AN APPLICATION OF KNOWLEDGE-BASED SYSTEMS TO SATELLITE CONTROL

Barbara Skiffington
Jim Carrig
Jim Kornell
General Research Corporation
7655 Old Springhouse Road
McLean, Virginia 22102

ABSTRACT

This paper describes an expert system prototype which approaches some issues of satellite command and control. The task of the prototype system is to assist a spacecraft controller in maneuvering a geosynchronous satellite for the purpose of maintaining an accurate spacecraft pointing angle, i.e. station keeping. From an expert system's point of view, two features of the system are notable. First, a tool for automated knowledge acquisition was employed. Because the domain experts were in Maryland while the AI experts were in California, a means to automate knowledge acquisition was required. Second, the system involves a blend of simulation and expert systems technology distributed between a DEC VAX computer and a LISP machine (a special purpose AI computer). This kind of distribution is a plausible model for potential real-world installations.

INTRODUCTION

On orbit satellite state-of-health monitoring/assessment and maintenance functions constitute a large portion of the effort involved with operating satellites. Numerous tasks are performed which depend heavily on ground support. The increased number and complexity of future satellites will significantly increase the work load and the dependency on these ground stations. This dependency is breeding growing emphasis on autonomous spacecraft and mobile units. [2] A reasonable charter is to build an automated controller's assistant to lessen the burden of the ground site controllers. In addition, it improves our understanding of how an automated assistant can be implemented either in a mobile unit or (in the future) on-board the satellite.

Scheduling routine tasks, command recommendation and monitoring should be some of the tasks required of an automated satellite controller's assistant. It is also important to provide assistance in preparing and validating the commands as well as making sure nothing is overlooked. In the event of a malfunction, the system should be able to quickly detect the problem and provide recommended procedures for quickly reconfiguring or stabilizing the spacecraft until further diagnosis can be performed. This type of system will be vital in a mobile unit where the availability of experienced satellite controllers will not be assured.

This paper describes a knowledge-based prototype system under development by General Research Corporation and Space Communications Company. This prototype is being designed to assist a controller in performing a satellite station keeping maneuver. Station keeping requires maintaining accurate spacecraft longitude and orbit inclination. Routinely, velocity change maneuvers are performed to make corrections in the spacecraft orbit. The station keeping maneuver is currently commanded by the ground station controllers. The controllers follow a predefined set of procedures; however, there are decisions that they have to make as they are performing the maneuver. These decisions are based on the current state of the satellite as determined by incoming telemetry values, telemetry history and trends, and recently performed commanding. The controllers are under timing constraints and may be required to perform multiple tasks concurrently. Also, in the event of a problem, the spacecraft controllers must decide whether to continue or to cancel the maneuver based on the characteristics and severity of the problem.

Due to the nature of the station keeping task, i.e, evaluation of telemetry data and the specialized expertise, the prototype is being developed utilizing 1) traditional methods for manipulating and monitoring the telemetry data stream, 2) knowledge-based (expert system) methodology for incorporating the knowledge and experience of the controllers for decision making, and 3) a graphics display for easy interpretation of the satellite's state. This paper describes the architecture of the system and each functional part currently being developed.

SYSTEM ARCHITECTURE

Figure 1 depicts the components of the prototype system. The system is distributed between a DEC VAX and a XEROX 1108 Dandetiger, a LISP-based computer. They communicate over an ethernet network. Currently, the numerically intensive programs and the TIMM-based expert system reside on the VAX. TIMM™, the expert system generator, is written in Fortran and its inference engine is callable as a subroutine. This makes it easy to interface the expert system with traditional engineering software.

The prototype system is divided into four multi-tasked components: 1) the simulator, 2) the commander, 3) the front-end telemetry monitor and expert system, and 4) the graphics display. These components transmit data and messages via a VAX supported interprocess communication utility. Information related to the state of the spacecraft is placed in a shared common area or “blackboard”. The expert system uses the blackboard to learn the current state of the spacecraft and to keep track of what has occurred during the maneuver.

The station keeping maneuver can be divided into three stages: 1) spacecraft configuration, 2) switching spacecraft operation modes and performing the maneuver, and 3) reconfiguration for normal operation. These stages require evaluation of different telemetry parameters. The rate at which the parameters are evaluated is dependent on the effective resolution of the parameters of interest for each stage. The first and third stage require lower resolution parameters, while the second stage requires high resolution parameters. The simulator provides telemetry to the front-end processor and graphics display program at rates corresponding to actual telemetered rates.

