

Principles of
Knowledge
Representation
and
Reasoning

Proceedings of the
Third International
Conference

Cambridge, Massachusetts
October 25–29, 1992

Edited by:
Bernhard Nebel
Charles Rich
William Swartout

With support from the American Association for Artificial Intelligence, the European Coordinating Committee on Artificial Intelligence, and the Canadian Society for Computational Studies of Intelligence; in cooperation with International Joint Conferences on Artificial Intelligence, Inc.

An Architecture for Integrating Reasoning Paradigms

James M. Skinner
 Air Force Office of Scientific Research
 SNL-ISRC/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

The objective of this research is to determine what degree of synergistic behavior can be achieved from combining reasoning methodologies in a proper framework, where the strengths of one methodology compensate for the weaknesses in another, and result in a level of performance not achievable by any of the methods individually. We selected four complementary reasoning methods (case-based reasoning, rule-based reasoning, procedural reasoning, and model-based reasoning) for research. The integrating architecture modifies the traditional blackboard problem-solving model to allow multiple reasoning approaches to be combined. A control algorithm for the system is derived from heuristics for employing each of the individual reasoning methods and established blackboard control principles. A prototype demonstrates the production of a synergistic effect by diagnosing faults in a subsystem of the Hubble Space Telescope. Four aspects of the synergism are noted: cooperation, confirmation, refutation, and follow-up. We define these terms and discuss the power gain possible with a integrated reasoning approach to a problem-solving task.

1 Introduction

Earlier research [Skinner 88, Skinner & Luger 91] strongly suggested that the best approach to many problem-solving tasks may not be through a single method of reasoning, but rather by allowing several reasoning methods to be blended together. This view is compatible with that of researchers in the larger field of hybrid representation in which systems employ two or more integrated subsystems, each with distinct representation languages and inference systems. Researchers often cite the ease of expression and increase

in efficiency from allowing specialized languages as the major advantages of using hybrid representation [McSkimin & Minker 79, Cohn 89, Frisch 89].

The blending of reasoning methodologies raises questions about the relationship between diverse knowledge representations and the control of an architecture for their integration. In our current research we analyze selected reasoning methodologies and use this analysis to design a system that benefits from their individual strengths while minimizing their respective weaknesses. The belief is that combining reasoning methodologies in the proper framework can result in synergism. We have designed such a framework through modifications to the traditional blackboard problem-solving model.

We developed a prototype for diagnosing faults in the Hubble Space Telescope Reaction Wheel Assembly and observed synergism through interactions best described as cooperation, confirmation, refutation, and follow-up. The remainder of this paper provides an analysis of the reasoning methods employed, a description of the prototype, an explanation of the interactions observed, and an analysis of their benefits.

2 Synergistic Reasoning

Synergistic reasoning occurs when a system employing multiple reasoning methodologies is able to solve problems that cannot be solved by any single method. To develop such a system it is necessary to select reasoning methods that are complementary, as opposed to redundant, and to design and develop a structure capable of supporting their use in an opportunistic manner. The four reasoning approaches selected for integration are: case-based reasoning (CBR), rule-based reasoning (RBR), conventional (or procedural) reasoning (CR), and model-based reasoning (MBR). We designed an architecture for integrating the reasoning methodologies by modifying the traditional blackboard architecture. We call the resulting system the Synergistic Reasoning System (SRS).

The difference between SRS and the traditional blackboard model can be understood by contrasting analogies. A common analogy used to describe the traditional blackboard model is that of a group of people trying to assemble a jigsaw puzzle on a large sticky blackboard. Each member of the group looks at his or her pieces to see whether any fit with the pieces already on the blackboard. If so, those with appropriate pieces go up to the blackboard and update the evolving solution. The new updates cause other pieces to fall into place, allowing additional pieces to be added. The entire 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 [Engelmore 88, Luger & Stubblefield 93].

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. These approaches roughly correspond to the machine reasoning methods of CBR, RBR, CR, and MBR respectively.

Implementing this approach requires a fundamental modification to the blackboard model. Rather than partitioning the domain knowledge functionally into knowledge sources, SRS segments the problem-solving approach into reasoning modules, with each individual module employing one of the reasoning methodologies. 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*. Cooperation allows the individual reasoning modules to post partial solutions, enabling the system to solve problems that could not be solved by any single module. Thus, one module posts a partial solution not obtainable by any of the other modules, and while this module might not be able to generate the entire solution, one of the remaining modules, also unable to generate the desired solution from the original problem, is able to do so based on this new result.

