

N 9 1 - 2 2 7 7 9

## The Generic Spacecraft Analyst Assistant (GenSAA): A Tool for Automating Spacecraft Monitoring with Expert Systems

**Peter M. Hughes**

*NASA/Goddard Space Flight Center*

**Edward C. Luczak**

*Computer Sciences Corporation*

### Abstract

Flight Operations Analysts (FOAs) in the Payload Operations Control Center (POCC) are responsible for monitoring a satellite's health and safety. These analysts closely monitor real time data looking for combinations of telemetry parameter values, trends, and other indications that may signify a problem or failure. As satellites become more complex and data rates increase, FOAs are quickly approaching a level of information saturation.

The FOAs in the spacecraft control center for the COBE (Cosmic Background Explorer) satellite are currently using a fault-isolation expert system named the Communications Link Expert Assistance Resource (CLEAR), to assist in isolating and correcting communications link faults. Due to the success of CLEAR and several other systems in the control center domain, many other monitoring and fault-isolation expert systems will likely be developed to support control center operations during the early 1990s.

To facilitate the development of these systems, a project has been initiated to develop a domain-specific tool, named the Generic Spacecraft Analyst Assistant (GenSAA). GenSAA will enable spacecraft analysts to easily build simple real-time expert systems that perform spacecraft monitoring and fault isolation functions. Expert systems built using GenSAA will assist spacecraft analysts during real-time operations in spacecraft control centers.

This paper describes lessons learned during the development of several expert systems at Goddard, thereby establishing the foundation of GenSAA's objectives and offering insights on

how problems may be avoided in future projects. This will be followed by a description of the capabilities, architecture, and usage of GenSAA along with a discussion of its application to future NASA missions.

### Introduction

NASA's Earth-orbiting scientific satellites are becoming increasingly sophisticated. They are operated by highly trained personnel in the mission's Payload Operations Control Center (POCC). Currently at the Goddard Space Flight Center (GSFC), missions utilize either a dedicated control center (e.g. LANDSAT and the Hubble Space Telescope) or share resources in the Multi-Satellite Operations Control Center (e.g. Cosmic Background Explorer and the Gamma Ray Observatory). In either case, POCC personnel called Flight Operations Analysts (FOAs), are responsible for the proper command, control, health, and safety of the satellite.

The satellite control centers operate round-the-clock throughout the lifetime of the spacecraft. There are typically multiple real-time communications events daily with each satellite. During these events, the FOAs must:

- establish and maintain the telecommunications link with the spacecraft,
- monitor the spacecraft's health and safety,
- send commands or command loads to the satellite for on-board execution,
- oversee the transfer of the scientific data from the on-board tape recorders to ground systems for processing and analysis, and
- manage spacecraft resources (including on-board memory, batteries, and tape recorders).

To accomplish these activities, the analyst must thoroughly understand the operation of the spacecraft and ground systems and continuously monitor the current state of operations as indicated by telemetry parameters displayed on the POCC consoles. During an event, the analyst typically monitors hundreds of telemetry parameter values on multiple display pages that may be updating several times per second. Monitoring the operation of these satellites is a demanding, tedious task that requires well-trained individuals who are quick-thinking and composed under pressure.

As satellites become more complex, they become more difficult to operate. FOAs are reaching a level of information saturation as more and more data must be monitored and analyzed during real-time supports. Large volumes of low-level information can overwhelm analysts and disrupt their ability to identify and resolve conflicting constraints. Human operators may soon be unable to consistently monitor all of the information available. The need to automate some data monitoring functions is apparent.

Expert system technology is proving to be effective in automating some spacecraft monitoring functions. This paper first summarizes CLEAR, the first spacecraft monitoring expert system deployed at GSFC. The paper then reviews several lessons learned from CLEAR and other monitoring and fault isolation expert system projects undertaken at GSFC. Finally, the paper describes the Generic Spacecraft Analyst Assistant (GenSAA), a tool that will facilitate the development of future spacecraft monitoring expert systems. GenSAA has been defined based on the lessons learned from CLEAR and other expert system projects at GSFC.

### **Initial Work: The CLEAR System**

The Communications Link Expert Assistance Resource (CLEAR) is the first operational expert system at GSFC that automates one of the spacecraft analyst's tasks (Hughes & Hull, 1987). CLEAR is a fault-isolation expert system that supports real-time operations in the POCC for the Cosmic Background Explorer (COBE) mission. This system monitors the communications link between COBE and the Tracking and Data Relay Satellite (TDRS), alerts the analyst

to any problems, and offers advice on how to correct them.

CLEAR is a forward chaining, rule-based system that operates in the COBE POCC. It monitors over 100 real-time performance parameters that represent the condition and operation of the spacecraft's communications with the relay satellite. Using this information, together with knowledge of TDRS operations, COBE's on-board communications system and the expected configuration of the scheduled event, CLEAR accurately portrays the status of the communications link.

