The Generic Spacecraft Analyst Assistant (GenSAA): A Tool for Developing Graphical Expert Systems

Peter M. Hughes
Computer Engineer/GenSAA Project Manager
Software and Automation Systems Branch (Code 522)
NASA/Goddard Space Flight Center
Greenbelt, Maryland 20771

Abstract

During numerous contacts with a satellite each day, spacecraft analysts must closely monitor real time data watching for combinations of telemetry parameter values, trends, and other indications that may signify a problem or failure. As the satellites become more complex and the number of data items increases, this task is becoming increasingly difficult for humans to perform at acceptable performance levels. At the NASA Goddard Space Flight Center, fault-isolation expert systems are in operation supporting this data monitoring task. Based on the lessons learned during these initial efforts in expert system automation, a new domain-specific expert system development tool named the Generic Spacecraft Analyst Assistant (GenSAA), is being developed to facilitate the rapid development and reuse of real-time expert systems to serve as fault-isolation assistants for spacecraft analysts. Although initially domain-specific in nature, this powerful tool will readily support the development of highly graphical expert systems for data monitoring purposes throughout the space and commercial industry.

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[1] [3].

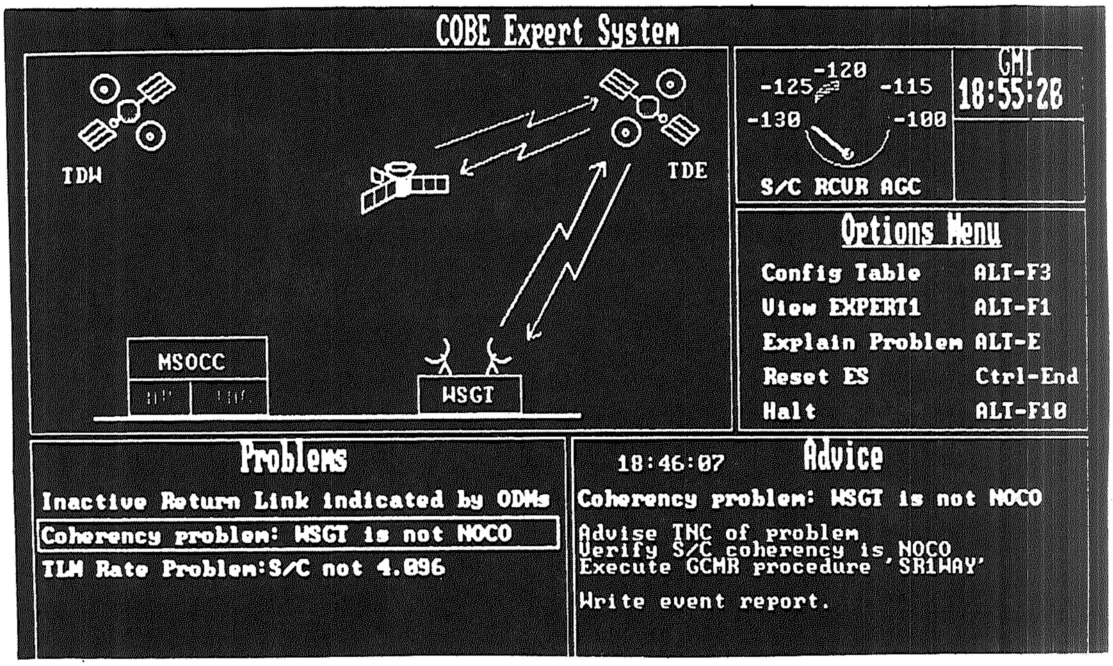
The satellite control centers operate round-the-clock throughout the lifetime of the spacecraft. There are typically multiple real-time communications events daily with each satellite. During these events, the FOAs must:

- 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 updated several times per second. Such large volumes of low-level information can overwhelm analysts and disrupt their ability to identify and resolve conflicting constraints. They 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 real-time 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, thereby establishing the foundation of a domain-specific expert system development tool called the Generic Spacecraft Analyst Assistant (GenSAA). This new tool will be introduced followed by a discussion of its capabilities, architecture and benefits, its potential uses in industry, and the approach for adapting this tool to other domains.

[1] This paper is based on a previous publication by Hughes, P. & Luczak, E. (October, 1991), reference 3.