For configuration and reconfiguration, the front-end processor accepts data from the simulator and processes it. Processing includes extracting the telemetry values of interest, computing derived parameters and checking for abnormal conditions. This information is passed to the expert system who makes recommendations based on predefined rules. The recommendations are routed to the display screen.

The second stage, the actual maneuver, involves monitoring specific parameters and if they deviate from expected limits, assessing the situation and determining if action is required. the task of the front-end telemetry program to monitor these parameters and if an abnormal situation arises, the expert system will be invoked. The expert system will determine if the maneuver should continue or be canceled. If it is canceled, the appropriate fail-safe procedure will be proposed.

Throughout the entire maneuver the graphics display program accepts telemetry data from the simulator and updates the screen appropriately. The commands recommended by the expert system are displayed. For development purposes, the commands are automatically relayed to the commander program, although there is the capability for the user to override the commands recommended by the expert system. Each component of the system is now described in more detail.

SATELLITE COMMANDER

The satellite commander provides the link between the user and the satellite simulator. It waits for commands sent via the display screen and relays them to the simulator. The commander can operate in two modes: 1) manually where the user dictates whether or not a command is sent and, 2) automatically where commands recommended by the expert system are automatically forwarded to the simulator. When a recommended command is sent to the simulator or is cancelled by the user, the commander program relays this information to the blackboard so that the expert system can follow up on its recommendations.

SATELLITE SIMULATOR

The satellite simulator for this prototype provides telemetry data that is characteristic of the Reaction Control Subsystem (RCS) and parts of the Attitude Control Subsystem (ACS). Both electrical and thermal characteristics are modeled. It also accepts and executes commands. The main function of the simulator is to provide a close facsimile of actual satellite behavior for evaluation of expert system performance. Testing is facilitated by using the simulator to create unusual situations and observe how the expert system handles them.

FRONT-END PROCESSOR

The front-end telemetry processor is the telemetry data manipulator. The function of this front-end processor is multi-purpose. For monitoring, it is instrumental in detecting abnormalities, indicating when to signal an alarm both to the user and the expert system. This detection can be done by modeling or knowing what normal conditions are and noticing deviations from normal. Also, the front-end processor computes derived and/or time history parameters used by the expert system to determine the timeliness and appropriateness of commanding. It checks the telemetry stream for indications that a command has been executed, i.e., command verification. The front-end processor essentially reduces the telemetry stream into information that is useful to the expert system. This information is sent to the expert system for evaluation.

EXPERT SYSTEM

In this section, we describe the expert system portion of the prototype and TIMM™, The Intelligent Machine Model, being utilized to generate the expert system.

The expert system includes five knowledge bases shown in Figure 1. The first four, RCS, thruster thermal, spacecraft flight computer RAM management, and gyro, are associated with subtasks of the maneuver. For instance, one task performed in the spacecraft configuration stage is optimizing the thruster temperatures. This increases their efficiency and more critically, their overall life span. The expert system determines the optimal time and sequence to switch the heaters on or off. Its decisions are based on particularities of the maneuver, timing and equipment constraints as well as the temperature changes during this stage. The expert system uses the thruster thermal knowledge base for the rules applying to this task. The fifth knowledge base, fail-safe, contains rules for cancelling or continuing the maneuver. In the case of cancelling, it directs the user to the appropriate fail-safe procedure.

To develop and implement these knowledge bases, we are utilizing TIMM™, a domain independent expert system generator. TIMM™ reasons by analogical or pattern-directed inference, comparing the current situation to similar experiences (rules) in its knowledge base. When an exact match occurs between the antecedent clauses of a rule and the situation, TIMM™s decision is the related consequent clause of the rule. TIMM™ incorporates the capability for handling cases where there is no exact rule match. TIMM™ computes a numerical distance between the rule and the current situation. A distance equal to zero indicates an exact match; otherwise, the non-zero distance provides a measure of how well the current situation is matched by the rule. This is denoted as a reliability measure.

TIMM™ also provides an easy to use knowledge acquisition tool kit that allows the expert to deal directly with the computer to define, train, and evaluate the expert system. This promotes rapid acquisition and representation of knowledge during expert system development.

