

ISTAR: Intelligent System for Telemetry Analysis in Real-time

Charles Simmons
Computer Systems Division
The Aerospace Corporation
El Segundo, CA 90245
simmons@arcibo.aero.org

ABSTRACT

The Intelligent System for Telemetry Analysis in Real-time (ISTAR) is an advanced vehicle monitoring environment incorporating expert systems, analysis tools, and on-line hypermedia documentation. The system was developed for the Air Force Space and Missile Systems Center (SMC) in Los Angeles, California, in support of the Inertial Upper Stage (IUS) booster vehicle. Over a five year period the system progressed from rapid prototype to operational system. ISTAR has been used to support five IUS missions, and countless mission simulations. There were a significant number of lessons learned with respect to integrating an expert system capability into an existing ground system.

INTRODUCTION

There are several trends taking place in the world of defense spacecraft that are making the mission support task increasingly difficult. First, defense spacecraft are becoming increasingly complex. With increased complexity comes an increase in the amount of health and status information that is sent to the ground for monitoring. It is currently not uncommon for a booster or satellite to downlink many thousands of telemetry measurands in a fraction of a second. These massive amounts of data must be quickly and reliably interpreted on the ground by mission control. Teams of controllers must detect, diagnose, and respond to anomalies in the vehicle. This task may be subject to rigid time constraints, especially with short duration booster vehicle missions. Current methods used by ground controllers for performing detection and diagnosis of anomalies are primarily manual and therefore slow and unreliable.

Second, there are an increasing number of defense space vehicles and booster launches to support. Launch frequency will increase in the future to support new satellite systems and to increase constellations for existing systems. The workload placed upon mission support will dramatically increase to accommodate this trend.

Third, there is a diminishing supply of qualified mission controllers. Typically, controllers with the greatest amount of experience and expertise are also the ones closest to retirement age. As they retire, they are often replaced with inexperienced controllers who may lack the capability to deal with complex problems.

Finally, there is the trend towards reduced staffing, and less contractor support. Due to cutbacks in the defense budgets, government spacecraft programs are being forced to make do with static or decreased budgets. This usually translates into hiring freezes or layoffs. The result is that there are fewer personnel available for mission support. Many mission support duties currently performed by vehicle contractors will be assigned to military personnel with considerably less experience and training.

It is clear that with respect to mission support, defense related spacecraft programs cannot afford to operate as they have in the past. Several government space organizations have realized this and have begun to look to advanced technologies for solutions [1,2,3].

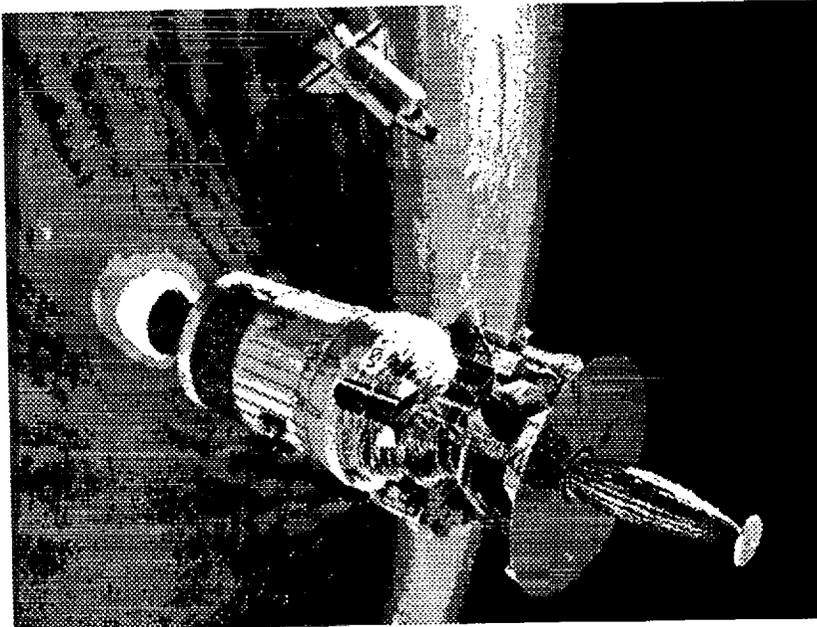


Figure 1: Inertial Upper Stage (IUS) booster deployed from Shuttle with Galileo Probe

IUS MISSION SUPPORT

The Inertial Upper Stage (IUS) vehicle presents a typical example of the problems outlined above. The IUS is a multi-stage booster (see figure 1) that supports delivery of civilian and defense satellites from low earth orbit to higher energy earth or interplanetary orbits. It is deployed from either the Shuttle or a Titan rocket. With many thousands of interdependent and redundant components, the IUS vehicle is highly complex. Ground controllers at the Air Force Consolidated Space Test Center in Sunnyvale, California are responsible for IUS mission control. They are required to monitor and reliably interpret over 1300 telemetry measurements to determine the health status of the vehicle. There are several significant problems that currently face the IUS mission control team in performing this task.