Inactive Return Link indicated by ODMs

Coherency problem: WSGT is not NOCO

TLM Rate Problem:S/C not 4.096

18:46:07

Coherency problem: WSGT is not NOCO

Advise TNC of problem
Verify S/C coherency is NOCO
Execute GCMR procedure 'SR1WAY'

Write event report.

Figure 1. Black & White Photo of the CLEAR User Interface

CLEAR: Breaking New Ground

The Communications Link Expert Assistance Resource (CLEAR) is the first operational expert system at GSFC that automates one of the spacecraft analyst's tasks [2]. It is a fault-isolation expert system that supports real-time operations in the POCC for the Cosmic Background Explorer (COBE) mission. CLEAR 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 (Figure 1). 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 the "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 [6].

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. It has been adapted for the Gamma Ray Observatory and was utilized during early orbit.

Lessons Learned

Several important lessons have been learned from the experience gained in developing and operating CLEAR [2]. 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) [5], 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. It was recognized that an integrated tool that simplified modifications to the knowledge base or user interface could significantly accelerate this process.

- 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.

- Use graphical diagrams to illustrate the system being monitored. Users have responded very positively to the use of schematic displays that graphically represent 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.

- 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 hypergraphic techniques. These techniques [1] 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.

These lessons learned have actuated the definition and development of GenSAA.

GenSAA

Overview

GenSAA is an advanced tool that will enable spacecraft analysts to rapidly build simple yet highly graphical expert systems that are capable of performing real-time spacecraft monitoring and fault isolation functions. Expert systems built using GenSAA will assist spacecraft analysts during real-time operations in spacecraft control centers. The use of GenSAA will reduce the development time and cost for new expert systems in this domain. GenSAA will allow graphical displays and fault-isolation knowledge to be reused from mission to mission.

Expert systems developed with GenSAA will have the following characteristics:

- Easily created and modified– The process for developing specific expert systems using GenSAA will be straightforward enough that it can be performed by trained spacecraft analysts on the flight operations team. No compilation step is necessary before executing the expert system.

- 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.

- Highly graphical– The GenSAA operational user interface will support both textual and graphical presentations of health and status information and fault isolation conclusions. GenSAA user interfaces are built with the GenSAA WorkBench which uses the X-window toolkit and the Motif widget set. Hyperlink techniques will be supported to simplify navigation between GenSAA windows.

- Transparently interfaced with the data source– Initially, GenSAA will be used to create expert systems 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.
- Real time– GenSAA expert systems will be driven by real time spacecraft telemetry that indicate the current status of the spacecraft and its operation.

GenSAA is being developed as a generic tool to support the development of expert systems in any TPOCC-based control center such as SAMPEX, Wind/Polar, SWAS, SOHO, and others. However the initial use of GenSAA will be targeted for the SMEX and ISTP series of missions. SAMPEX flight operations team members have expressed a need for a tool like GenSAA, and the launch timeframe for SAMPEX, the first SMEX mission, is compatible with the GenSAA development schedule.

GenSAA Architecture

GenSAA is an advanced, domain-specific tool for developing spacecraft control center expert systems. It is analogous to many commercial expert system shells because it facilitates the development of specific expert systems. However, GenSAA is tailored to the specific requirements of spacecraft analyst assistant expert systems in TPOCC control centers.

GenSAA operates in the TPOCC environment and 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.

Figure 2 shows the major elements of GenSAA. They are divided into two sets: the GenSAA Workbench and the GenSAA Runtime Environment. The Workbench is used in an off-line mode to create a specific GenSAA Expert System and the Runtime Environment is used to execute the GenSAA Expert System to support real-time operations in a spacecraft control center. These elements 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 expert system. A GenSAA expert system is a specific expert system that performs real-time monitoring and