Confirmation allows reasoning modules to verify results from other modules. As an example, when the RBR module recommends a tactic for solving a problem, the CBR may be able to provide past cases in which the tactic was successful. Confirmation is used to increase confidence in the conclusion or to choose between two competing tactics for problem solving.

Refutation is the ability of one reasoning module to refute conclusions of another module. That is, while incomplete information may cause one reasoning module to arrive at an incorrect conclusion, a second module may have information that disputes this conclusion.

Using the same example as above, the CBR may be able to demonstrate that past attempts at solving the problem with the tactics proposed were unsuccessful. Again, this is used to increase confidence in a conclusion or to select between competing proposals.

Follow-up searches for trends in the conclusions of the system indicative of deeper problems. As an example, CBR may be used to detect repeated adjustments to a system that alleviates a problem only temporarily. This repeated occurrence of the problem is then seen as symptomatic of a deeper problem.

3 A Survey of the Selected Reasoning Methodologies

We wished to design an architecture that capitalizes on the strengths of each reasoning methodology while compensating for their individual weaknesses. As an initial step, each of the four selected reasoning methodologies were evaluated; a summary of their characteristics is shown in Table 1. A detailed analysis and explanation of the table can be found in [Skinner 92].

Obviously this is a partial list, one which will grow as research continues. The table reveals advantages unique to each individual reasoning method. Specifically, CBR employs historical knowledge and offers shortcuts, error checking, and insight. RBR employs experiential knowledge and offers speed, high performance in a limited domain, and modularity. Conventional reasoning employs procedural knowledge and offers simplicity, correctness, and verifiability. MBR employs structural knowledge and offers robustness, transferability, and causal explanations.

The disadvantages of each reasoning methodology in Table 1 can often be compensated for by one of the other reasoning modules. Cases can supplement incomplete or inconsistent rules by providing exceptions, interpretations, examples, and explanations; and can reduce the time requirements for MBR by recording results or explanations for future use. Rules can improve performance of CBR in almost all aspects of the process (i.e., bootstrapping, anticipating problems, indexing, modifying cases, verifying solutions) and can enable a MBR system to respond faster, search models more efficiently, include experiential knowledge, and focus reasoning. Models can improve CBR with causal explanations and can improve the robustness and explanation capabilities of RBR. In addition, each methodology can act as a backup in case of failure of the other methodologies.

Table 1: Characteristics of Reasoning Methodologies (Summary).

Method	Advantages	Disadvantages
CBR	ability to employ historical knowledge allows shortcuts in reasoning avoids past errors no domain model required existing cases for some domains knowledge acquisition relatively easy coding relatively easy clever indexing can add insight	lacks fundamental knowledge of domain complexity issues with large case base hard to define criteria for matching hard to define criteria for indexing difficult to construct/maintain index
RBR	ability to employ experiential knowledge modularity eases construction & maintenance high performance possible in limited domain simple method of providing explanations rules map naturally onto search space rules are easier to trace and debug steps in process are open to inspection separation of knowledge/control	lacks fundamental knowledge of domain cannot solve unforeseen problems rapidly degrades near edges of domain explanations often inadequate knowledge is task dependent difficult to verify heuristics multiple experts may disagree
CR	ability to employ procedural knowledge correct answers when problem is constrained proven V&V techniques exist simple implementation	must have algorithm for task difficult to incorporate heuristics
MBR	ability to employ structural knowledge robust knowledge transferable between tasks can provide causal explanations versatile	lacks experiential knowledge of domain CPU/ time intensive requires an explicit domain model

4 Controlling Multiple Reasoning Paradigms

We derived a control algorithm suitable for a synergistic approach by combining principles for controlling blackboards with the findings from the survey of the reasoning methodologies. This algorithm is exercised by an Executive Module (EM) which is responsible for coordinating the problem-solving process.

In blackboard terms, scheduling knowledge sources to minimize the number of steps in a problem-solving session is known as the *focus of attention*. The developers of HEARSAY-II identified five fundamental principles for controlling the focus of attention [Hayes-Roth 77]. While these principles are defined in terms of knowledge sources, we have adapted them to the control of reasoning modules. The principles are:

(1) *The competition principle*: the best of several local alternatives should be performed first. This governs behavioral options which are locally competitive in the sense that a definite outcome of one may obviate the others.

