GenSAA: A Tool For Advancing Satellite Monitoring With Graphical Expert Systems

Peter M. Hughes
Software and Automation Systems Branch (Code 522)
Mission Operations and Data Systems Directorate
NASA/Goddard Space Flight Center
Greenbelt, Maryland 20771

N94-23945

Edward C. Luczak
Computer Sciences Corporation
4600 Powder Mill Road
Beltsville, Maryland 20705

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 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 have been developed to support data monitoring and fault detection tasks in satellite control centers. 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 support the development of highly graphical expert systems for data monitoring purposes throughout the space and commercial industry.

Key Words: expert systems, mission control, X-windows, automation, real-time, data monitoring.

1. Introduction

NASA's Earth-orbiting scientific satellites are becoming increasingly sophisticated. They are operated by highly trained personnel called Flight Operations Analysts (FOAs) in the satellite's control center. The FOA's are responsible for the proper command, control, health, and safety of the satellite [Ref. 3].

Goddard's 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,
- manage spacecraft resources (e.g., on-board memory, power, 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 Payload Operations Control Center (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, and the approach for adapting this tool to other domains.

2. The CLEAR Expert 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 [Ref. 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. 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 [Ref.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 periodically 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.

2.1 Lessons Learned from CLEAR

Several important lessons have been learned from the experience gained in developing and operating CLEAR [Ref. 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) [Ref. 5]. The following is a subset of the lessons learned from these projects that have strongly motivated and influenced the development of GenSAA. [See Reference 2 for a complete listing and detailed description of each.]

• *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 because the if-then structure closely parallels the stimulus-response behavior that they develop through extensive training.

• *Allow the domain expert to participate in the rule formation.* The participation of the domain expert in knowledge acquisition and translation has many advantages: it can reduce knowledge translation time and errors; facilitate knowledge verification and validation; and improve user confidence in the system thereby better ensuring acceptance.

• *Expect to maintain/fine-tune the expert system after it becomes operational.* Although the operational characteristics of a spacecraft can be fairly accurately predicted, actual behavior can vary. The use of rule constructs facilitates changes necessary to refine the system quickly and easily.

• *Don't underestimate the effort to necessary to integrate the expert system with the data acquisition and user-interface elements.*

• *Provide an inviting, intuitive, and user-configurable user-interface.* The human-computer interface is frequently the most underdeveloped component of an expert system. Often, it can be the factor that determines how well the system will be accepted and trusted by the end-user.

• *Capitalize on the expressive power of graphics.* 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.

• *Simplify the operation of the system as much as possible by:*
  - *Reducing user input to a minimum.*
  - *Using hypergraphic techniques to allow quick and intuitive access to information and navigation through the system.*

3. GenSAA

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. These expert systems will assist the flight operations team in spacecraft control centers.

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 for many new spacecraft missions at GSFC. GenSAA will also be adaptable to 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 for TPOCC-based control centers supporting missions in the Small Explorer (SMEX) and International Solar-Terrestrial Physics (ISTP) programs. The initial use of GenSAA is targeted for the SAMPEX, FAST and the WIND/POLAR missions.

3.1 GenSAA Architecture

The GenSAA architecture is based on distributed processing and a variety of emerging industry standards. 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. It is designed to operate in a TPOCC network of computers which consists of a small set of specialized front-end processors and Unix-based workstations on an Ethernet network using the TCP/IP network protocol.

3.2 The GenSAA Workbench

The GenSAA Workbench is composed of three utilities that enable a spacecraft analyst to create a GenSAA expert system which performs real-time monitoring and 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 1 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. Values are assigned to these variables by