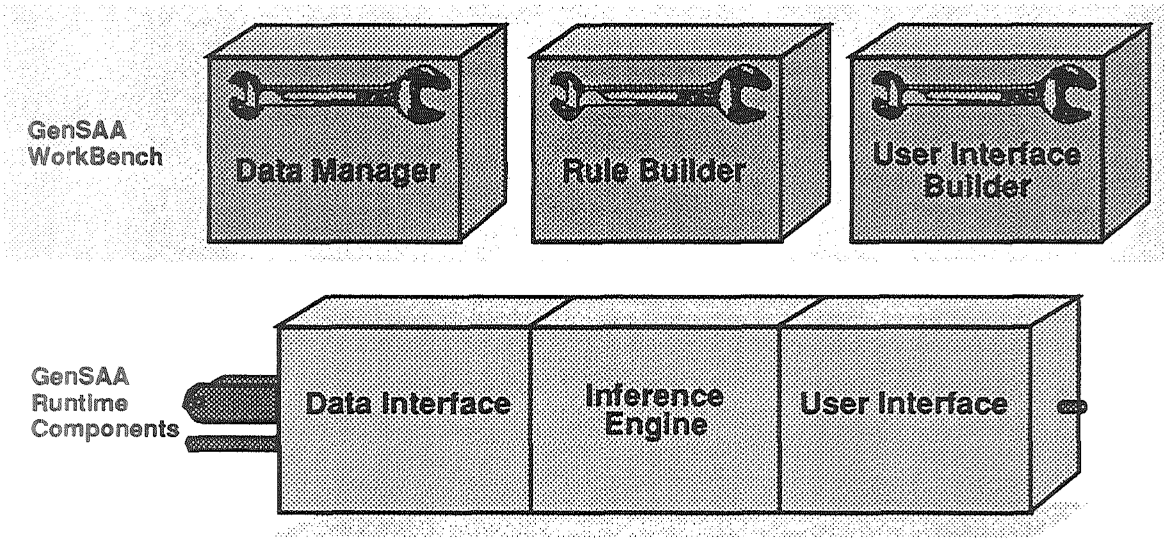


Figure 2. The Elements of GenSAA

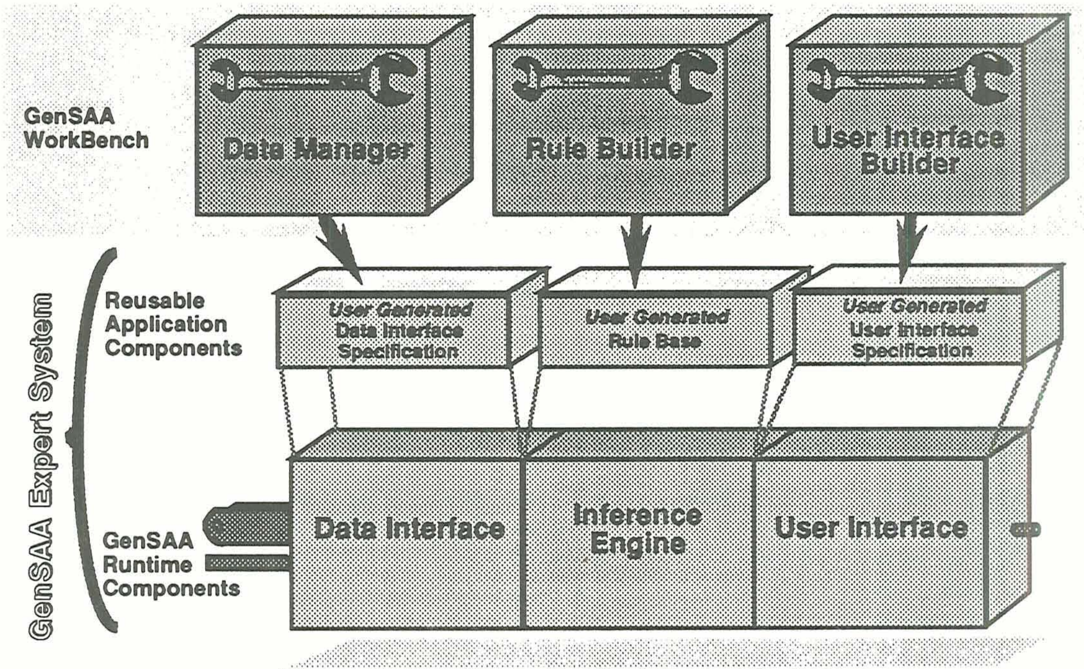


Figure 3. Creating a GenSAA Expert System

fault isolation functions in a TPOCC spacecraft control center.