(2) *The validity principle*: knowledge sources operating on the most valid data should be executed first. Everything else constant, the preferred knowl-

edge source should be the one working with the most credible data.

(3) *The significance principle*: knowledge sources whose responses are most important should be executed first. This principle ensures the most important steps are performed first.

(4) *The efficiency principle*: knowledge sources which perform most reliably and inexpensively should be executed first.

(5) *The goal satisfaction principle*: knowledge sources whose responses are most likely to satisfy processing goals should be executed first.

When applied in the context of SRS, these principles led to the following set of general heuristics. From Principle (1), recommendations should be followed in order of their specificity, likelihood, and frequency. From Principle (2), recommendations should be executed according to their confidence values. From Principle (3), suspected catastrophic or time critical recommendations should be capable of preempting other tasks. From Principle (4), the most efficient reasoning modules should be used first. From Principle (5), top-level goals should have priority over sub-goals.

Additional heuristics derived from the advantages of each methodology as given in Table 1 suggest that:

(1) CR should be used whenever a polynomial-time algorithm exists; (2) CBR should be used if no specific recommendation is present, and as a means of error checking; (3) each of the reasoners should be used as a failure backup to the others; (4) RBR should be used for quick fixes, if no causal explanation is required, or if time constraints are strict; (5) MBR should be used if a causal explanation is required, but only if adequate time is available; and (6) the results of the sessions should be stored by the CBR module.

The resulting guidelines for a diagnostic application are shown below grouped by the principle from which they were derived. These guidelines are by no means static - the intent is for the set to grow and to be refined as dictated by results of continuing research in integrating reasoning paradigms. While no priority is intended, we chose to implement them in the order of appearance.

From the efficiency principle:

- (a) If two reasoning modules can perform a task, choose the most efficient for that task.
- (b) If status is nominal, employ the CR module.
- (c) If more than one reasoner can act on a goal, employ in the order of RBR, CBR, MBR.
- (d) If no specific recommendation is present, employ the CBR module.
- (e) If no specific recommendation exists & CBR fails, employ the MBR module.
- (f) If no recommendation exists & CBR, MBR fail, employ the RBR module.

From the competition principle:

- (g) If multiple components are suspected, diagnose the most specific (lowest level).
- (h) If multiple recommendations exist, perform the one closest to isolating a fault.
- (i) If multiple components are suspected, diagnose the least reliable.
- (j) If multiple components are suspected, diagnose the one with the most recommendations.

From the significance principle:

- (k) If a symptom could be catastrophic, diagnose that symptom first.
- (l) If a recommendation is time critical, perform it first.

From the strengths of the individual methodologies:

- (m) If time is constrained, employ RBR.
- (n) If no causal explanation is required, employ RBR.
- (o) If causal explanation required & time allows, employ MBR.
- (p) If the diagnosis session is complete, employ CBR to store session results.

From the goal satisfaction principle:

- (q) If multiple goals exist, act on the top-level goals first.

From results of current research:

- (r) If a fault has been diagnosed, initiate confirmation, refutation, & follow-up.

Guideline (a) states the most efficient and reliable reasoning module will be used to solve a problem; this is the general case for all reasoning modules. Guidelines (b)-(f) implement (a) for ordering the recommendations of the four selected reasoning modules and selecting between competing modules to achieve goals. Guideline (b) is the specific case in which no fault has occurred. In this case, conventional algorithms exist capable of handling the situation at a lower cost (in terms of time and space requirements) and with a higher reliability than any other reasoning method.

Guideline (c) presents the criteria for choosing between competing reasoning modules to perform a task. The priority used is to rely on RBR, then CBR, then MBR. In general, robustness increases and efficiency decreases in order of CBR, RBR, and MBR. By favoring RBR, a balance between robustness and efficiency is achieved. CBR is selected second because it can be executed quickly.

