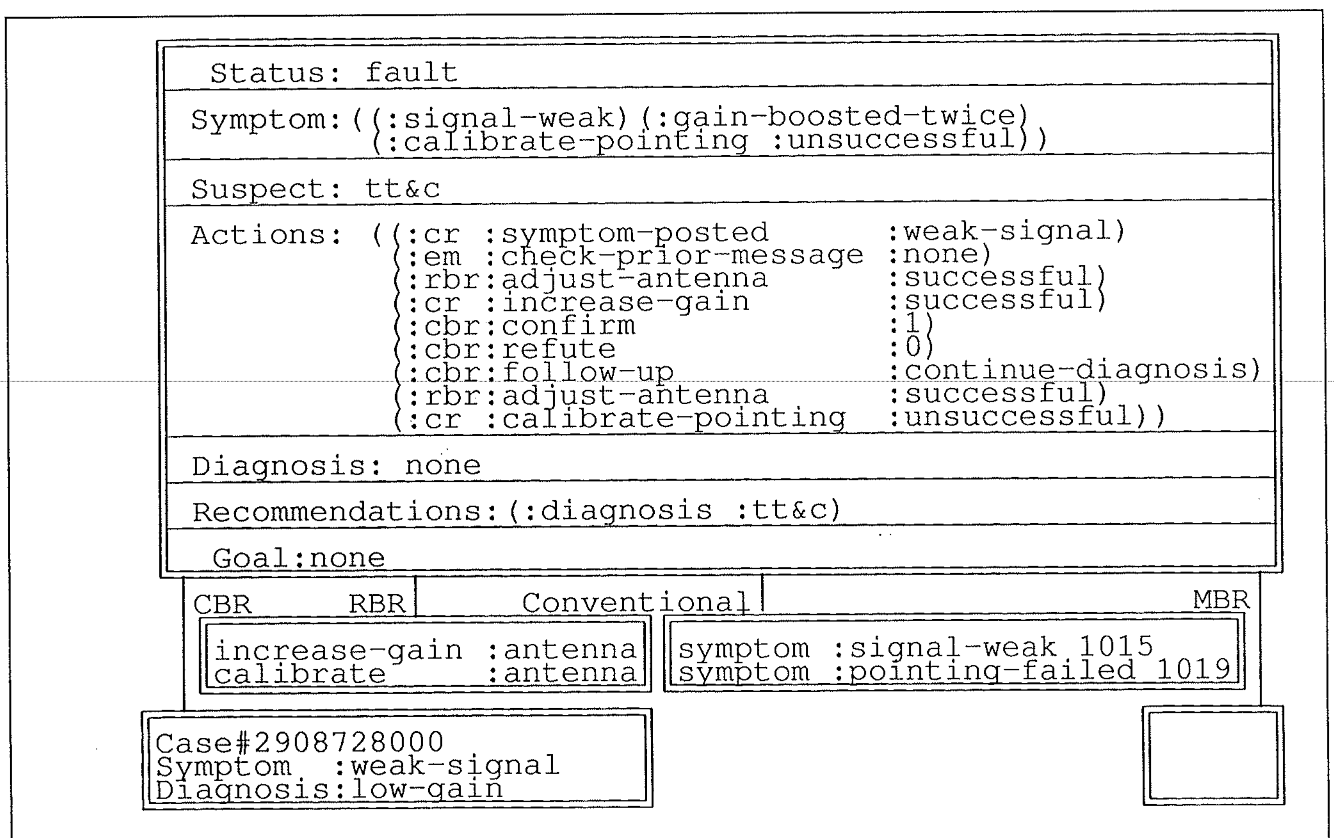

Figure 4. **Sample Operation (cont'd).** Through *follow-up* and *cooperation* additional symptoms have been identified. The CR module has recommended that the TT&C subsystem be diagnosed.