The GenSAA Workbench will operate in an off-line mode on a Unix workstation. A GenSAA expert system is created by defining the expert system's runtime specifications using the GenSAA Workbench. Figure 3 illustrates that these specifications, called Reusable Application Components, together with the GenSAA Runtime Components, compose a GenSAA Expert System. The GenSAA Workbench utilities are as follows:

• Data Manager– This utility is used to create the Data Interface Specification for a GenSAA expert system. The Data Interface Specification defines four types of data that are used by the GenSAA expert system during real-time operations:

– *Mission data*– Mission data variables represent real-time status of the monitored spacecraft and related ground support systems. (mission 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 Manager, the GenSAA Workbench user selects the mission variables needed for the expert system being created from a list of all the mission variables available from the TPOCC Data Server. Values for only those variables selected will be received by the expert system during run-time.

– *User-defined data*– User-defined data variables represent expected operating modes and equipment configurations. For example, a user-defined data 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.

– *Externally Generated GenSAA (EGG) data*– EGG data consists of Inferred and User-Defined data which is identified by the user as being "public". These data are routed to the GenSAA Data Server to make them available to any process requesting them by name. For example, one GenSAA expert system may require information about the status of a subsystem which is being monitored by another GenSAA expert system. Such inter-expert system communication is conducted through EGG data.

• Rule Builder– This utility is used to create the rule base for a GenSAA expert system. 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 expert system.

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 mission, user-defined, and inferred data variables specified with the Data Manager. 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. Templates are provided for specifying conditions and actions thereby allowing a user to build rules quickly using drop-and-drag techniques.

- User Interface Builder– This utility is used to create the User Interface Specification for a GenSAA expert system. The User Interface Specification defines the user interface windows and the layout and behavior of the graphical objects that comprise the operational user interface of the GenSAA expert system.

The Workbench user can use a variety of X-toolkit and Motif widgets, including pushbuttons, option menus, scrolling text lists, user-created graphical icons, and data-driven objects such as meters and gauges. The designer constructs a user interface by selecting graphical objects from a palette or drawing them with the graphics tools provided and placing them wherever desired. Lines can be drawn using connector items to create animated schematic diagrams. The Workbench user can associate each graphical object with a mission, user-defined, inferred data, or EGG variable, and specify how changes in the value of the variable will affect the presentation of the item. Characteristics of a graphical object's behavior that can change based on the value of its associated data variable include its color, the icon displayed, and the position of the dynamic portion of a data-driven object. Simple drawing editors are provided to allow the creation of new graphical icons. Any graphical object can also be specified to be a hyperlink button; clicking on such a button during run-time can cause a window to be displayed, or cause an informational pop-up window to appear.

The GenSAA Workbench utilities are highly interoperable and use a graphical, direct-manipulation method of interaction (commonly referred as "point-and-select" or "drag-and-drop") to facilitate use. For example, when using the Data Manager, the user may select a given mission mnemonic to be included in the Data Interface Specification. Later, when using the Rule Builder, the user can drag the mnemonic from the Data Manager into a condition of a rule. Similarly, when using the User Interface Builder, the user can drag a GenSAA data variable onto a graphical item in the user interface to associate the variable with the graphical object. This pointing technique prevents keyboard mistakes and is faster than typing.

In Figure 3, the outputs of the GenSAA Workbench utilities are described as reusable application components. These components will be placed in a library so that they can be reused in creating the specifications for new GenSAA expert systems. Operations like cut and paste will be available to allow portions of previously created specifications to be used in constructing a new expert system.

GenSAA Runtime Environment

The elements of the GenSAA Runtime Environment are called the GenSAA Runtime Components; they are used without change in each GenSAA expert system. They control the operation of a GenSAA expert system 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 expert system. The GenSAA Runtime Framework is implemented as a pair of Unix processes that communicate with one another via message queues. Their functions are as follows:

- User Interface Process– This component manages the user interface of the GenSAA Expert system. It displays user interface windows 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 inferred facts and conclusions received from the Inference Engine. In response to user inputs that include hypertext button events, the User Interface displays selected graphics windows, help text, and other informational text. The user interface windows, data-driven objects, and interaction objects are defined in the User Interface Specification that was generated by the GenSAA User Interface Builder .

- Data Interface Subsystem– This sub-element of the User Interface Process requests telemetry from the TPOCC Data Server, as specified in the Data Interface Specification. It formats the real-time data it receives and makes it available to the Inference Engine and User Interface components.

- Inference Engine Process– This component manages 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, user-defined, and inferred data variables. Inferred facts and messages may be sent to the User Interface component and displayed to the FOA as defined in the User Interface Specification. NASA's 'C' Language Integrated Production System (CLIPS) inference engine forms the core of this component.