Guidelines (d), (e), and (f) handle the situation when no specific recommendations are present. Under these circumstances, the reasoning modules are prioritized as CBR first, then MBR, then RBR. The Executive Module forms a goal for the CBR to match on the list of symptoms in the current session; the CBR module returns a recommendation to diagnose the faulty component from a similar past case. Next, a goal is set for the MBR to diagnose each of the components in the list of suspected components (if the list is empty, a goal is created to diagnose the model of the entire system). Finally, a goal is created for the RBR module to diagnose the current list of symptoms and return appropriate recommendations. The rationale behind the prioritization is that the information available favors CBR over MBR, and MBR over RBR. At least one symptom is guaranteed to be present (otherwise the CR module would be in control), and the case base is indexed by symptoms. The list of suspects provides a focus for the MBR to diagnose the fault. While the RBR may be able to diagnose the symptom, it was unable to do so with the information available at the time the symptom was first recorded.

Guidelines (g) and (h) are implemented by tracking the level of the subcomponent suspected. The entire system under diagnosis is designated Level 1 with all direct subcomponents assigned to Level 2, and in general, subcomponents of a component on Level n are assigned Level $n+1$. Under this scheme, a fault isolated to Level n is at a lower level and more specific than a fault isolated to Level $n-1$. For purposes of this

research, a heuristic is employed that faults isolated to a lower level are closer to isolating the fault; this is strictly true only if all subcomponents have the same number of levels.

Guideline (i) advises that the least reliable of the components suspected be diagnosed first because this is the component most likely to be the cause of the fault. This is implemented by comparing the values of the Reliability slots of the components. Reliability is expressed as the mean time between failure, in hours, for the component.

Guideline (j) favors the most frequently proposed recommendation. It is implemented by counting the number of occurrences pending for each recommendation and executing the recommendation with the highest number of occurrences.

Guidelines (k) and (l) are from the significance principle; catastrophic or time critical events should be handled first. Guideline (k) is implemented by diagnosing catastrophic symptoms first. Guideline (l) is implemented by considering time-critical recommendations first. Both catastrophic symptoms and time-critical recommendations are application dependent and determined *a priori*.

Guidelines (m)-(o) arise from the strengths of the reasoning methodologies and require knowledge of the user's desires. They are implemented through messages posted on the blackboard. That is, a user may post that a causal explanation is required or that time is constrained on the blackboard. The default values are that a causal explanation is not required and time is not constrained.

Guideline (p) dictates the problem-solving information be stored for use in future diagnostic sessions. This is accomplished by the CBR module.

Guideline (q) is due to the goal satisfaction principle. Goals in the system are stored hierarchically; a top-level goal may have sub-goals. This guideline is implemented by acting on top-level goals first (i.e., goals without links to higher level goals).

Guideline (r) is a result of this research and the discovery of how SRS can produce synergism. It is implemented by posting the three goals when a diagnosis is reached. Each module then responds according to its ability.

5 The SRS Prototype

We constructed a prototype of the Synergistic Reasoning System for diagnosing faults in 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 [Keller 90].

5.1 Structure for a SRS Prototype

The SRS prototype was implemented in a commercial shell known as the Generic Blackboard (GBB), a toolkit based on the Common LISP Object System (CLOS) [BBT 91]. GBB provides the facilities required to construct a typical blackboard application including the blackboard database, knowledge sources, and the control shell.

SRS is a modified blackboard architecture with a hierarchical blackboard database, four reasoning modules (i.e., CBR, RBR, CR, and MBR), and an Executive (control) Module. 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.

The CBR module is implemented through GBB's pattern-matching facilities. We currently maintain a case base that includes a case number, the source of the case (either actual or hypothetical), the list of symptoms, the suspected components, the diagnosis, the list of actions taken to correct the fault, and the result (success or failure). The identification number comes from a LISP function call to universal time; the case number therefore serves not only as a unique identification number, but a means by which the CBR module can employ temporal reasoning during follow-up.

The RBR module uses the embedded GBB/OPS inference engine as a means of implementing a rule-based system. The RBR module treats the symptoms and goals of the problem-solving session as facts, asserting them into its knowledge base. It begins a data-driven inference resulting in the creation of recommendations or actions to be taken. As it fires each rule, the RBR module records its consequence on the RBR blackboard.

While the CR module can be involved in the diagnosis process, its primary purpose is to reason about the domain in the absence of any faults. During normal operation, the CR module posts messages from the user on the blackboard concerning expected out-

ages or anomalies. When a fault is detected, the CR module posts the symptoms on the blackboard and surrenders control. The CR module for our prototype is implemented in CLOS.