Textual and graphical information about the condition of the COBE/TDRS/ground communications links is displayed in a tiled-window format. A graphics window displays the elements of the communications network from the COBE Spacecraft to the POCC; green lines represent healthy links between elements. When the performance parameters indicate that a communications link or processing system is degrading or down, the associated line or icon turns yellow or red, respectively. The display enables analysts to assess the current status of the communications event in a quick glance.

When CLEAR isolates a problem, a short description of the problem is displayed in a "problems" window. If multiple problems are found, the problem descriptions are ranked and displayed in descending order of criticality. CLEAR suggests analyst actions to correct the problem; however, the system does not take any corrective action itself.

To further assist the analyst and to provide support for its advice, the CLEAR system provides an explanation facility. When the analyst selects a problem displayed in the problems window, CLEAR provides a detailed explanation of why the expert system believes that the problem exists.

CLEAR has approximately 165 rules and isolates approximately 75 different problems. The types of problems include: non-reception of data within the control center (system or communication problems, or data reporting not activated); misconfigurations between the COBE POCC and the TDRS ground station

(coherency/non-coherency, doppler compensation on/off, power mode, actual TDRS in use, antennae configurations); discrepancies in telemetry rate or format; inactive or non-locked links; and degrading or critical signal strength situations (Perkins & Truskowski, 1990).

CLEAR operates on any of the seven PC/AT-class workstations that are used for console operations in the POCC. It is written in the 'C' language and uses the 'C' Language Integrated Production System (CLIPS) and a custom-developed graphics library.

The CLEAR Expert System has supported the COBE Flight Operations Team since launch in November 1989. It is used to monitor nearly all of the TDRS supports (COBE occasionally communicates directly to the Wallops ground station) and is regarded as the fault-isolation "expert" for the COBE/TDRS telecommunications link. CLEAR represents a successful attempt to automate a control center function using an expert system. Several other missions have requested to use it and, at the time of this writing, efforts are underway to adapt it to support the Gamma Ray Observatory mission which is scheduled for launch aboard the shuttle in Spring 1991.

### Lessons Learned

Several important lessons have been learned from the experience gained in developing and operating CLEAR. Key lessons have also been learned from other monitoring and fault isolation expert systems developed recently at GSFC, including the Ranging Equipment Diagnostic Expert System (REDEX) (Luczak, et. al., 1989), and other systems. These lessons learned have strongly influenced the definition of GenSAA.

- **Production rules effectively represent analysts' knowledge for automating fault-isolation in spacecraft operation.** The rule-based method of knowledge representation has proven to be quite powerful for fault-isolation in spacecraft operations. Production rules provide a direct method of encoding the fault-isolation knowledge of spacecraft analysts; the if-then structure closely parallels the stimulus-response behavior that they develop through extensive training. This

knowledge can be translated smoothly into rule form. The development of CLEAR would have taken much longer using conventional, non-rule-based programming techniques.

- **Knowledge engineering is an iterative, time-consuming process.** Early in CLEAR's development, the primary concern was the perceived difficulty of the knowledge acquisition effort. However, the knowledge engineering task was found to be relatively straightforward, albeit time-consuming. The development of the rule base was a lengthy process due to the interactive nature of the knowledge acquisition. Basically, the expert would describe a specific piece of knowledge to the "knowledge engineer" who would attempt to transcribe it into a rule and pass it back to the expert for validation. When the rule accurately represented the piece of knowledge (which usually took multiple iterations between the expert and the knowledge engineer), it was passed to the test team for formal testing, and then, finally, released for operational use.

The involvement of various players in this process resulted in long turnaround times from the point at which a piece of knowledge was determined to be important until it was translated into a rule and placed into operation.

- **Allow the domain expert to participate in the rule formation.** The CLEAR development team learned that the knowledge structure of the fault-isolation process employed by the FOAs is shallow (i.e. the identification of a problem generally does not rely on the identification of other subproblems, and so on). Most of the problems identified by the analysts were discrete problems that seldom overlapped other problems. Conflicts between rules were minimal; this simplified testing, verification, and validation of the rulebase.

The participation of the analyst in knowledge acquisition and translation has many advantages. First, it can reduce the knowledge translation time and, more importantly, reduce knowledge translation errors that occur when a knowledge engineer formulates rules based on the knowledge extracted from documentation or interviews with the domain expert. Second, the verification and validation of the knowledge will be facilitated since the expert will better

comprehend the rulebase. Third, the in-depth understanding of the knowledge base and its capabilities is likely to result in a higher degree of user confidence in the system thereby ensuring user acceptance.

