N93-11930

Intelligent Fault Management for the Space Station Active Thermal Control System

Tim Hill
McDonnell Douglas Space Systems Company
Space Station Division
16055 Space Center Boulevard
Houston, TX 77062
TIMHILL@NASAMAIL.NASA.GOV

Robert M. Faltisco
McDonnell Douglas Space Systems Company
Space Station Division
16055 Space Center Boulevard
Houston, TX 77062
RFALTISCO@NASAMAIL.NASA.GOV

Abstract

This paper describes the Thermal Advanced Automation Project (TAAP) approach and architecture for automating the Space Station Freedom (SSF) Active Thermal Control System (ATCS). The baseline functionality and advanced automation techniques for Fault Detection, Isolation, and Recovery (FDIR) will be compared and contrasted. Advanced automation techniques such as rule-based systems and model-based reasoning should be utilized to efficiently control, monitor, and diagnose this extremely complex physical system. TAAP is developing advanced FDIR software for use on the SSF thermal control system. The goal of TAAP is to join Knowledge-Based Systems (KBS) technology, using a combination of rules and model-based reasoning, with conventional monitoring and control software in order to maximize autonomy of the ATCS. TAAP's predecessor was NASA's Thermal Expert System (TEXSYS) project which was the first large real-time expert system to use both extensive rules and model-based reasoning to control and perform FDIR on a large, complex physical system. TEXSYS showed that a method is needed for safely and inexpensively testing all possible faults of the ATCS, particularly those potentially damaging to the hardware, in order to develop a fully capable FDIR system. TAAP therefore includes the development of a high-fidelity simulation of the thermal control system. The simulation provides realistic, dynamic ATCS behavior and fault insertion capability for software testing without hardware related risks or expense. In addition, thermal engineers will gain greater confidence in the KBS FDIR software than was possible prior to this kind of simulation testing. The TAAP KBS will initially be a ground-based extension of the baseline ATCS monitoring and control software and could be migrated on-board as additional computational resources are made available.

Introduction

Thermal control systems are very complicated to operate in both nominal and off-nominal states. This problem is compounded in space. The Space Station Freedom (SSF) Thermal Test Bed (TTB) at the National Aeronautics and Space Administration's (NASA) Johnson Space Center (JSC) in Houston, Texas is one such complex system. The operation takes several thermal experts (typically thermal engineers) as well as a staff of technicians, safety officials and other support personnel. In short, the TTB is expensive to operate and technically complex, particularly in off-nominal states. For this reason and due to the lack of plausibility of having such a TTB support staff on the space station, the need for autonomous operation of the Space Station Freedom Project's (SSFP) Thermal Control System (TCS) is apparent. However complicated the task of autonomous monitoring and control of the SSFP TCS, it is necessary to have a high degree of automation due to the harsh environment of space and the time which astronauts would spend on such "housekeeping" functions (e.g. monitoring temperatures, opening valves, etc...) without such system autonomy.

It is this problem that NASA began addressing in the System's Autonomy Demonstration Program's (SADP) Thermal Expert System (TEXSYS) project. The TEXSYS project was developed in the middle to late 1980's and was demonstrated in August of 1989. The project's major accomplishment was that it was the first real-time expert system to autonomously operate and diagnose faults in a system as complex as the thermal testbed TCS.

TEXSYS consisted of several software/hardware modules which enabled the expert system to accomplish its task. Figure 1 shows the architecture of the TEXSYS project. The overall job of TEXSYS was accomplished via the use of 4 separate computers. The expert system was on a Symbolics 3650 (denoted TEXSYS SYMBOLICS) and was the "brains" of operating the TTB.

The TEXSYS software diagnosed 17 faults and operated the thermal testbed ATCS accomplishing system startup, shutdown and temperature set-point change. The expert system utilized model-based reasoning with qualitative modeling and hypothetical reasoning via an assumption-based truth maintenance system [Glass 90]. Rules were used in conjunction with the models to accomplish diagnosis of most of the faults successfully isolated. The expert system was relatively large in terms of memory usage (on the order of several megabytes of memory).

The TEXSYS prototype diagnosed faults and controlled the behavior of a GROUND-based system in the TTB. It is desirable to utilize this technology in a space-based system located on the Space Station itself. The constraints of resources (memory and power) on the Space Station will not allow for such a large software system to be implemented. In addition, a high cost was incurred due to the use of such a complex hardware system for debugging and testing. The notion of using a simulation of the ATCS was not fully explored in TEXSYS.