There are three stages in the TIMM™ knowledge acquisition process. First, the dimensions of the problem space are defined. By this, we mean that all of the following are enumerated:

- the scope of the problem;
- each of the possible decisions which can be made,
- each of the factors which must be considered in making a decision, and
- both the types of legal values for each factor (some combination of symbolic or numeric, ordered, unordered or circular), and the specific possible values or ranges of values for each factor.

For example, the name of the decision made by the expert for thruster thermal control is “recommended action for a particular heater”. The possible outcomes are “command on”, “command off”, or “do nothing”. The factors are items such as heater status information (on or off, total number of heaters on, etc.), temperature information (current temperature of associated thruster, the projected thruster temperature after the maneuver, etc.) and timing parameters (time heater has been on, number of minutes until maneuver start, etc.).

The second step in knowledge acquisition is the development or training of a knowledge base describing the behavior of the factors in determining the decisions. During this step, the expert generates training cases by specifying factor values and the decision associated with that combination of factor values. Training examples can also be created by TIMM™. Based on the problem space definition, TIMM™ generates, plausible situations in the domain, and queries the expert for the correct response to these situations. The expert can identify one of the decisions as being the unambiguous choice; alternately, she or he may wish to hedge by saying, for example, that the decision is about 70% likely to be one of the legal choices, 20% a second, a 10% a third. If it is appropriate, the expert can declare that the case being presented by TIMM™ is impossible, or so unlikely that it should not be considered by the system. TIMM™ has a special category for such negative or exclusionary rules. The expert is not forced to accept any and all combinations of legal factor values when some combinations may be unrealistic.

During the course of developing the knowledge base, the expert always has the option of adding, deleting or modifying factors according to an evolving understanding of the best way to represent the real world system being modeled. The expert can also view the current state of the knowledge base whenever she or he wishes.

The expert decides when the second step in the knowledge acquisition process is complete. As well as having TIMM™ generate training cases, the expert can direct the system to provide answers to the cases generated. By monitoring the quality of these answers, the expert can observe the increasing competence of the system. In this way, ongoing evaluation of the validity of the system can be made by the expert in the course of the training. Some further, and important, capabilities are available to the expert at this point for determining the fitness of the knowledge base: the tools for checking the consistency and completeness of the rule set. Completeness checking looks for underspecified areas of the problem space, and poses training cases specific to those areas. Consistency checking looks for multiple, conflicting rule coverage regarding situations legal in the problem space. If two or more rules (with differing decisions) apply to a single situation, such situations are called to the attention of the expert for remedial action.

The third and final stage of knowledge acquisition is provided by the option to invoke the generalization (or induction) tool. In this step, the user can direct TIMM™ to attempt to induce simpler, more general rules for system behavior from the foundation of the rules already present. There are two ways this can be done. In the first, the expert simply commands TIMM™ to generalize. In the second, the command is given, but each potential generalization is presented to the expert for approval prior to being incorporated in the knowledge base. In either case, no rules are lost in the process, so that if it is desired to return to the pre-generalization state, this can be accomplished. If the expert likes the new rules, TIMM™ can then be told to get rid of redundant rules. In experiments, this has resulted in up to a 70 percent decrease in the size of rule sets.

GRAPHICS DISPLAY

We used a graphic interface to show the subsystem of the satellite modeled by the simulation. Actual operators are accustomed to looking at reams of difficult to distinguish telemetry data, represented by numbers and plots. By providing a visual representation of the satellite subsystem, appropriately labeled, information can be visually associated with the component from which it comes. Thus, the status of any part of the subsystem can be seen at a glance. Further, plots of various parameters can be requested adjacent to the visual image. The graphics screen for the Reaction Control Subsystem is shown in Figure 2.

SUMMARY

The major components of the architecture presented here have been developed and demonstrated with the expert system performing the optimization of the thruster temperatures. We are incorporating and testing the remaining expert system components for the prototype system. The goal is to demonstrate that the system can assist and perhaps eventually handle a complex spacecraft control procedure such as the station keeping maneuver. In the event of an abnormal situation, we want the system to detect the condition and associate it with a predefined fail-safe procedure.

The automated acquisition tool, TIMM™, has been instrumental in prodding the expert to acknowledge situations that could occur but had not previously surfaced. The consistency checking and generalization procedures are also important, particularly when more than one expert is training the system.

Incorporating the TIMM™-based expert system with traditional software allows us to use the knowledge-based portion for decision making tasks and maintain the telemetry monitoring and data reduction tasks via conventional software. Also, incorporating the spacecraft simulator enables us to test the performance of the expert system by creating simulated unit failures and/or unusual situations and seeing how the expert system copes. This is an important aid for insuring credible performance of the expert system.