- **Expect to fine-tune the expert system after it becomes operational.** For CLEAR, the rule-based method of knowledge representation has provided the flexibility to easily adapt the knowledge base to unforeseen changes in the operational behavior of the spacecraft. For example, even though the operational nature of COBE was fairly accurately understood by the design engineers and flight operations team before the launch, slight behavioral variations and complications arose once the spacecraft was in orbit. Although the FOAs were able to adjust to such variations quickly, some of the ground systems required complex software modifications. However, the changes required to CLEAR's rule-base were simple and quickly implemented. After modification, CLEAR provided consistent operational assistance. It is important to provide the capability to modify an operational expert system in a controlled, yet straightforward way.

- **Don't underestimate the integration process.** One of the most valuable lessons learned is that while prototypes can often be developed rapidly, operational expert systems require considerable effort. A major factor in this effort is the difficulty of interfacing the expert system to the data source.

The CLEAR development team learned that most of the development time for the operational system was spent on issues not directly related to the construction of the knowledge base. A surprising amount of effort focused on the integration of the expert system with the data source and graphics display system. This required in-depth programming knowledge of the interfacing systems and the ability to troubleshoot problems within them. Provide tools to simplify the complicated task of integrating the expert system with the interfacing systems and, if possible, reuse any interface code developed for a similar (expert) system.

- **Don't neglect the user-interface.** The human-computer interface is frequently the

most underdeveloped component of an expert system. An effective user interface is inviting, comprehensible, credible, and simple to operate. To make it inviting, simplify the display layout and present only that information needed to efficiently perform the task. Graphics can greatly enhance the visual communications of a system; capitalize on their expressive power to provide system output that can be assimilated quickly and accurately.

The following lessons are also related to the use of graphical user interfaces:

- **Use colors prudently and consistently.** Although often misused, colors are valuable for emphasizing or coding information. Use them sparingly and in a manner consistent with other systems or conventions employed in the targeted operational environment.

- **Include a graphical user-interface (even in the first expert system prototype.)** CLEAR utilized a graphical user interface in the initial prototype to demonstrate the capabilities of the proposed expert system; this elicited valuable feedback from the expert and other FOAs. In contrast, a non-graphical user-interface was used in the initial REDEX prototype; as a result, user interest and feedback was limited early in the project

- **Use an object-oriented diagram editor to ease diagram creation and maintenance.** Ideally, the diagram editor should enable diagram objects to be easily associated with status values and fault conditions inferred by the expert system. In the REDEX project, a diagram editor with only limited capability was used, and as a result, significant effort was required to construct and modify diagrams.

- **Use block diagram displays to graphically illustrate the system being monitored.** Users have responded very positively to the use of schematic displays that graphically show monitored system status and fault locations. Analysts and technicians usually learn about the systems they monitor by studying system block diagrams in training classes and reference manuals. By using similar block diagram displays, a monitoring expert system can present status to the user in a familiar and intuitive format. Color coding of status

conditions on such displays has been found to be an effective way to present succinct status summaries. For example, REDEX users have been enthusiastic about the color block and layout diagrams in that expert system; over 35 diagrams graphically depict the functional and physical structure of the equipment being diagnosed.

- **Make the system easy to operate.** Operation of the expert system should be simple enough that the user can concentrate on the problem, not on how to operate the system. The following techniques were applied in CLEAR and REDEX to simplify operation:

- **Reduce user input to a minimum.** CLEAR operates in a highly autonomous mode; no user input is required after system initialization. CLEAR has been well-accepted by its users, partially because it operates as an independent intelligent assistant, allowing the spacecraft analyst to focus on other responsibilities during real-time satellite contacts.

- **Use hypertext and hypergraphic techniques.** These techniques (Bielawski & Lewand, 1991) enable the expert system user to quickly select and display desired diagrams by clicking on link buttons that appear on each diagram. Links can be used to create diagram hierarchies, off-page connections, diagram sequences, and other structures. REDEX uses these techniques to enable users to navigate through a set of several dozen graphical display pages; they find the approach intuitive and easy to use.

These lessons learned have all influenced the definition and development of GenSAA.

## GenSAA

### Overview

GenSAA is an advanced tool that will enable spacecraft analysts to rapidly build simple real-time expert systems that perform spacecraft monitoring and fault isolation functions. Expert systems built using GenSAA will assist spacecraft analysts during real-time operations in spacecraft control centers.

Use of GenSAA will reduce the development time and cost for new expert system applications in this domain. GenSAA will allow major expert system software functions and portions of knowledge bases to be reused from mission to mission.

GenSAA has the following primary characteristics:

- **Easily used** — The process for developing specific expert system applications using GenSAA will be straightforward enough that it can be performed by trained spacecraft analysts on the flight operations team.

- **Rule-based** — GenSAA will support the use of rules to represent spacecraft and payload monitoring and fault isolation knowledge. Rule-based representations are easily learned and can be used to describe many of the reasoning processes used by spacecraft analysts. (An object representation technique will be included in a subsequent phase.)