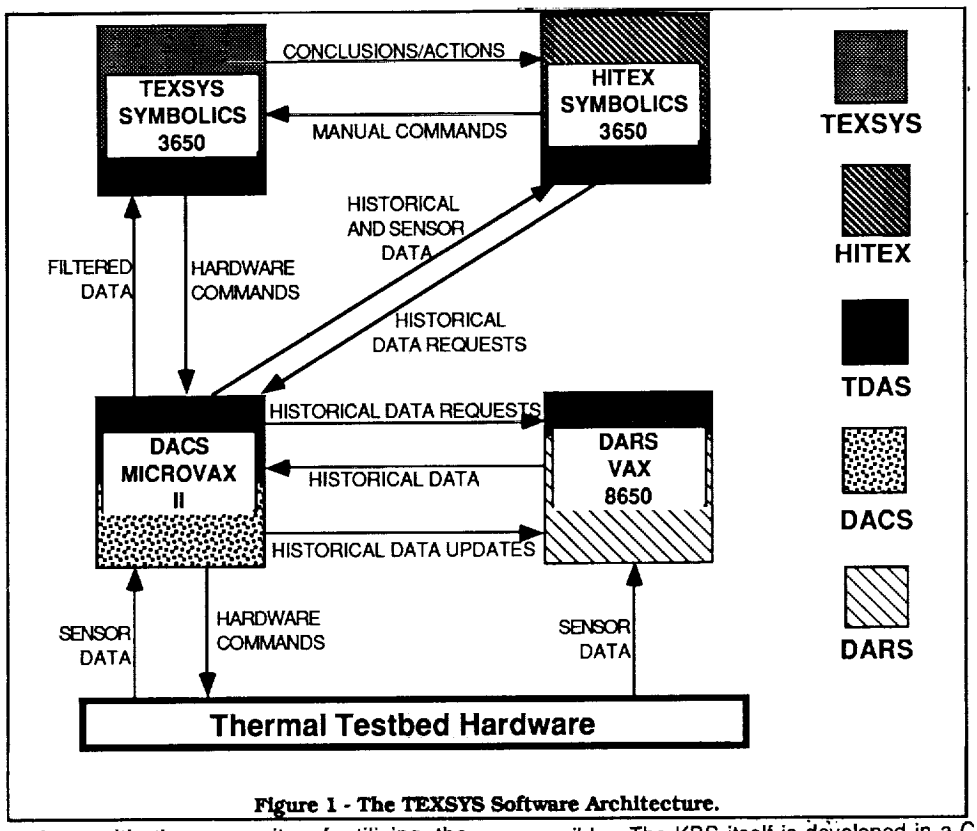

Figure 1 - The TEXSYS Software Architecture.

These reasons along with the necessity of utilizing the technology of TEXSYS for SSFP ATCS operation made follow-on work to TEXSYS important if not inevitable. NASA Headquarters' Advanced Development for SSFP initiated the Thermal Advanced Automation Project (TAAP) to address these problems.

The goals of TAAP include the development of a knowledge-based system (KBS) to cover all known faults on the Thermal Control System, a greater number of faults than TEXSYS covered. Also, to detect inconsistent behavior which cannot be attributed to a known fault. A high-fidelity simulation of the thermal control system will be developed and utilized for testing of the KBS. The architecture and the hardware and software platforms of the SSFP's Data Management System (DMS) should be adhered to whenever possible. In addition, since the contracting organization responsible for developing the SSFP Thermal Control System is doing the TAAP work, communication with the SSF team is greatly increased (i.e. McDonnell Douglas Space Systems Company - Space Station Division is building the TAAP systems).

The TAAP project addresses these issues. A high-fidelity simulation is under development with particular attention paid to the simulation of faults which are used for the testing of the Fault Detection, Isolation, and Recovery (FDIR) KBS. The simulation is being developed with a code called the Advanced Thermal Hydraulic Energy Network Analyzer (Athena) [ATHENA 86] which performs thermal-hydraulic calculations required to simulate operation of the Active Thermal Control System (ATCS). The plans are to run the simulation in a batch/file mode with real-time (or near real-time) capability added later.

TAAP uses a similar architecture and platform to the Space Station's Data Management System. That is, an Intel 80386 based computer with Ada-based code is utilized when possible. The KBS itself is developed in a C-based language called CLIPS (C Language Integrated Production System). CLIPS has been ported to provide to an Ada-based version. The simulation and KBS portions will be connected via a Space Station-like database system. This system is also developed in Ada.