ACKNOWLEDGMENTS

The authors wish to acknowledge Owen Short, John Briggs, Chuck Gay, Jim Blue and Terry Rubin of Space Communications Company for development of the simulator and commander software, and contributing their satellite control expertise and support.

REFERENCES

1. Hayes-Roth, F., Waterman, D., Lenat, D., 1983, Building Expert Systems, Addison-Wesley, Reading, MA.

2. Marshall, M., 1981 , “Goals for Air Force Autonomous Spacecraft ,” USAF Report SD-TR-81-72, Jet Propulsion Laboratory.

3. Weiss, S., Kulikowski, C., 1984, A Practical Guide to Designing Expert Systems, Rowman & Allenheld: Totowa, NJ.

4. Kornell, J., 1984, “A VAX Tuning Expert Built Using Automated Knowledge Acquisition,” First IEEE Conference on Artificial Intelligence Applications, Denver, CO.

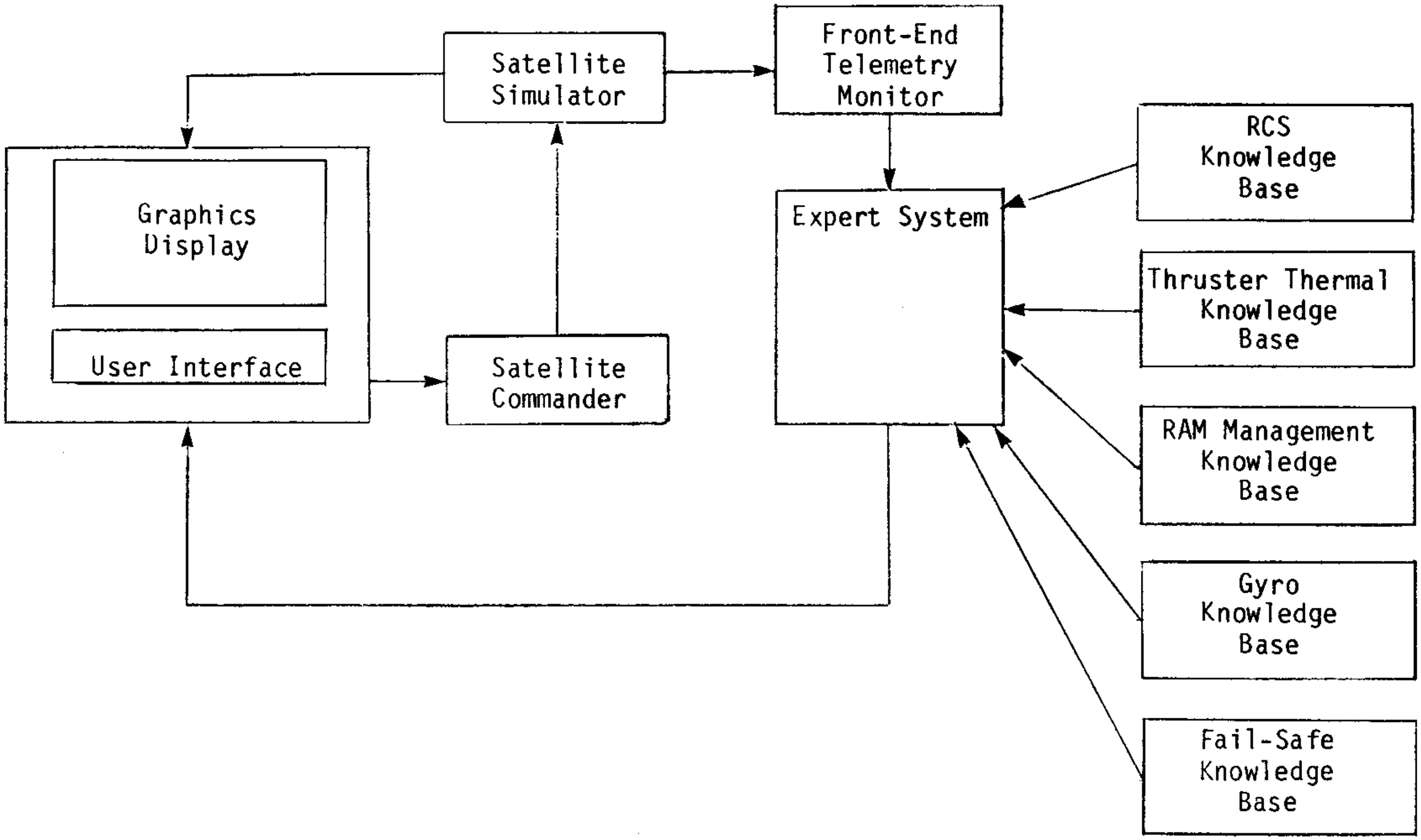

Figure 1. System Architecture

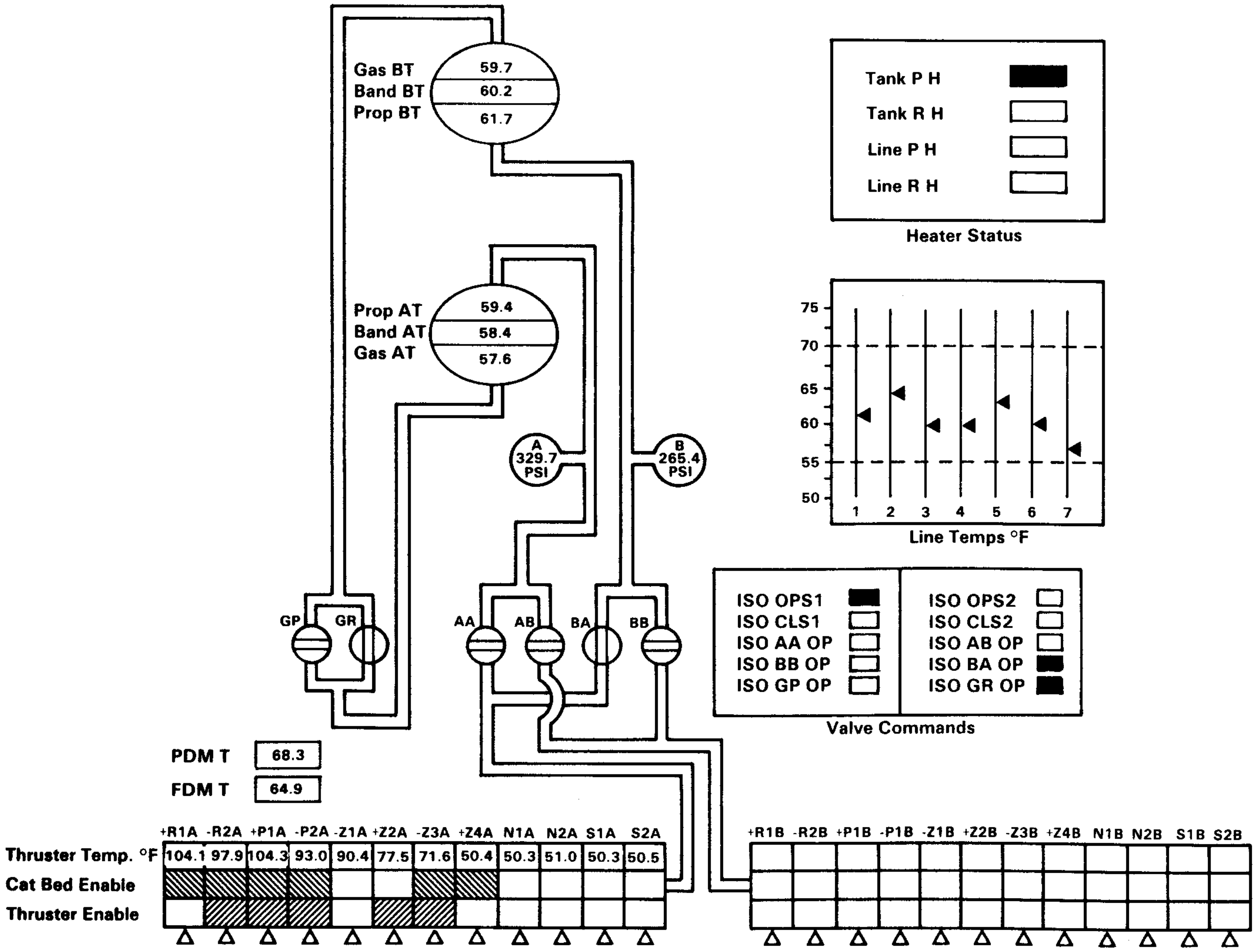

Figure 2. Graphics Display