- **Highly graphical** — The GenSAA operational user interface will support both textual and graphical presentations of health and status information and fault isolation conclusions. Hypertext and hypergraphic techniques will be supported to simplify operational interaction with GenSAA expert systems.

- **Transparently interfaced with TPOCC** — GenSAA will be used to create expert system applications that will support analysts in spacecraft control centers that use the new Transportable Payload Operations Control Center (TPOCC) architecture. TPOCC is a new Unix-based control center system architecture that will be used on many new spacecraft missions at GSFC. GenSAA will be adaptable to also support non-TPOCC data interfaces.

GenSAA is being defined and prototyped by the Automation Technology section (Code 522.3) of the Data Systems Technology Division at GSFC. A system and operations concept has been defined (Hughes & Luczak, March 1990), and a multi-phase prototype effort is underway (Hughes & Luczak, August 1990).

## GenSAA Architecture

GenSAA is a shell, or software framework for developing spacecraft control center expert systems. It is analogous to many commercial expert system shells because it facilitates the development of specific expert system applications. However, GenSAA is tailored to the specific requirements of spacecraft analyst assistant expert systems in TPOCC control centers. GenSAA therefore shares many of TPOCC's architectural features.

The TPOCC architecture is based on distributed processing, industry standards, and commercial hardware and software components. It employs the client/server model of distributed processing, the Network File System (NFS) protocol for transparent network access to files, and the X Window System (X.11) with the Motif library and window manager for the graphical operator interface. A TPOCC configuration consists of a small set of specialized front-end processors and Unix-based workstations on an Ethernet network using the TCP/IP network protocol. GenSAA operates in this environment.

GenSAA allows spacecraft analysts to rapidly create simple expert systems without having to deal directly with the complicated details of the systems with which the expert system must interface. In addition, it will allow the expert system developer to utilize and/or modify

previously developed rule bases and system components, thus facilitating software reuse and substantially reducing development time and effort.

Figure 1 shows the six major elements of GenSAA. They are divided into two sets: the GenSAA Workbench and the GenSAA Runtime Environment. These are described in the sections below.

### The GenSAA Workbench

The GenSAA Workbench is composed of three utilities that enable a spacecraft analyst to create a GenSAA application. A GenSAA application is a specific expert system that performs real-time monitoring and fault isolation functions in a TPOCC spacecraft control center.

A GenSAA application is created by defining the application's runtime specifications using the GenSAA Workbench. Figure 2 illustrates that these specifications, called Reusable Application Components, together with the elements of the GenSAA Runtime Components, comprise a GenSAA Application.

The GenSAA Workbench utilities are as follows:

- **Data Interface Development utility** – This utility is used to create the Data Interface

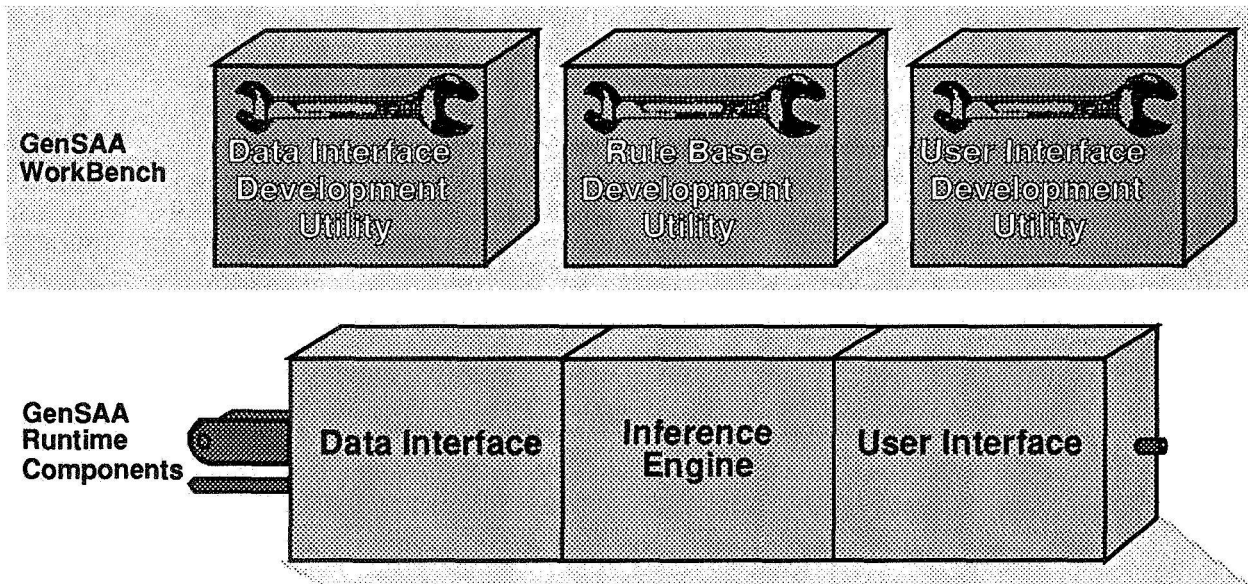


Figure 1. The Elements of GenSAA

Specification for a GenSAA application. The Data Interface Specification defines three types of data that are used by the GenSAA application during real-time operations:

- *Telemetry data* - Telemetry data variables represent real-time status of the monitored spacecraft and related ground support systems. (Telemetry variables are sometimes called telemetry mnemonics.) Values for these variables are received and updated during spacecraft operation periods from the TPOCC Data Server process, which is part of the TPOCC software. Using the Data Interface Development Utility, the GenSAA Workbench user selects the telemetry variables needed for the expert system being created from a list of all the telemetry variables available from the TPOCC Data Server. Values for only those variables selected will be received by the expert system during run-time.

- *Configuration data* - Configuration data variables represent expected operating modes and equipment configurations. For example, a configuration variable might represent the setting of a switch that determines which of two redundant components is to be used. Values for these variables are entered by the spacecraft analyst during spacecraft operation periods.

- *Inferred data* - Inferred data variables represent conclusions inferred by rules in the rule base. For example, an inferred data variable might represent the health or fault status of a component in a spacecraft subsystem. (The name of an inferred data variable together with its current value is called an *inferred fact*.) Values are assigned to these variables by actions executed in the "then" part of rules that fire.

• **Rule Base Development utility** - This utility is used to create the rule base for a GenSAA application. The rule base is a set of expert system rules in "condition-action" ("if - then") format that may infer new facts based on currently asserted facts. The inference engine controls the firing of rules in the rule base during execution of the GenSAA application.

During run-time, if all the conditions of a rule are satisfied, then the rule fires and all its actions are performed. Conditions can be

constructed using the telemetry, configuration, and inferred data variables specified with the Data Interface Development Utility. Actions may include: asserting/retracting an inferred fact, enabling/disabling a rule or ruleset, performing a mathematical calculation, and displaying text messages on the user interface.

• **User Interface Development utility** - This utility is used to create the User Interface Specification for a GenSAA application. The User Interface Specification defines the user interface panels and the layout and behavior of the display items that comprise the operational user interface of the GenSAA application. The Workbench user can create a variety of display items, including graphical icons, scrolling text lists, and data-driven objects such as meters, gauges, and simple strip charts. The display designer constructs a panel by dragging display items from a palette and placing them wherever desired. Lines can be drawn using connector items to create block diagram displays. The Workbench user can associate each display item with a telemetry, configuration, or inferred data variable, and specify how changes in the value of the variable will affect the presentation of the item. Characteristics of an item presentation that can change include its color, the icon displayed, and the position of the dynamic portion of a data-driven object. Simple drawing capabilities are provided to allow the creation of new graphical icons. Any display item can also be specified to be a hypertext button; clicking on such a button during run-time can cause an informational pop-up window to appear, or cause a new panel to be displayed.

The GenSAA Workbench utilities use a graphical, point-and-select method of interaction to facilitate use. The utilities are also highly inter-operable. For example, when using the Data Interface Development utility, the user may select a given telemetry mnemonic to be included in the Data Interface Specification. Later, when using the Rule Base Development utility, the user can easily access the Data Interface Specification and to reference the mnemonic in a condition of a rule. Similarly, when using the User Interface Development utility, the user can again easily access the Data Interface Specification when associating telemetry mnemonics with display objects.

In Figure 2, the outputs of the GenSAA Workbench utilities are described as reusable application components. These specifications will be placed in a library so that they can be reused in creating the specifications for new GenSAA applications. Operations like cut and paste will be available to allow portions of previously created specifications to be used in constructing a new specification.

**GenSAA Runtime Environment**

The elements of the GenSAA Runtime Environment are called the GenSAA Runtime Components; they are used without change in each GenSAA application. They control the operation of a GenSAA application during its execution in a TPOCC control center. They read the Data Interface Specification, Rule Base, and User Interface Specification files to determine the specific behavior of the GenSAA application. Each of the GenSAA Runtime Components is implemented as a separate Unix process; they communicate with one another via shared memory and message queues. Their functions are as follows:

- **Data Interface** – This component requests

telemetry from the TPOCC Data Server, as specified in the Data Interface Specification. It reformats the real-time data it receives and makes it available to the Inference Engine and User Interface components. (It also exchanges data with the GenSAA Data Server, as described below in the section Multiple GenSAA Applications.)

- **Inference Engine** – This component controls the firing of rules in the rule base. A rule is fired when all its conditions are satisfied; the conditions will often involve the current values of telemetry, configuration, and inferred data variables. Inferred facts and messages may be sent to the User Interface component and displayed to the spacecraft analyst as defined in the User Interface Specification. NASA's CLIPS inference engine forms the core of this component.