One problem relates to anomaly detection methods. Detection of anomalous conditions is primarily done by recognizing combinations and trends of telemetry measurements displayed on computer screens. Because this task is largely one of visual detection, the

possibility of missing or incorrectly interpreting a significant event is substantial. The potential for controller fatigue and overload makes this problem even worse. An IUS mission controller's shift can last anywhere from 8 to 18 hours. At the end of such lengthy shifts, fatigue and boredom levels are high, and the potential for operator mistakes is great. Also, there is the question of how many telemetry measurements a controller can adequately monitor. If the controller is asked to monitor too much data, information overload can significantly handicap his or her ability to recognize anomalous conditions. It is very clear that the IUS mission controllers needed a tool that could assist them in their vital task of vehicle monitoring.

Another problem relates to the time critical nature of IUS missions. The typical IUS mission is quite short relative to most spacecraft missions. The time from deployment to spacecraft separation is less than seven hours. If problems develop after deployment, there is a limited time window in which to respond. There is significant pressure placed upon mission support to quickly assess anomalies and recommend some action. In this "heat of battle" situation, the potential for mistakes by the controller is at its greatest. IUS mission controllers needed a tool that would assist them in making correct decisions in time critical situations.

Loss of expertise is a major problem currently facing the IUS program. To perform the tasks of a mission controller, the person needs to possess a significant level of monitoring "expertise". The controller must know what measurements to view at a given phase of the mission, and how to interpret their values. This expertise is based largely on experience and detailed knowledge of the internal operation of the IUS. Many of the current controllers have been with the program from its inception and have developed a tremendous amount of experience and knowledge of the vehicle. However, many of these controllers are at or near retirement age, or are transferring to other jobs. Loss of just a few valuable controllers could have a substantial impact on the ability of the mission control team to adequately support a mission. IUS mission control needed some way to retain portions of the expertise of these controllers for training new controllers, and for utilization during future missions.

ISTAR SOLUTION

In response to the problems facing IUS mission control, the ISTAR automated telemetry monitoring prototype system based on expert system, graphical user interface, and hypermedia technologies was commissioned [4,5,6,7]. The prototype was developed by the Expert Systems Section of the Aerospace Corporation, a federally funded research and development contractor. It was determined that, in addition to automated fault detection, the system should provide a variety of tools for assisting the mission controller in performing normal duties. In essence, the system was not to attempt to "replace" the controller, but rather to "assist" the controller in the task of mission monitoring. It was determined that the system should be able to receive telemetry in real-time or playback, detect anomalous IUS conditions, and interact with the user to isolate and diagnose the conditions. The system should provide a graphical user interface, allow replay and graphic presentation of past telemetry, provide diagrams representing the current state of the vehicle, provide on-line documentation pertinent to mission support, and operate on existing ground workstations.

The ISTAR system was designed to be operational with the Spacecraft Monitoring And Real-time Telemetry (SMART) system within MCC-8 of the Air Force Consolidated Space Test Center (CSTC). SMART is a real-time data acquisition and analysis system based on the System-90 Product by Loral Data Systems. It distributes its processing load among three types of equipment: telemetry front end equipment (TFE), a Host processor (DEC VAX 4000), and a network of workstations. Figure 2 shows the basic architecture for the SMART system. Telemetry is processed by the TFE and Host processor, stored on the Host disk, and broadcast to the workstations over an ethernet network, in the form of a current value table.

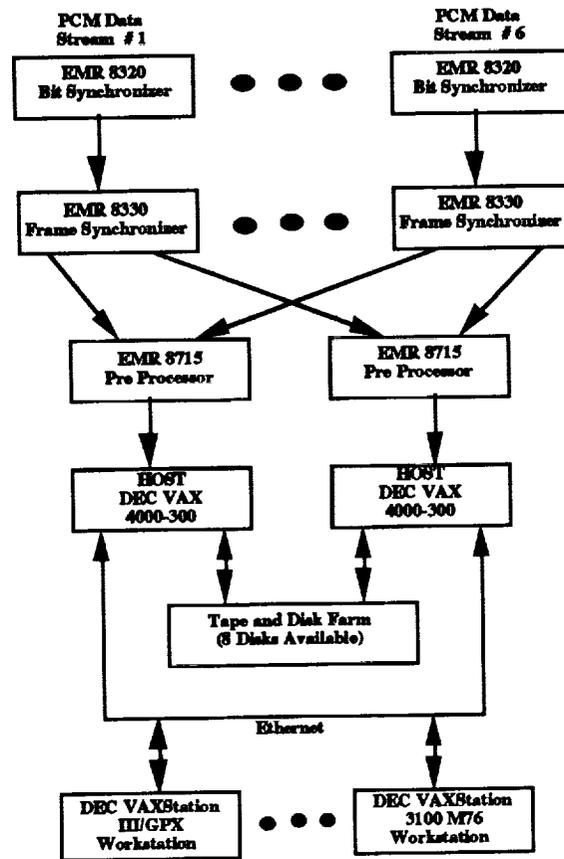


Figure 2: SMART System Architecture

The ISTAR system was implemented to run on existing workstations within the SMART system, which are Digital Equipment Corporation VAXstation IIIs and 3100s, running the VMS operating system. The ISTAR system utilizes a distributed architecture consisting of components for performing display, analysis, retrieval, and archive of IUS telemetry data. These *processes* communicate data and information by way of dedicated communication links (VMS mailboxes). Processes for display, analysis, and on-line documentation use Commercial Off The Shelf (COTS) products. The system can be operated in either real-time or playback mode. Figure 3 shows a functional diagram of the ISTAR architecture.