More details on the TAAP approach are given in the ATCS Advanced Automation section. The purpose of this introduction was to provide an understanding of the origins of the concept of automating the Space Station Thermal Control System and an introduction to the TEXSYS and TAAP approaches.

Space Station Freedom Thermal Control System Overview

The Thermal Control System (TCS) for Space Station Freedom (SSF) will provide the heating and cooling control necessary to maintain elements, systems, and components within their required temperature ranges. The SSF TCS is comprised of systems for passive and active thermal control. Passive thermal control is performed through the use of coatings, insulation, isolators, and selective placement of heaters. Active thermal control includes both internal and external systems. The internal thermal control system collects and transports waste heat to the external system from the habitation modules, resource nodes, and airlocks. The external active thermal control system consists of a thermal bus, its control hardware and software, and a set of space radiators. The external system is the focus of the TAAP, and is referred to here as the ATCS.

The ATCS is a central facility, transporting waste heat away from crew quarters, experiment packages, computers, etc., and radiating it into space. It utilizes ammonia as the working fluid and interfaces via heat acquisition devices

(HADs) with the habitation and laboratory modules, and pallet mounted equipment where the heat dissipation rates are too high to be controlled passively. HADs are heat exchangers that remove heat from fluid systems and electronic equipment directly. Liquid ammonia is supplied to the HADs by the ATCS and is vaporized by the particular heat load being serviced. The vapor is transported to the radiators which reject heat to space. Figure 2 shows a configuration of the major ATCS components on SSF.

Key pumping and control elements are the Rotary Fluid Management Device (RFMD) and the Back Pressure Regulating Valve (BPRV). The RFMD is a special centrifugal pumping device designed to separate liquid from vapor and operate in zero-gravity. Temperature control of the system is accomplished by control of RFMD drum pressure. The BPRV regulates saturation conditions in the main chamber of the RFMD by regulating pressure over a range corresponding to the desirable temperature set point of the system. Both of these components exhibit highly nonlinear nominal behavior which makes constructing dynamic numerical simulations difficult.

The FDIR Automation Problem

Performing Fault Detection, Isolation, and Recovery means monitoring and identifying fault conditions through interpretation of sensor inputs and calculated data. System control is required for executing the approriate actions to recover from the condition and return to a nominal operating mode, if possible. In an automated ATCS system, monitoring and control (M&C) and FDIR functions will be integrated. The system will be able to detect and diagnose the causes of faults, and then request confirmation or automatically invoke appropriate recovery or reconfiguration procedures.

The need for automation is clear. Complex space-based systems require constant monitoring of health information. Human operators scan telemetry for slight deviations from expected norms. In real-time, continuous operations this is very expensive. For shuttle operations, many systems require several engineers to monitor parameters, configuration, and component health changes for anomalies. These activities continue 24 hours a day during a mission. Shuttle missions last only a few days, while the SSF is projected to be in orbit for 30 years. By automating some or most of the FDIR functions the need for constant direct human monitoring and intervention will be decreased.

It is difficult to overstate the complexity of monitoring and diagnosing continuous variable dynamic systems in which few system parameters are observable. A variation of the General Diagnostic Engine (GDE), a model-based reasoning algorithm, was applied to a prototype space station thermal bus on the TEXSYS project. It was found that maintaining multiple fault hypotheses in currently available truth maintenance systems made real-time (approx. 1 minute) performance very difficult to achieve. Fault detection capability is also dependent on sensor distribution and variety. Less instrumentation increases the complexity of obtaining data for FDIR and increases reliance on algorithms to infer the state of a component from less direct sensor information.

Baseline Approach

This section describes the current issues facing the automation of the SSFP ATCS. The SSF team has recently undergone a Congressionally mandated restructuring effort. This effort has changed the particulars of the SSFP ATCS design including the design of the monitoring and control software which is responsible for the FDIR. However, the general choice of software techniques seems to remain the same, so this technique is described. Additionally, the constraints which led to this design concept are described and the relative advantages and disadvantages of such an approach are given.