- **User Interface** – This component manages the user interface of the GenSAA Application. It displays user interface panels that contain both text and graphics. Color is used to enhance the display of state data. Data-driven display objects are associated with telemetry values received from the TPOCC data server and

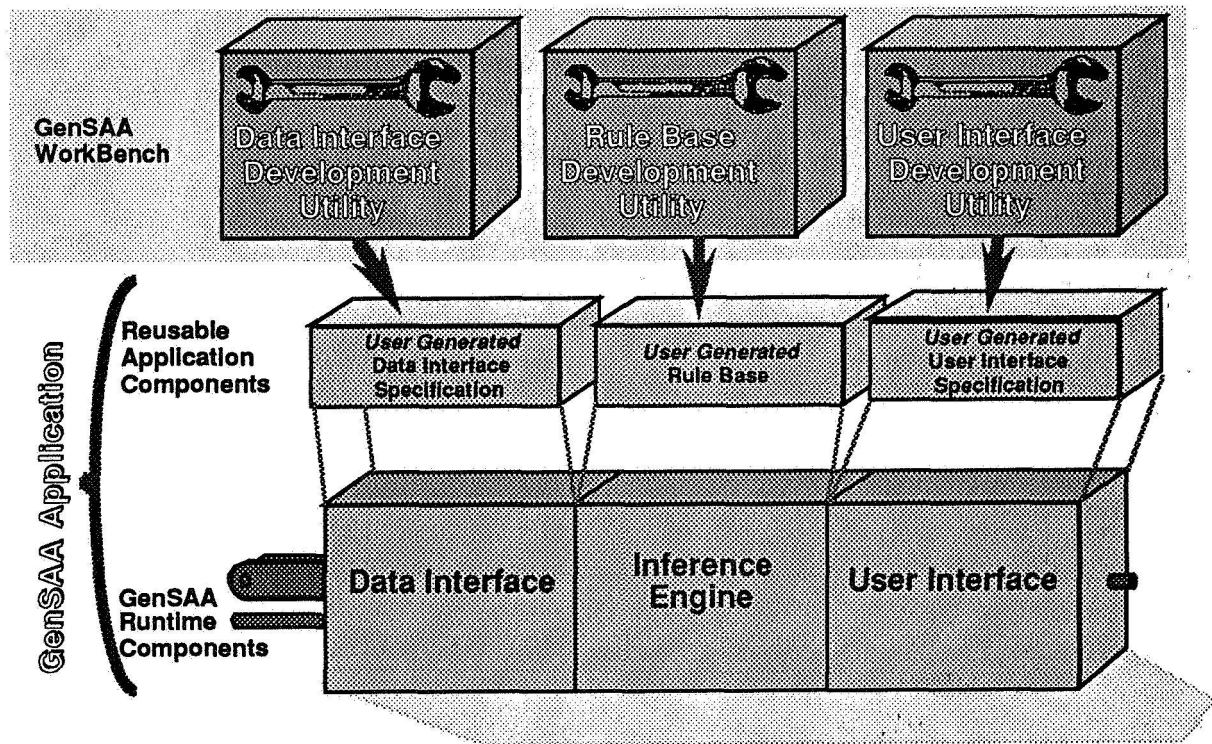


Figure 2. Creating a GenSAA Application



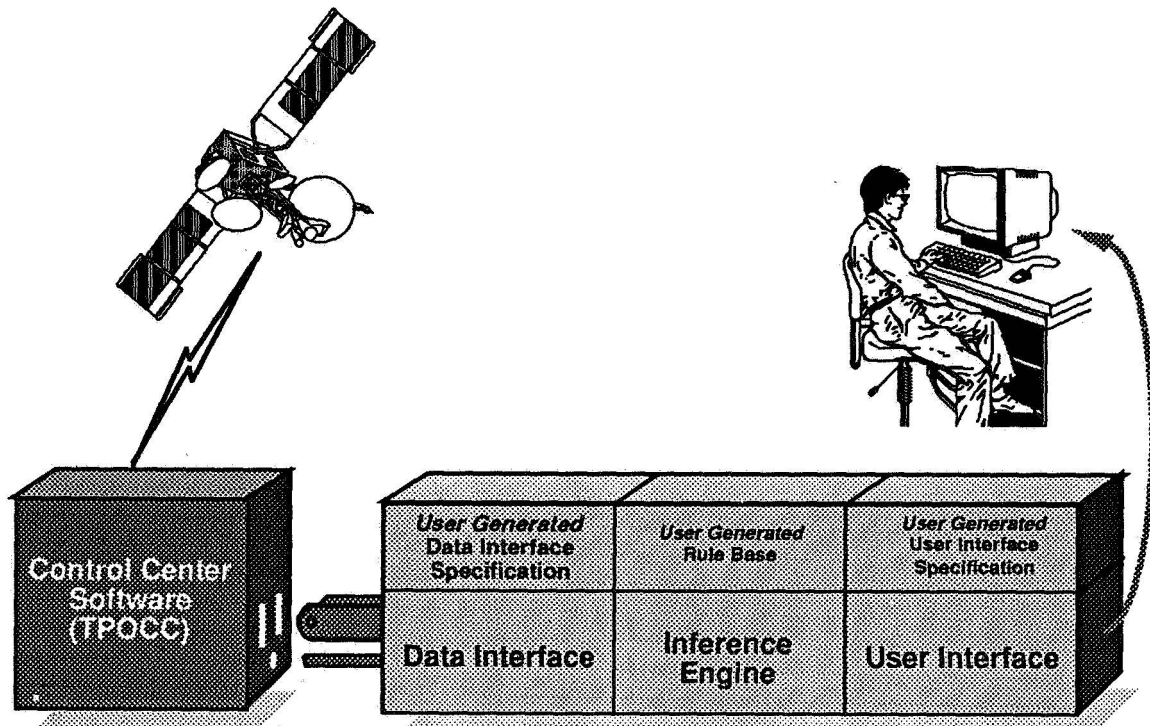


Figure 3. A GenSAA Expert System Application in operation

inferred facts and conclusions received from the Inference Engine. In response to user inputs that include hypertext button events, the User Interface displays selected display panels, help text, and other informational text. The user interface panels, data-driven objects, and interaction objects are defined in the User Interface Specification that was generated by the GenSAA User Interface Development utility.

Figure 3 shows a completed GenSAA expert system application in operation. GenSAA expert systems will run on Unix workstations using the X Window System. The operational interface with the spacecraft analyst will typically include color block diagrams and animated data-driven objects (such as rotating meters, sliding bar graphs, and toggle switches) that graphically display the dynamic values of telemetry data, configuration data, and inferred conditions. The user interface will also typically contain hypertext and hypergraphic links to make it easy for the spacecraft analyst to quickly select desired display panels. The GenSAA Workbench supports the creation of these interface features.

### *Multiple GenSAA Applications*

GenSAA applications are intended to be relatively simple expert systems with small rule bases that are typically developed by a single analyst. For example, a GenSAA application might monitor and isolate faults for one subsystem on board a spacecraft. To handle more complex monitoring situations, involving for example several spacecraft subsystems, multiple GenSAA applications can be built and executed concurrently. Each GenSAA application would be allocated one portion of the monitoring task, and share key conclusions with one another.