The MBR module diagnoses the suspected components to determine the likely cause of the symptoms. The MBR module is implemented in CLOS and uses the principle of locality. This principle considers how components are connected (mechanically, electrically, physically) to determine how behavior of one component can be influenced by another component [Davis 85].

The Executive Module (EM) exercises explicit control over SRS by determining the order in which the reasoning modules work on the problem and coordinating the problem-solving process. It is implemented through a combination of GBB's control shell, knowledge sources, and CLOS.

5.2 Cases for the Domain

Cases were constructed based on consultations with a satellite analyst [Campbell 92]. 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 will be 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
```

5.3 Rules for the Domain

The rules 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 rotor control electronics (RCE) 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.
```

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 [Keller 90].

5.4 Procedures for the Domain

As noted in the section on control guidelines, the conventional reasoner is responsible for reasoning about the environment as long as the status of the system is nominal. For the HST RWA, 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 acts as the interface between the user and SRS. It allows the user to post messages concerning scheduled maintenance on the blackboard and to induce a fault in any of the components of the HST. The CR module provides access to the GBB graphics facilities which allow the user to view the objects posted on the blackboard, examine their slots, and follow links from one object to another.

5.5 Models for the Domain

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 [Keller 90, Gruber 90] 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 [Wertz 91]. In addition, we held knowledge engineering sessions with a satellite operator [Garnham 90] and a satellite analyst [Campbell 92] 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.

6 Sample Operation of SRS

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.

6.1 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 1.

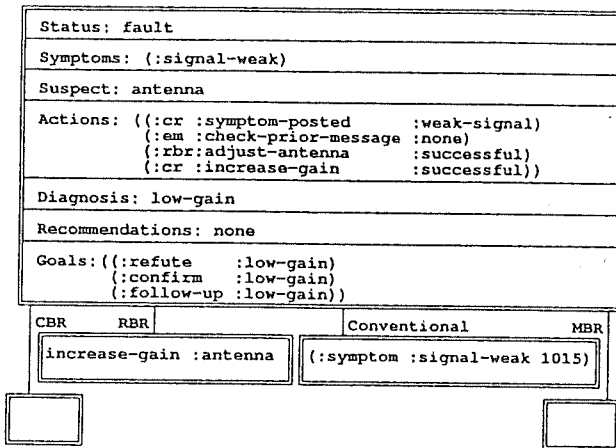


Figure 1: Sample Operation. The upper portion of the figure shows the contents of the seven major spaces of the top-level blackboard. The lower four boxes reveals the contents of the individual modules.

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

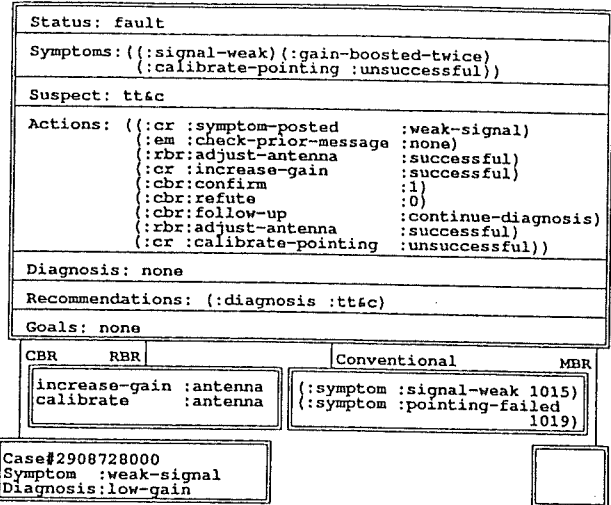


Figure 2: Sample Operation (cont'd). Through follow-up and cooperation additional symptoms have been identified. The CR module has recommended that the TT&C subsystem be diagnosed.

Tracking, Telemetry, and Control (TT&C) subsystem to be suspected. The state of the blackboard at this point is shown in Figure 2.

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 3. The results are stored by the CBR module, the blackboard is scrubbed, and control is returned to the CR module.