This redesign affects software in a very large way. The number of onboard processors is now reduced to two Intel 80386 machines for initial phases of the SSFP. The bottom line for the monitor and control (M&C) software of the TCS is that there will be approximately 80kBytes of memory for the entire system. This poses a major constraint. It was decided that the M&C software would be divided into two parts, an onboard portion and a ground-based portion. The

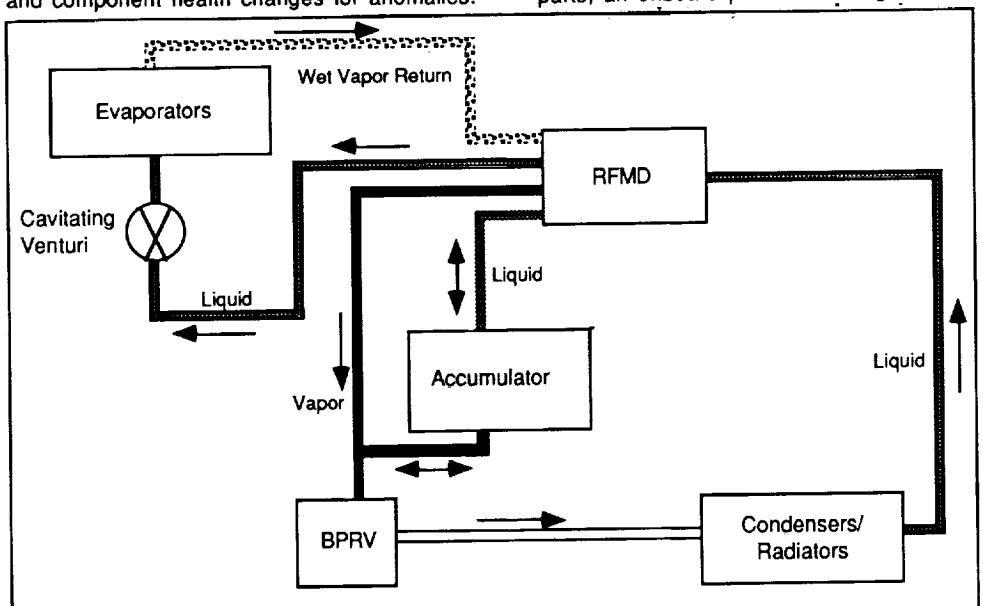

Figure 2 - SSF Thermal Component Overview.

onboard portion would have the task of handling the safety-critical faults, faults which must be isolated and/or recovered from immediately due to the severity of the fault's consequences. The portion moved to the ground would handle all other faults which are not as constrained by criticality or response time.

The baseline approach, prior to the restructuring, has been a table-driven design and it is likely this same technique will continue as the initial design for the M&C software. Basically, an event (e.g. sensor out of limits), triggered by the DMS will trigger the FDIR. The FDIR software will execute, using table data applicable to the event. This technique places the fault signatures across the rows of the table. An executive module matches the sensor data to one of these rows which contain sensor data approximations of what the fault corresponding to that row looks like (the fault signature). Each column will correspond to a particular sensor or other data item. When a match is made, that row's fault is isolated and the pertinent recovery action is taken.

This sort of table driven or table lookup approach fits well with the constraints when limited to safety critical faults only. It takes up small amounts of memory and storage, is relatively fast, and can isolate all *enumerated* faults. There are some problems with this approach that need to be examined.

One problem is that this can only diagnose faults written into the system. Multiple faults are not typically enumerated and if a sensor fails, the fault signatures change for all faults using that particular sensor, or, if the sensor reads bad information, it could diagnose a fault that isn't there. The table-lookup mechanism cannot, by itself, handle a problem not explicitly coded in the table.

It's important to have a mechanism for sensor failures to be contained. Sensor noise could force a false positive of a particular fault signature. Without a sensor check mechanism, a false recovery action could be taken. If a sensor fault is allowed to persist, which is quite likely due to the lack of reliability of sensors, particularly thermocouples, the ability to isolate even the enumerated faults is greatly diminished, if not eliminated.

Another major concern is the problem of FDIR on systems which have been reconfigured. Failed sensors and components could take months to replace. The tables would need to be coded for every newly reconfigured system state. This can lead to a combinatorial explosion of tables. Each fault multiplied by each sensor, multiplied by each new system state is quite a large number of tables. Considering the difficulty of enumerating all the faults, or all the system reconfigured states, this is quite a large problem.

However problematic the baseline approach is for handling FDIR on the SSF ATCS, it is well thought out considering the computational restraints onboard. With regards to onboard design and emergency fault handling capability, there may initially be no other choice. The major issue is that a predominant amount of faults will be handled in ground-based systems. This ground based portion of the system must avoid most of the problems with table lookup enumerated above. The automation of FDIR requires the use of the most advanced programming techniques available.