A fourth component of the GenSAA Runtime Environment, the GenSAA Data Server, is used to enable multiple GenSAA applications to exchange data. As shown in Figure 4, the GenSAA Data Server is a Unix process that can receive a real-time stream of configuration and inferred data variable updates from any GenSAA application. The GenSAA Data Server distributes the data to any GenSAA application that has requested it. A given GenSAA

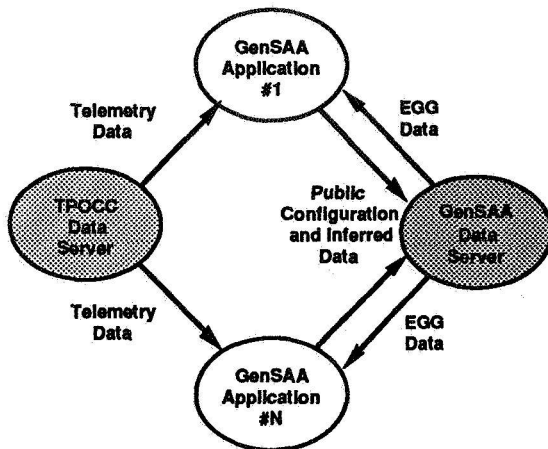


Figure 4. Sharing data among multiple GenSAA Applications

application only receives those variables it has specifically requested. The data received by a GenSAA application from the GenSAA Data Server is called externally generated GenSAA (EGG) data. A GenSAA application receives EGG data via its Data Interface component in exactly the same way as it receives telemetry data from the TPOCC Data Server.

Within a GenSAA application, EGG data can be used in the conditions of rules, and can be associated with display items in exactly the same way as telemetry, configuration, and inferred data. The Workbench supports the specification of EGG data as a fourth variable type. The Workbench also allows any local configuration or inferred data to be specified as public, to cause it to be sent to the GenSAA Data Server, and thereby shared with other GenSAA applications.

### **Benefits of GenSAA**

The following benefits are expected to be realized by using GenSAA to build spacecraft monitoring expert systems for future NASA missions:

- **Assists the FOAs with data monitoring—**FOAs monitor real time data looking for combinations of telemetry parameter values, trends, and other indications that may signify a problem with the satellite or its instruments. The expert systems created with GenSAA will assist the FOAs with the tedious task of data monitoring and allow them to focus on other,

higher-level responsibilities during real-time contacts with the satellite. This, in turn, will likely result in more efficient and effective operations.