Status: fault	
Symptoms: ((:weak-signal) (:gain-boosted-twice) (:calibrate-pointing-unsuccessful) (:ambient-temperature-high :rwa))	
Suspect: actuator-motor-A	
Actions: ((:cr :symptom-posted :weak-signal) (:em :check-prior-message :none) (:zbr :adjust-antenna :successful) (:cr :increase-gain :successful) (:cbr :confirm :1) (:cbr :refute :0) (:cbr :follow-up :continue-diagnosis) (:zbr :adjust-antenna :successful) (:cr :calibrate-pointing :verified-sound) (:mbr :diagnose-ttfc :diagnose-acs) (:cbr :recommend :rwa-faulty) (:mbr :diagnose-acs :closed-louver) (:zbr :diagnose-rwa :open-louver) (:cr :refute :actuator-faulty) (:mbr :diagnose-thermal :0) (:cbr :confirm :0) (:cbr :refute :0) (:cbr :follow-up :none) (:cr :switch-to :actuator-B))	
Diagnosis: actuator-A	
Recommendations: none	
Goals: none	
CBR	RBR
increase-gain :antenna	calibrate :antenna
open-louver :rwa-bay	
Conventional	
(:symptom :signal-weak 1015)	
(:symptom :pointing-failed 1019)	
MBR	
Case#2908728000	antenna :good
Symptom:weak-signal	amplifier :good
Diagnosis:low-gain	phase shifter :good
Case#2902248000	power-supply :good
Symptom:calibrate-failed	ground-output :good
Diagnosis:faulty-acs	ttfc :good
	ace :good
	earth-sensor :good
	sun-sensor :good
	rate-gyro :bad
	rwa :bad
	louver :bad
	actuator-A :bad
	actuator-B :off
	lce :good
	temp-sensors :good

Figure 3: 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.

6.2 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 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.

7 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 contrasting knowledge. 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 shown in Table 1, 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.

While we have presented SRS as an approach to diagnostics, it represents 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 also provides a basis for future research in integrating reasoning paradigms.

References

- Blackboard Technology Group. *GBB User's Guide Version 2.0*. Amherst, MA: BBT Group, February 1991.
- Campbell, William S., Satellite Analyst. Personal interviews. Kirtland AFB, NM, March 1992.
- Cohn, A. G. "On the Appearance of Sortal Literals: a Non Substitutional Framework for Hybrid Reasoning," *Proceedings of the First International Conference on Knowledge Representation*, edited by R. Brachman *et al*, 55-66. San Mateo, CA: Morgan Kaufmann (1989).
- Davis, Randall. "Diagnosis Via Causal Reasoning: Paths of Interaction and The Locality Principle," *Artificial Intelligence in Maintenance*, edited by J. Jeffrey Richardson, 102-122. Park Ridge, NJ: Noyes Publications, 1985.
- Engelmore, R.S. and A.J. Morgan. *Blackboard Systems*. Wokingham, England: Addison-Wesley Publishing Company, Inc., 1988.
- Frisch, Alan M. "A General Framework for Sorted Deduction: Fundamental Results on Hybrid Reasoning," *Proceedings of the First International Conference on Knowledge Representation*, edited by R. Brachman *et al*, 126-136. San Mateo, CA: Morgan Kaufmann (1989).
- Garnham, John, Satellite Operator. Personal interviews. Kirtland AFB NM, October 1990.
- Gruber, Thomas and Yumi Iwasaki. "How Things Work: Knowledge-based Modeling of Physical Devices," KSL 90-51. Stanford, CA: Stanford University, August 1990.
- Hayes-Roth, Fredrick and Victor R. Lesser. "Focus of Attention in the Hearsay-III System," *Proceedings of the International Joint Conference on Artificial Intelligence*: 27-35 (August 1977).
- Keller, Richard *et al*. "Model Compilation: An Approach to Automated Model Derivation," NASA report RIA-90-04-06-1, April 1990.
- Koton, Phyllis. "Reasoning about Evidence in Causal Explanations," *Proceedings of AAAI-88*: 256-261 (1988).
- Luger, George F. and William A. Stubblefield. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Redwood City, CA: Benjamin/Cummings, 1993.
- McSkimin, James R. and Jack Minker. "A Predicate Calculus Based Semantic Network for Deductive Searching," *Associative Networks: Representation and Use of Knowledge by Computers*, edited by Nicholas Findler, 205-238. New York: Academic Press, 1979.
- 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.
- 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).
- Skinner, James M. *A Synergistic Approach to Reasoning*, Ph.D. Dissertation, Department of Computer Science, University of New Mexico, Albuquerque, NM 87131, May, 1992.
- Wertz, James R. and Wiley J. Larson. *Space Mission Analysis and Design*. Dordrecht, Netherlands: Kluwer Academic Publishers, 1991.