ATCS Advanced Automation

Advanced automation includes techniques and applications for monitoring, control, and FDIR. Some of the criteria used for determining the degree of automation are:

- safety,
- lowered operations life cycle costs,
- development cost,
- efficiency (ground controller and crew productivity),
- reliability,
- technology maturity,
- integration with existing controls,
- operational support requirements,
- human factors concerns (response time, task monotony),
- impact to established schedules and baselines,
- chances of operational acceptance.

Another factor effecting the thermal system automation scope was the removal of all but safety critical FDIR from on-board.

The TAAP has two primary activities: development of a monitoring and diagnosis system for the ATCS and development of a faultable high-fidelity simulation (HFS) of the ATCS. While the HFS is not the main subject of this paper, it is crucial to the overall success of the KBS development. The high-fidelity simulator will have interactive capabilities and provide dynamic bus performance to simulate known ATCS fault modes. Development and testing of FDIR software requires faulting the hardware in various ways while observing and tuning the detection and isolation process. This type of hardware testing is extremely expensive. Faults which might damage the ATCS must obviously be avoided, yet a method is needed for testing all conceivable faults. The HFS will allow inexpensive testing of virtually any fault without risk to hardware.

The TAAP KBS is a hybrid design approach using a combination of conventional programming, tables, rule-based technology and model-based reasoning to provide powerful and flexible diagnositic expertise for the ATCS. TAAP will initially act as an extension to the SSF baseline FDIR capabilities, and could be migrated on-board in stages as additional computational resources are made available. The top level fault isolation process used by the TAAP KBS is:

1) The KBS determines that a sensor has passed one of its alarm limits;
2) A table-based check is performed to quickly catch a small number of potentially catastrophic failures;
3) The KBS uses local propagation and consistency checking in a model to determine whether the sensor reading is valid or the result of faulty instrumentation;
4) If the alarm is valid, the KBS uses a rule-based system to isolate faults defined in the Failure Modes and Effects Analysis (FMEA) [FDIR 88];
5) If the rule system is unable to isolate the fault conclusively, then control is returned to a model-based reasoning (MBR) algorithm for further diagnosis.

Part of the system is implemented using CLIPS v5, which includes an object system called COOL (CLIPS Object Oriented Language). CLIPS is primarily a rule-based forward-chaining inferencing tool. The TAAP rule-based system consists of forward-chaining rules driven by ATCS sensor data and calculated values. The COOL environment is different in some ways from a pure object oriented programming (OOP) language, however it does have the primary characteristics that an OOP system must possess: abstraction, encapsulation, inheritance, polymorphism and dynamic binding. This allows us to implement a "functional simulation" of the ATCS by defining objects and relationships in the COOL environment.

Procedural and numeric-intensive portions of the KBS are currently implemented in structured C code. This includes interface functions needed to handle input and output of parameters to external functions, as well as some calculations involved in model propagation.

Figure 3 shows a functional view of the TAAP KBS architecture. Real-time sensor
data from the HFS or actual hardware is accepted into the KBS by Data Queue code which stores time-slices of data and performs several checks. It checks flags for each sensor to see which values have been updated, then it creates a Suspect Sensor List (SSL). The SSL contains information about which sensor values have exceeded KBS alarm limits. The Data Monitor module is primarily responsible for standard calculations performed at the end of each time-slice. These include: recalculating/updating pseudo values, updating short- and long-term trends, checking/updating historical deltas, maintaining current-value-duration information, and determining qualitative mappings. The Data Queue and Data Monitor together contain a representation of the current system configuration at any time (i.e. valve states, switches, power, etc.).

The KBS Controller is the primary switching center for the KBS. First, it inspects the SSL to determine if sensor validation is required. If necessary, it then supplies the model with sensor readings corresponding to model slots and invokes sensor validation. It also builds facts and asserts them into the Rule-Based System (RBS) facts list. When the RBS or model-based sensor validation is complete, the KBS Controller interprets their diagnoses. If the instrumentation is deemed valid and the RBS cannot match against a known fault, then the Controller initiates model-based reasoning again, this time for the purpose of component fault diagnosis.