- **Reduces development time and effort; allows quicker response to necessary modifications —** The behavior of an orbiting satellite is quite dynamic and occasionally different than anticipated. To quickly create or modify expert systems that can effectively monitor satellites, tools are needed that allow analysts to formulate rulebases easily without the intervention or delay of knowledge engineers and programmers. Several benefits are expected by eliminating these traditional developers. Analysts will be able to create rules quickly in response to unforeseen changes in spacecraft behavior or operational procedures. Also, knowledge translation errors will be reduced or, at least, more easily corrected. Knowledge translation errors are errors which are inadvertently introduced during the process of translating a piece of expert knowledge into rule form.

- **Serves as a training tool —** In addition to assisting the FOAs with real-time spacecraft operations, GenSAA will be useful as a training tool in two ways. First, by utilizing the playback utilities provided by TPOCC, analysts will be able to replay a previous spacecraft communications event. Thus, a student analyst can observe how the expert system handles a specific problem scenario. Exercises like this will provide a realistic, hands-on environment for training FOAs in a safe, off-line mode. Second, experience from previous expert system projects indicates that the development of rules used in an expert system is a beneficial mental training exercise for the FOA. When FOAs create rules themselves, they must consider alternatives more closely and may therefore develop a deeper understanding of the problem domain. This approach may enable more effective fault isolation methods to be identified.

- **Protects against loss of expertise —** Another benefit of automating fault-isolation tasks with rule-based systems is that the resulting rulebase serves as accurate documentation of the fault-isolation method. The rulebase can be studied by student analysts to

learn about fault-isolation techniques. Even more importantly, mission operations can be better protected against the effects of personnel turnovers. POCC expert systems that capture fault-isolation knowledge preserve expertise from mission to mission and mitigate the impact of the loss of experienced FOAs.

GenSAA is well suited for use on spacecraft projects that involve a series of similar but nonidentical missions such as NASA's Small Explorer (SMEX) and International Solar-Terrestrial Physics (ISTP) programs.

### Conclusion

As satellites become more complex, their operation is becoming increasingly difficult. FOAs who are responsible for the command, control, health, and safety of these spacecraft must monitor increasing volumes of data, and are quickly reaching a level of information saturation. As demonstrated by the CLEAR Expert System, fault-isolation expert systems can help FOAs monitor the flood of data. Expert systems can accurately monitor hundreds of real-time telemetry parameters, isolate discrepancies and anomalies the instant they can be detected, and alert the analysts and provide advice on how to correct problems swiftly and effectively. Unfortunately, development of these systems is often time consuming and costly moreover, they often cannot be easily reused for other missions.

Consequently, GenSAA is being developed for use by the FOAs who work in satellite control centers. GenSAA is designed to enable fault-isolation expert systems to be developed quickly and easily, and without the delay or costs of knowledge engineers and programmers. By facilitating the reuse of expert system elements from mission to mission, GenSAA will reduce development costs, preserve expertise between missions and during periods of personnel turnover, and provide more effective spacecraft monitoring capabilities on future missions.

### References

- Bielawski, L., & Lewand, R. (1991). *Intelligent Systems Design: Integrating Expert Systems, Hypermedia, and Database Technologies*, New York: John Wiley & Sons.
- Hughes, P.M. (1989). *Integrating Expert Systems into an Operational Environment*. AIAA Computers in Aerospace VII Conference, Monterey, California.
- Hughes, P.M., & Hull, L.G. (1987, May). CLEAR: Communications Link Expert Assistance Resource. 1987 Goddard Conference on Space Applications of Artificial Intelligence and Robotics, Greenbelt, Maryland.
- Hughes, P.M., & Luczak, E. (1990, March). Generic Spacecraft Analyst Assistant-Concept Definition. Data Systems Technology Laboratory Document DSTL-90-008. Greenbelt, Maryland: NASA/Goddard Space Flight Center.
- Hughes, P.M., & Luczak, E. (1990, August). Generic Spacecraft Analyst Assistant-Prototype Definition. Data Systems Technology Laboratory Document DSTL-90-033. Greenbelt, Maryland: NASA/Goddard Space Flight Center.
- Luczak, E.C., Gopalakrishnan, K., & Zillig, D.J. (1989, May). REDEX: The Ranging Equipment Diagnostic Expert System. 1989 Goddard Conference on Space Applications of Artificial Intelligence, Greenbelt, Maryland.
- Perkins, D., & Truszkowski, W. (1990). *Launching AI in NASA Ground Systems*. AIAA/NASA Second International Symposium on Space Information Systems, Pasadena, California.