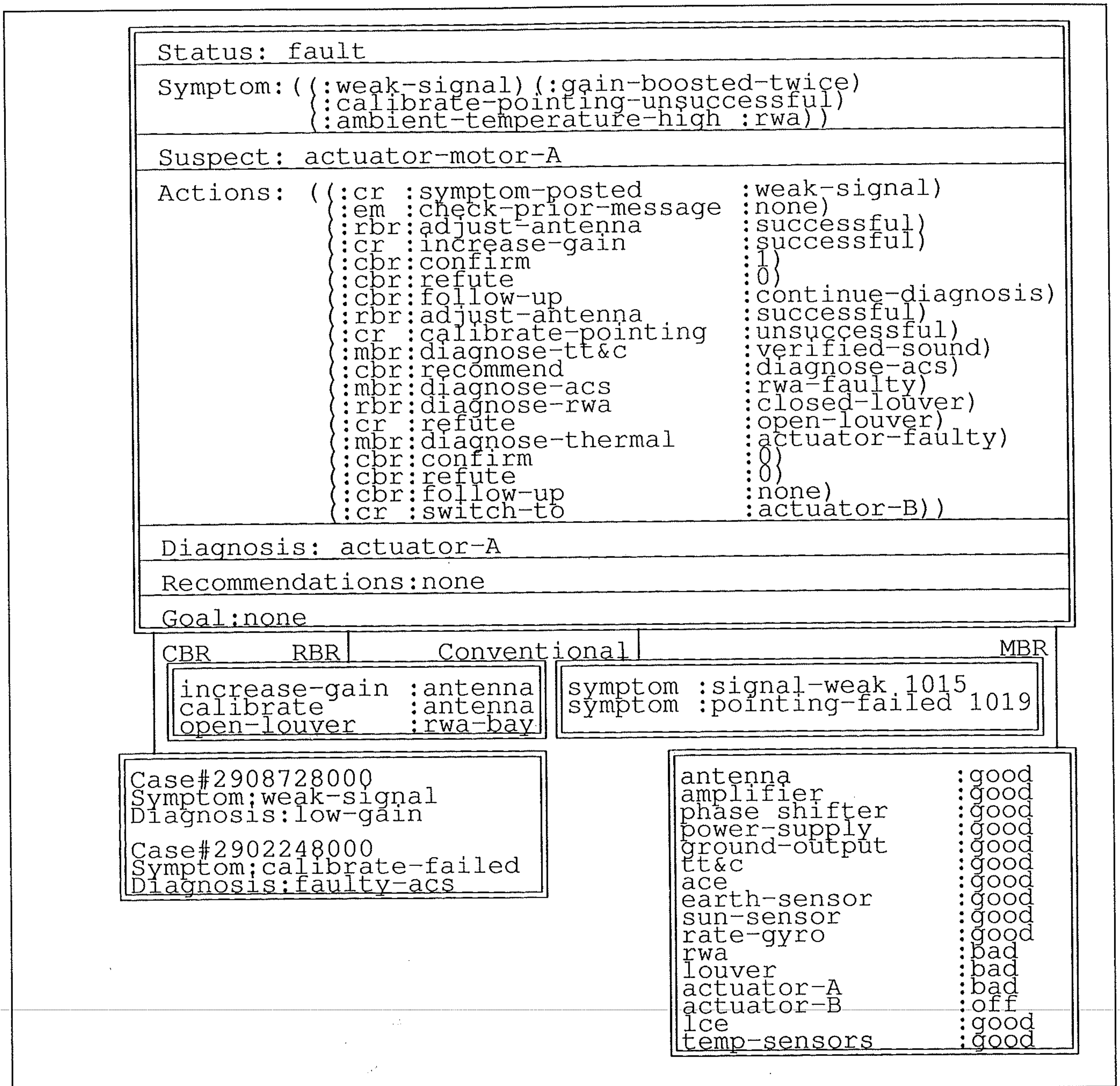

Figure 5. Sample Operation Final State. The fault has been isolated to the actuator motor. The results will be stored by the CBR module for use in future sessions.

4.2 Four aspects of synergism

This example illustrates the four aspects of synergistic behavior: *cooperation*, *confirmation*, *refutation*, and *follow-up*. While the original symptom was a weak signal, the actual cause was the failure of an actuator motor in the thermal control system. This failure caused the louver to remain closed, thereby overheating the reaction wheel assembly. This in turn prevented the attitude control system from maintaining the correct attitude, causing the antenna to be improperly calibrated. As a result, the signal strength continually degraded, and a weak signal was observed.

Individually, none of the reasoners would have responded with a correct diagnosis. Both the CBR and RBR modules would have attributed a weak signal to low-gain. The CR module had no algorithm to solve the problem. The MBR module would have diagnosed the TT&C model, only to find all components were sound. Yet, the reasoning modules were able to produce a proper response by collaborating on the problem.

Cooperation is the ability to construct a solution from partial postings. *Cooperation* was apparent as the reasoning modules worked together to isolate the problem. The CBR used historical knowledge to determine the inability to calibrate the antenna could be due to a fault in the ACS. The MBR used structural knowledge of the ACS to isolate the problem to the RWA, but, since heat flow was not included in the model, was unable to determine the cause of the faulty behavior. The RBR used experiential knowledge to identify the source of the problem as a closed louver.

Confirmation, the ability of one reasoning module to verify the results of another module, was demonstrated by the use of the CBR module to increase the confidence of the decision to increase the gain of the antenna. While this was only a temporary fix, it was the correct response for the available information.

Refutation is the ability of one reasoning module to refute the conclusions of another module. This feature was exercised when the CR reported that the louver was already open. The RBR module had incomplete knowledge of the current configuration of the system, leading to an erroneous conclusion that the louver was closed. The additional information provided by the CR led to mediation of the contradiction by the MBR.

Finally, *follow-up* is the ability to identify trends indicative of deeper problems. This aspect of synergistic behavior occurred when the CBR noted the repeated gain increase. Had this not been noted as a symptom of a deeper problem, an autonomous system might have continued to increase the gain, without addressing the underlying thermal problem which could eventually cause permanent damage.