Sensor validation is performed when there is exactly one suspect sensor. When more than one sensor is referenced by the SSL, sensor validation is not performed. The assumption is that if only one sensor is out of its application limits, then it is more reasonable to suspect that the sensor has failed than to suspect that an ATCS component failure is causing a single perturbation. After the model has been instantiated with current sensor values, pre-compiled hierarchical information (generated directly from the model) is used to determine the scope of propagation required. The model is then propagated to bind state variables not connected to sensor instrumentation and to compute an expected value for the suspect sensor. The observed readings at each point are then compared with the corresponding computed values in the model. If these two values are within tolerance for all non-suspect sensors, then the suspect sensor is diagnosed as invalid. If one or more non-suspect sensors are out of tolerance (reading vs. computed value), then no determination can be made regarding the validity of the suspect sensor. More specifically, we now have reason to believe that our initial assumption of nominal operation was incorrect.

Note that the model is propagated with the assumption that the bus is operating nominally. In fact both nominal and off-nominal operating conditions cover continuous ranges. For example, if one evaporator has been shut down, the reconfigured bus can still be viewed as operating nominally. Conversely, there is no discrete distinction between off-nominal behaviors. An infinite number of scenarios are possible.

The RBS uses forward chaining rules, matching on patterns of system status information. The design intention of the RBS was to represent the thermal expertise at the highest possible level in the rules. Underlying functionality required for this purpose is generally implemented in the faster procedural language - C. The following is an example of the left-hand side of a rule showing the use of high level thermal knowledge.

```
(defrule ...
    (st-trend TOTAL_HEAT_LOAD steady)
    (qpseudo SUBCOOLING very_low)
    (qsensor BPS702 nominal)
    (test
      (> (extrap BPS702 20.0 STT) 300))
    ... )
```

Referencing qualitative states (e.g. steady, very_low, nominal) rather than numbers in rule premises makes the

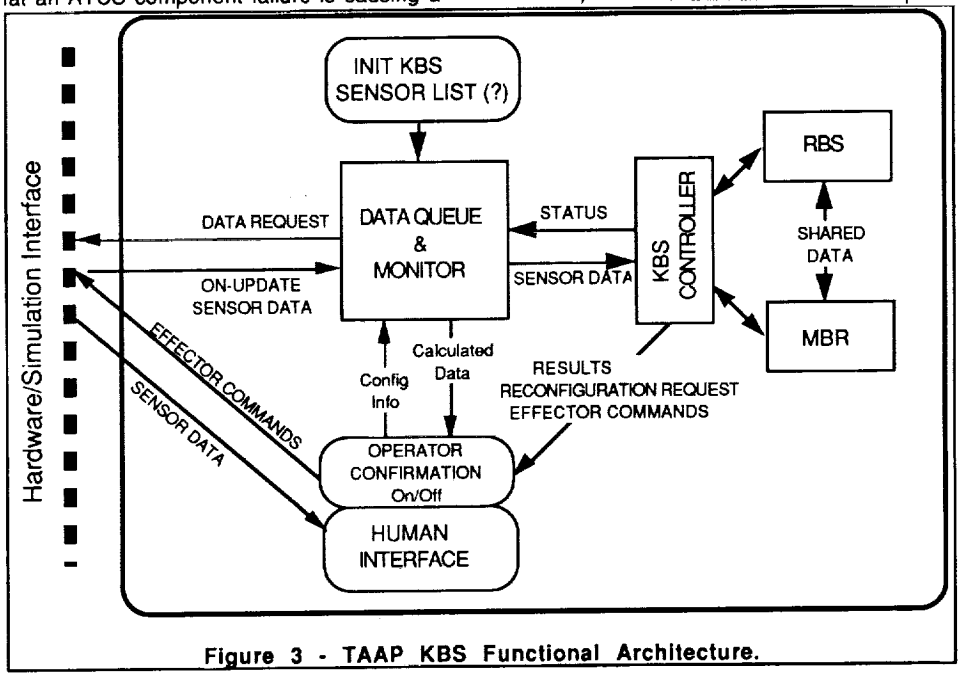

Figure 3 - TAAP KBS Functional Architecture.

RBS much more robust. Transition points are defined and can be modified such that a quantitative value range corresponding to a qualitative translation of "low" may be different depending on the current system mode. This allows the rules to represent thermal knowledge the way it is expressed by the thermal engineer. A straight-forward statement such as "determine if the heat load is steady" represents a significant amount of knowledge, non-trivial calculation and data tracking. The fundamentally numeric-intensive portions are handled procedurally while only the diagnostic knowledge is placed into the rules. Another benefit of qualitative techniques is that pre-compiled relationships (extracted directly from the model) can be used to replace primary sensor values with secondary choices. On-orbit a failed sensor may not be replaceable for several months. Working with the instrumentation available is imperative.