The operational interface with the FOA will typically include color schematics and animated data-driven objects (such as rotating meters, sliding bar graphs, and toggle switches) that graphically display the dynamic values of telemetry data, user-defined 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 display graphics windows.

Figure 4 shows a completed GenSAA expert system in operation. A GenSAA expert system will execute on a Unix workstation in the control center. A dedicated Unix workstation is not required, i.e., a GenSAA expert system can execute on the same workstation as other TPOCC processes. However, to avoid potential performance impacts, the initial GenSAA expert system will reside on a dedicated Unix workstation in the SMEX POCC connected to the TPOCC Local Area Network.

During operation, a GenSAA expert system will interface to the TPOCC software via the TPOCC Data Server process. This interface will use the standard external interface conventions defined by the TPOCC Project. For example, a GenSAA expert system simply submits a request to the TPOCC Data Server process for the telemetry items that were specified in the expert system. No additional data or commands are sent to any of the TPOCC processes.

Implementation

The GenSAA development team is utilizing a spiral development approach in which two prototypes and an operational system will be implemented. The first prototype, called the 'Proof-of-Concept Prototype', investigated the basic concepts of GenSAA and was completed in August, 1990. The second, called the 'Functional Prototype', demonstrated the functional characteristics of the GenSAA Workbench. This prototype was completed in October, 1991 and was used to assess and refine the functional requirements for the operational system. The operational version is scheduled for release in mid 1993.

GenSAA will be implemented in C++ using an object-oriented design. This approach has been selected because of the following four benefits: First, it will allow the reuse of an existing class library developed at GSFC (Code 522) for the rapid development of software. Second, it will promote modularity and ease integration of the software components that will comprise GenSAA. Third, it will allow the core modules of

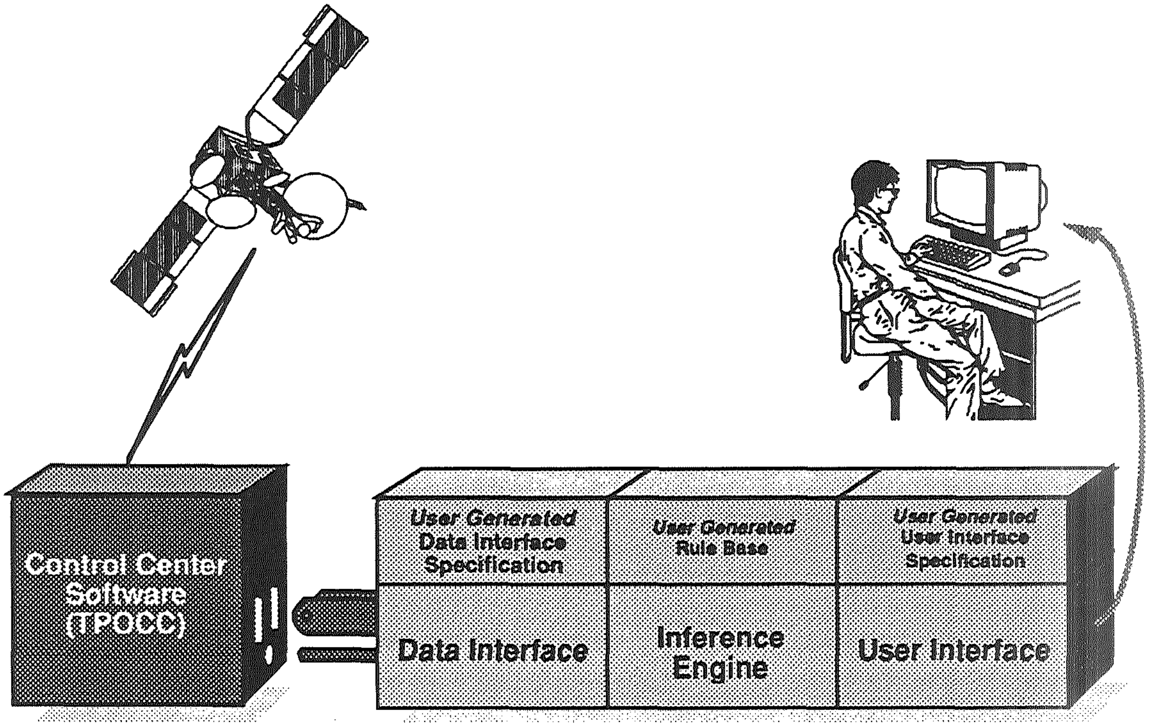

Control Center Software (TPOCC)
User Generated Data Interface Specification
User Generated Rule Base
User Generated User Interface Specification
Data Interface
Inference Engine
User Interface

Figure 4. A GenSAA Expert System in Operation

GenSAA to be implemented so that the system can be extended for future missions or industrial use without major design changes or extensive recoding. Fourth, the GenSAA development team is optimistic that the object-oriented approach will facilitate maintenance of this system.