5. Conclusions

We have designed an architecture that allows diverse reasoning paradigms to be integrated in a cohesive manner. We have enhanced the advantage of this integration by selecting four reasoning methods that are complementary in that they provide a convenient manner to gather and represent diverse knowledge in a complementary fashion. During the knowledge engineering phase of development the use of multiple approaches allows the problem to be viewed from many angles, resulting in a more complete picture of the domain. During execution, the system employs this diverse knowledge in a collaborative fashion to capitalize on the collective advantages of the methods, while diminishing the effect of their individual weaknesses.

The combination of the paradigms provides an ability to employ historical, experiential, procedural, causal, and structural knowledge during a problem-solving session and thus enables SRS to solve all problems solvable by any of the four reasoning methodologies individually. The control guidelines developed from established principles of blackboard control and our research of reasoning characteristics allow the system to produce a synergistic effect through *cooperation*, *confirmation*, *refutation*, and *follow-up*. The prototype demonstrated this synergistic effect by solving a problem that none of the individual reasoning methodologies could solve.

The application of the prototype to a subsystem in a space vehicle demonstrates the applicability of this approach to satellite control specifically, and to diagnostics in general. While SRS is an efficient and robust problem-solving model that can be applied to any domain suitable for one or more of the four reasoning methodologies employed, it is demonstrated here as an effective measure to cope with the complex domain of satellite control.

6. REFERENCES

1. Ashley, Kevin D. and Edwina L. Rissland, "Compare and contrast, a test of expertise," Proceedings of AAAI-87, pp. 273-278, August 1987.

2. Campbell, William S., Satellite Analyst, Personal interviews, Kirtland AFB, NM, March 1992.

3. Davis, Randall. "Diagnosis via causal reasoning: paths of interaction and the locality principle," Artificial Intelligence in Maintenance, edited by J. Jeffrey Richardson, pp. 102-122, Noyes Publications, Park Ridge, NJ, 1985.

4. Engelmore, R.S. and A.J. Morgan, Blackboard Systems, Addison-Wesley Publishing Company Inc., Wokingham, England, 1988.

5. 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 1989.

6. Garnham, John, Satellite Operator, Personal interviews, Kirtland AFB NM, October 1990.

7. Gruber, Thomas and Yumi Iwasaki, "How things work: knowledge-based modeling of physical devices," KSL 90-51, Stanford University, Stanford, CA, August 1990.

8. Hayes-Roth, Barbara et al., "BB1: An implementation of the blackboard control architecture," Blackboard Systems, edited by Robert Engelmore and Tony Morgan, pp. 231-244, Addison-Wesley Publishing Company, Inc., Wokingham, England, 1988.

9. Iwasaki, Yumi, "Qualitative physics," The Handbook of Artificial Intelligence Volume IV, edited by Avron Barr et al, pp. 323-413, Addison-Wesley Publishing Company, Inc., Reading, MA, 1989.

10. Jet Propulsion Laboratory, Autonomous Spacecraft Program - Spacecraft High-Level Autonomy Study (Final Report), Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 1986.

11. Keller, Richard et al., "Model compilation: an approach to automated model derivation," NASA report RIA-90-04-06-1, April 1990.

12. Luger, George F. and William A. Stubblefield, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Benjamin Cummings, Redwood City, CA, 1993.

13. Perkins, W.A. and A. Austin, "Experiments with temporal reasoning applied to analysis of telemetry data," Space Station Automation III, Vol. 851, pp 39-46, Society of Photo-Optical Instrumentation Engineers, 1987.

14. Rissland, E.L. and D.B. Skalak. "CABARET: rule interpretation in a hybrid architecture," International Journal of Man-Machine Studies 34, pp. 839-887, 1991.

15. Skinner, James M., A Diagnostic System Blending Deep and Shallow Reasoning, MSCE Thesis, AFIT/GCE/ENG/88D-5, Air Force Institute of Technology, OH, December 1988.

16. Skinner, James M. and Richard Higgins Jr, "Employing artificial intelligence to meet space requirements," Proceedings of the Twenty-seventh Space Congress, pp. 9-14, April 1990.

17. Skinner, James M. and George F. Luger, "A synergistic approach to reasoning for autonomous satellites," Proceedings of the Advisory Group for Aerospace Research and Development (AGARD) Panel on Machine Intelligence for Aerospace Electronic Systems, 1991.

18. Skinner, James M., A Synergistic Approach to Reasoning, Ph.D. Dissertation, Department of Computer Science, University of New Mexico, Albuquerque, NM, May 1992.

19. Skinner, James M. and George F. Luger, "An architecture for integrating reasoning paradigms," Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference (KR92), Morgan Kaufmann, San Mateo, CA. 1992.

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

21. Wertz, James R. and Wiley J. Larson, Space Mission Analysis and Design, Kluwer Academic Publishers, Dordrecht, Netherlands, 1991.