Model-based reasoning for component diagnosis is the most complex and "cutting-edge" portion of the KBS. While the model-based sensor validation phase could assume nominal operation, model-based component diagnosis is much more difficult. The MBR component fault diagnosis makes heavy use of thermal specific knowledge for hypothesis generation and validation.

In performing component diagnosis the entire model is propagated. This will prevent missing systemic faults, where failures in one part of the bus quickly impact a structurally distant portion. The assumption is made that the bus is operating normally, and that all of our sensor instrumentation is valid. Propagating the model and comparing sensor readings to computed values allows us to build a fault-symptom list. The fault-symptom list represents inconsistencies between observed and computed values.

The fault-symptom list is then "mapped" onto a causal relationship model of the ATCS. The causal description of the system is used to generate hypotheses of component failures which could be causing each of the symptoms. For each inconsistency represented in the fault-symptom list the causal model will determine single-component failures which could explain that specific inconsistency. The intersection of the component-failure hypotheses sets returns only those which could be responsible for all the symptoms. A future capability to diagnose multiple simultaneous independent failures would be added by expanding the scope/knowledge represented in the causal model. Due to reliability requirements, diagnosing simultaneous independent failures is not considered a primary goal of this project.

The hypotheses generation phase will result in hypotheses which could account for all the observed symptoms. The validation phase will determine which of these can also account for the remaining observations. Each failure hypothesis is simulated in the faultable, quantitative model. For example, if we hypothesize that a blockage just downstream of the BPRV could cause all the inconsistencies, then we reconfigure the model to reflect a valve at that location being closed and run a simulation. If one of the simulations results in state values comparable to current observations from the sensors, then that hypothesis has been validated (i.e., any surviving hypothesis represents a fault). If none of the hypotheses are supported by simulation, then further discrimination between them may not be possible. Prompting the human operator for more information might allow us to discriminate further between hypotheses. However, automatically determining the appropriate questions depending on the situtation is a difficult task not currently within the scope of this project.

The multi-stage FDIR process just described makes significant additions to the baseline approach. The design attempts to use the best parts of several methodologies while avoiding their restrictions.

The functional use of a Data Queue and Monitor for representation of the current system configuration has advantages over both the baseline and traditional model-based approaches. It provides the structure and data access necessary for complex procedural control processes to be kept updated while avoiding the overhead of conventional OOP systems. Since data control, monitoring and distribution are core requirements of every KBS function, the optimization of these activities is crucial to obtaining desireable performance.

In an early stage, the TAAP KBS uses a minimal table lookup for detecting safety-critical faults, similar to the baseline approach. The TAAP approach allows even the safety critical routine to utilize secondary sensor-source information generated from the model. Just as in the RBS, this makes the table lookup more robust.

The use of a KBS Controller as a centralized switching mechanism for reasoning accomodates a hybrid, multiple technique approach, and effectively modularizes the entire reasoning portion of the system. The relative maturity of rule-based reasoning means the RBS would be more likely to migrate on-board earlier than the MBR capabilities. The decoupling of diagnostic approaches has an advantage of allowing initial ground-based support versions to easily run on seperate, networked workstations if necessary. This improves the prospects of achieving satisfactory response times.

By assuming that multiple simultaneous sensors failures are extremely unlikely, the need for an ongoing sensor validation routine is avoided. When sensor validation is needed, a single propagation and straight-forward consistency check are sufficient.

As mentioned above, some variation of the RBS will likely be the first ATCS advanced automation to migrate on-board. Advantages of the rule-based system over the baseline table-driven approach are numerous. The rules represent diagnostic knowledge at a high level, allowing easier human interpretation, maintenance and meaningful explanation capabilities. The RBS is not tied to a specific configuration. Its data-driven nature combined with support code for primary and alternate instrumentation allows it to degrade more gracefully than a table lookup.

Many previous automation efforts have been heavily driven by schedule, resulting in technical design decisions that range from carefully considered to poorly designed. The fact that "deep reasoning" models can be built from design activities rather early in a program allows us to have begun implementation of this KBS long before first element launch, or even completion of the thermal testbed work. The TAAP model-based reasoning for component diagnosis will require several test and modification cycles before the system is fully functional. Before first element launch the thermal system hardware will go through at least three more major milestones representing potential design changes. All of this serves to further point out the advantages of a model-based approach.