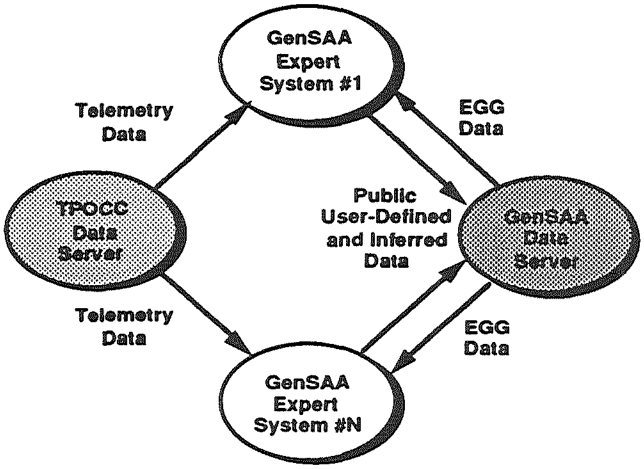


Figure 5. Sharing data among multiple GenSAA Expert Systems

Multiple GenSAA Expert Systems

GenSAA expert systems are intended to be relatively simple expert systems with small rule bases that are typically developed by a single analyst. A typical GenSAA expert system 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 expert systems can be built each responsible for a discrete subsystem or function. During operation, these expert systems would execute concurrently and could share key conclusions with one another using a "publish-and-subscribe" model of communicating.

To perform the publish-and-subscribe method of information sharing, a fourth component of the GenSAA Runtime Environment, the GenSAA Data Server, is used to serve as a central repository to which GenSAA Expert Systems can "publish" information and from which other "subscribing" GenSAA Expert Systems can receive the information when published. As shown in Figure 5, the GenSAA Data Server is a Unix process that can receive a real-time stream of user-defined and inferred data variable updates from any GenSAA expert system. The GenSAA Data Server distributes the data to any GenSAA expert system that has requested it. A given GenSAA expert system only receives those variables it specifically requested (subscribed). The data received by a GenSAA expert system from the GenSAA Data Server is called externally generated GenSAA (EGG) data. A GenSAA expert system 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 expert system, EGG data can be used in the conditions of rules, and can be associated with display items in exactly the same way as mission, user-defined, and inferred data. The Workbench supports the selection of EGG data as a fourth variable type. The Workbench also allows any local user-defined 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 expert systems.

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 quick and accurate 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 rule bases 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 rule base serves as accurate documentation of the fault-isolation method. The rule base 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.

Applicability to Industrial Systems

Although initially developed to support real-time data monitoring in satellite control centers, GenSAA will support the rapid construction of highly graphical expert systems for a variety of applications throughout industry. The Rule Builder and User Interface Builder of the GenSAA WorkBench facilitates the development of a knowledge base integrated with a graphical user interface complete with multiple windows, user input graphical objects, and data-driven graphical objects such as meters, gauges, and strip charts. Using these two WorkBench tools, for example, an organization could easily create expert systems to monitor traffic on a computer network or watch over an industrial manufacturing process, searching for problems and providing decision support or corrective action if a problem is detected.

For more complex systems, GenSAA's publish-and-subscribe method of information sharing enables multiple GenSAA expert systems to share knowledge (configuration values, system status, problem diagnoses, data analysis results) for distributed, hierarchical problem solving. For our application in the satellite control center environment, a number of individual GenSAA expert systems are planned to monitor and diagnose the subsystems onboard the spacecraft and within the ground system. These expert systems will publish their results to a "master" expert system which will monitor knowledge from a number of expert systems to provide a high level view and to isolate problems that exist across subsystem boundaries. This approach makes GenSAA applicable to a wealth of commercial, distributed systems such as on-line monitoring and diagnosis of telecommunication switching systems or realtime load control of power distribution networks.