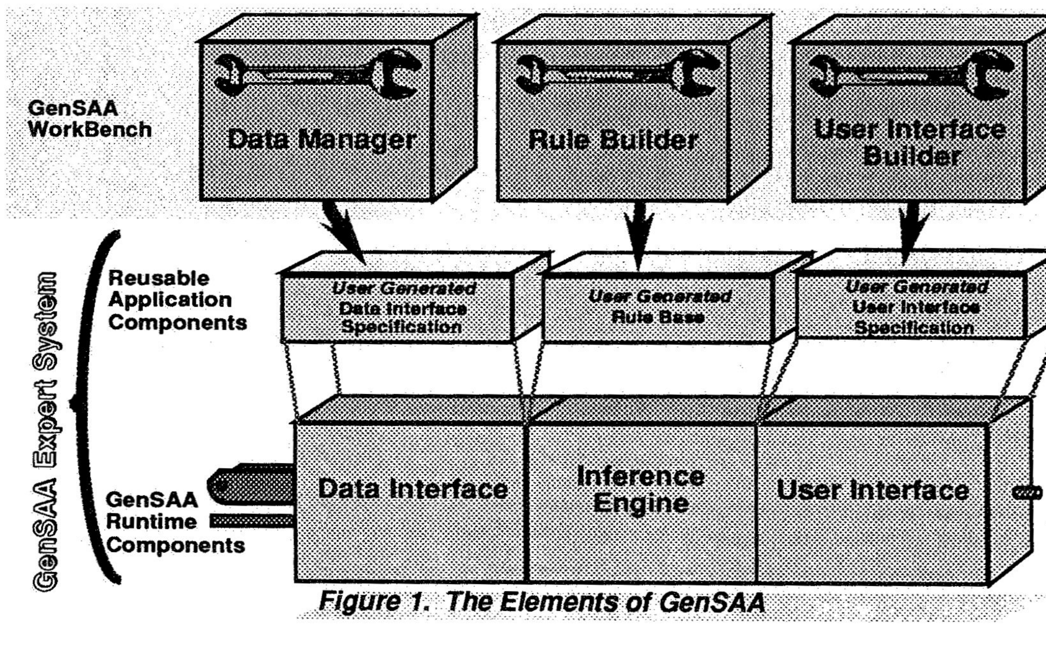

Figure 1. The Elements of GenSAA

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 available from other GenSAA expert systems. Conversely, a developer of a GenSAA expert system can identify Inferred and User-Defined data as "public" causing them to be routed to the GenSAA Data Server which distributes them to any process requesting them. 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 manages the matching and 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 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, gauges, and miniture stripcharts. The designer constructs a user interface by selecting graphical objects from a palette or drawing them with the graphics tools provided and positioning them where desired. Dynamic lines can be drawn 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 one or more windows to be displayed.

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.

GenSAA expert system 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.

3.3 GenSAA Runtime Framework

The GenSAA Runtime Framework provides the basic operational environment for a GenSAA expert system. The elements of the GenSAA Runtime Framework 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 operation of the expert system's user interface. It handles the user input, manages the display of windows (that may contain both text and graphics) and updates the visual attributes of the graphical objects. 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 mission data from the TPOCC Data Server, as specified in the Data Interface Specification. It formats the real-time data it receives and distributes it 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 mission, user-defined, EGG, and inferred data variables. Inferred facts and messages may be sent to the User Interface component and displayed to the user as defined in the User Interface Specification. NASA's 'C' Language Integrated Production System (CLIPS) inference engine forms the core of this component.

The user interface of a GenSAA expert system 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 hypergraphic links to make it easy for the spacecraft analyst to quickly display graphics windows.

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 connected to the TPOCC Local Area Network.

During operation, GenSAA expert systems 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 mission data that was specified in the expert system. No additional data or commands are sent to any of the TPOCC processes.

3.4 Implementation

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

3.5 Cooperating 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 onboard 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 Framework, 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 2, the GenSAA Data Server is a Unix process that can receive a real-time stream of "public" 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 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

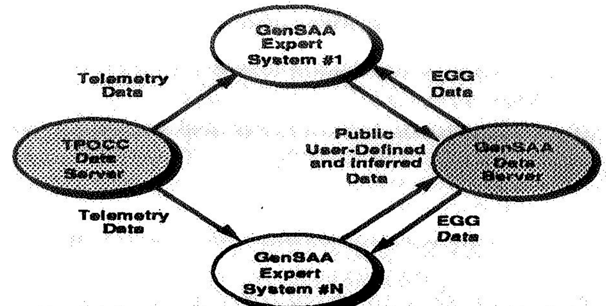

Fig 2. Sharing data among multiple GenSAA E.S.'s

mission 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.

3.6 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, 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.

4. 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 descriptions 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 Framework 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 seem to require considerable effort, our experience indicates that it may well be worth the investment. If multiple expert systems are to be developed in a given monitoring environment, the effort rquired to adapt the GenSAA data interface to this environment will likely be less than the effort required to integrate each expert system individually with its corresponding data source. The object-oriented design techniques used in GenSAA will simplify modifications and isolate them to specific objects in the GenSAA software.

5. 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 sector, similar benefits can be realized with expert systems. 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.

6. 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.