The device oriented representation used for MBR is the cognitive link between software and hardware. By keeping development responsibility for the KBS within the thermal systems group, device models for reasoning can be updated in a timely and accurate manner as new design information becomes available. It is clearly advantageous to have the KBS developers located with the ATCS system engineers.

Conclusion

The table lookup approach described fits well with current onboard constraints of memory and processing power. Although the tables will be difficult to maintain and expand, the inclusion of only safety-critical faults may keep them small enough to avoid any significant problems. Failure of onboard sensor instrumentation can cause problems for the table driven FDIR. Temporary changes in system configuration, such as isolating and shutting down a single evaporator, may require table changes. Finally, the continuous nature of the ATCS when operating nominally may expose the biggest drawback of the table lookup approach.

A hybrid KBS approach to advanced automation will provide both lower lifecycle costs and increased capability. The high-level knowledge representation used in the KBS allows changes to be made easily and new maintenance and support personnel to come up to speed quickly. The KBS will be more robust in reasoning over a degraded or reconfigured system than a table driven approach. The diagnosis of off-nominal behavior, not just known faults, will be possible with the hybrid approach described. The KBS will initially act as a ground-based extension to the core capabilities provided by the onboard table lookup.

The combination of stable, mature technologies with state-of-the-art MBR approaches is advantageous for several reasons. It allows portions of the KBS to be ported on-board as they are completed, confidence is developed, and computer resources become available. It is advantageous to the FDIR development effort to use existing resources and technology. The forward chaining rules used in the RBS, for example, are considered very mature technology. Each of the approaches require successively more speed and RAM for actual implementation.

Given significant mistrust of artificial intelligence techniques by the thermal engineers, user confidence in a system such as the TAAP KBS must be earned gradually. A system offering only a small improvement over baseline monitoring and control capabilities would have difficulty gaining acceptance. The TAAP approach will allow an evolution of functionality and in so doine, will enable engineers to become more confident is this technology. New technologies are being developed all the time and the TAAP approach will allow integration of such technologies as they become available (e.g. predictive maintenance, faster MBR systems, etc.).

The cost of automation endeavors has often been excessive and in some cases prohibitive. Viewed more closely, however, a KBS approach to automation is cost effective. The development of the TAAP KBS for FDIR also supports a console operator in real-time and has components in its implementation (both knowledge bases and inferencing techniques) useful in predictive maintenance systems, training, and autonomous operations. If the common underlying system knowledge can be used in deploying each application then additional knowledge acquisition costs are saved. If these factors are used to effectively prorate project costs, the apparent high price of automation is reduced.

The cost of advanced automation can also be compared to the cost of extensive human training for monitoring of space systems. The full integrated mission simulations used for training shuttle support personnel cost thousands of dollars per day. These training costs alone demand that any task which can be effectively automated, should be automated.

In a major multi-year, multi-national development effort such as the SSFP, many design changes for various reasons will be made, impacting all aspects of the project. These design changes also have serious impact on development of FDIR capabilities for effected systems. A dynamically changing hardware environment favors the use of a hybrid approach with a device oriented model-based representation over the baseline techniques described.

Acknowledgements

TAAP KBS design and development reflects the efforts of numerous people at MDSSC and NASA. Substantial contributions were made by: Roy Steel, William Morris, Charlie Robertson, Jerry Snider, and Sunil Fotedar. MDSSC elements involved are: Thermal Systems Group, Advanced Automation Technology Group (AATG), and Systems Engineering and Integration (SE&I). NASA involvement includes: JSC Crew and Thermal Systems Division, JSC Automation and Robotics Division, and NASA Ames Research Center Advanced Missions Technology Branch.

References

[ATHENA 86]
Idaho National Engineering Laboratory, "ATHENA Code Manual: Code Structure, System Models, and Solution Methods," EGG-RTH-7397, EG&G-Idaho, Inc, September 1986.

[Glass 90]
Glass, B. J., et. al., "TEXSYS: a large scale demonstration of model-based real-time control of a space station subsystem," First International Workshop on Principles of Diagnosis, Stanford University, July 23-25, 1990.

[FDIR 88]
Crew and Thermal Systems Division, "BAC Two-Phase Thermal Control System: FDIR Document," NASA - Johnson Space Center, 1988.