To receive full advantage of the programming-free approach of the GenSAA WorkBench, the third component of the GenSAA WorkBench, the Data Manager, requires minor modifications to support drag-and-drop capabilities with the other WorkBench tools. The following section briefly describes the approach for customizing GenSAA to support other domains.

Integrating GenSAA Into Other Environments

Even without modifications, GenSAA will readily support the development of highly graphical rule based expert systems. However, in order to receive the "programming-free" benefit that this toolset provides, two steps must be taken: 1) the domain data must be formatted to allow the Data Manager to display it and thereby facilitate the drag-and-drop interoperability with the other WorkBench tools, and 2) the Data Interface Subsystem of the GenSAA Runtime Framework must be configured to manage the stream of the data selected with the Data Manager. There are basically two approaches for adapting the Runtime Framework for a new environment:

- Modify the GenSAA Data Interface – In this approach, the Data Interface subsystem of the GenSAA runtime environment is modified to accommodate the existing interface of the data source. The advantage to this approach is that the existing data source remains unchanged. However, the disadvantage is that the new user must modify unfamiliar code (GenSAA) and re-implement these modifications for any subsequent GenSAA releases.

- Create a custom Data Server– Perhaps a better approach for integrating GenSAA with the new data source is to create an intermediary process that functions as a TPOCC Data Server from the perspective of the GenSAA Data Interface. This process would receive all data requests from GenSAA and forward all data from the data source utilizing the standard TPOCC interface used by GenSAA. Several advantages would result: the group performing the integration does not have to modify foreign (GenSAA) code, updates to the GenSAA tool will not require re-implementation of the customized portions, and conformance to the original GenSAA Data Interface is maintained. The primary disadvantage is the performance penalty that may result from the extra processing in the intermediary process.

Although the modifications necessary to adapt GenSAA to a new environment may initially sound like too much effort, our experience has demonstrated that it is well worth the investment; if multiple expert systems are to be developed, the time spent customizing the front end of GenSAA is easily less than the effort that would otherwise be necessary to integrate each expert system with the corresponding data source. By employing object-oriented design techniques in GenSAA, the modification is simplified and isolated to specific objects thereby preventing the inadvertent corruption of other GenSAA elements that do not require modification.

Conclusion

Detecting satellite anomalies is a challenging task that is becoming more difficult as spacecraft become more complex, the number of sensor points multiplies, and data rates increase. As demonstrated by the CLEAR System, fault-isolation expert systems can help human analysts monitor the flood of data. Expert systems can accurately monitor hundreds of real-time telemetry parameters and isolate discrepancies and anomalies the instant they can be detected. They can alert the analysts while providing advice on how to correct problems swiftly and effectively. Unfortunately, development of these systems is often time consuming and costly; moreover, they usually cannot be 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. In the commercial industry, similar benefits can be realized with expert systems, and, although GenSAA was originally developed to assist with spacecraft monitoring, it naturally supports the rapid development and deployment of graphical intelligent monitoring systems in a wide range industrial applications.

References

1. Bielawski, L., & Lewand, R. (1991). Intelligent Systems Design: Integrating Expert Systems, Hypermedia, and Database Technologies. New York, New York: John Wiley & Sons.
2. Hughes, P.M. (1989, October). Integrating Expert Systems into an Operational Environment. AIAA Computers in Aerospace VII Conference, Monterey, California.
3. Hughes, P.M., & Luczak, E. (1991, October). GenSAA: Advancing Spacecraft Monitoring with Expert Systems. The American Institute of Aeronautics and Astronautics Computers in Aerospace VIII Conference, Baltimore, Maryland.
4. Hughes, P.M., & Luczak, E. (1991, May). The Generic Spacecraft Analyst Assistant (GenSAA): A Tool for Automating Spacecraft Monitoring with Expert Systems. 1991 Goddard Conference on Space Applications of Artificial Intelligence, Greenbelt, Maryland.
5. 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.
6. Perkins, D., & Truszkowski, W. (1990, September). Launching AI in NASA Ground Systems. AIAA/ NASA Second International Symposium on Space Information Systems, Pasadena, California.
7. Wall, S.D., & Ledbetter, K.W. (1991). Design of Mission Operations Systems for Scientific Remote Sensing. Taylor & Francis, Inc., Bristol, Pennsylvania.