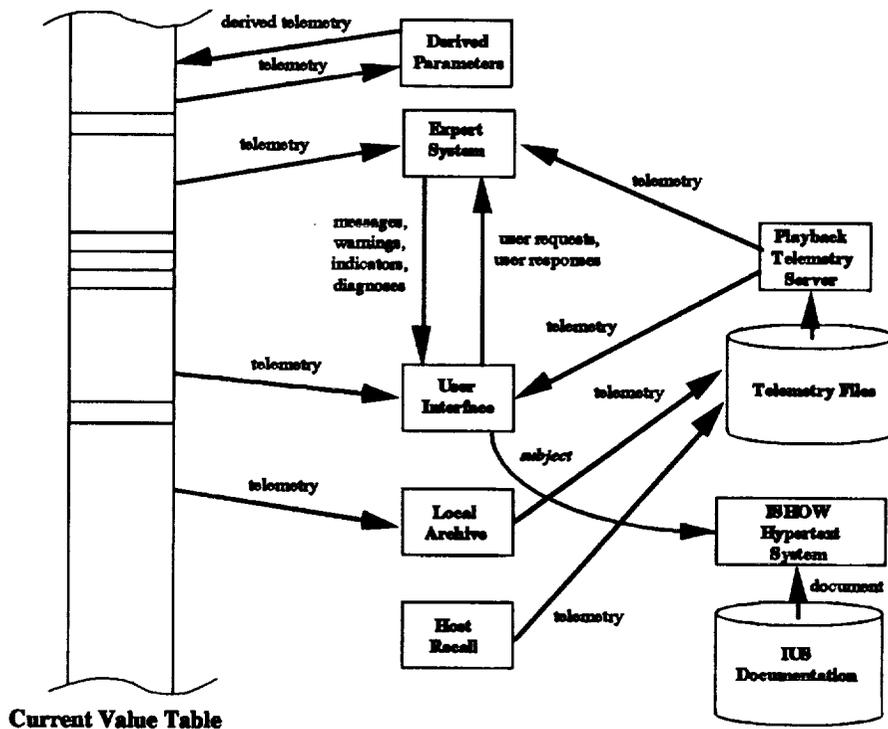


Figure 3: ISTAR Architecture & Data Flow

In the *real-time* mode, the Expert System Process performs analysis of dynamic telemetry data received from current value table (the current value table (CVT) contains one sample of every telemetry measurand) broadcasts from the SMART Host computer. It presents status information, and detects and diagnoses anomalous conditions. It utilizes the NEXPERT expert system shell developed by Neuron Data Corporation, and custom software. The User Interface Process is used to provide the user with status messages, warnings, diagnoses, and advice from the expert system, as well as animated schematics, telemetry graphs, and telemetry retrieval and manipulation tools. It consists of the Dataviews graphics development system developed by V.I. Corporation, and custom software. The Derived Parameter Process reads values from the CVT, performs calculations on the values, and inserts the resulting data back into the CVT. This derived data is then available to other processes through the CVT. The Local Archive Process samples specified telemetry data from the CVT and stores the data to telemetry data files. The Host Recall Process acquires specified telemetry data from the SMART Host computer and stores the data to telemetry data files. The resulting telemetry data files are available for playback into the ISTAR system. The IUS System for Hypertext on Workstations (ISHOW) documentation system displays IUS Orbital Operations Handbooks and other mission support documents. It consists of the Farview Hypertext System developed by Farsight Technologies Inc., and custom software. The User Interface process can request display of specific topics or sections from the ISHOW documents.

In the *playback* mode, instead of acquiring telemetry from the CVT, the expert and user interface processes receive telemetry from the Playback Telemetry Server Process, which reads a specified telemetry data file.

The ISTAR system was designed to operate concurrently on the same workstation with existing SMART software. The SMART workstation software's primary role is to provide graphical and alphanumeric displays of telemetry to the end user. It provides a large number of pre-configured displays for viewing various aspects of the IUS mission. During normal operation, the ISTAR system functions in the background. If an anomalous condition is detected, the operator is alerted and ISTAR is brought to the foreground, allowing further investigation of the problem.

In addition to an expert system, the ISTAR system provides an integrated set of tools for analysis of the vehicle. These tools include the following:

Graphing Tools - Graphs can be created on-the-fly showing either real-time or history graphs of telemetry. Support is provided for both analog and discrete telemetry measurements, and multiple graphs per screen.

Local Telemetry Archive - As the telemetry CVT is broadcast to the workstations, specific measurands can be stored to a file at the local workstation in real-time, for later replay into the ISTAR system. The CVT broadcasts, and thus the local archive, are limited to a once per second update rate.

Host Telemetry Recall - Specific telemetry data can be retrieved from the SMART System Host computer and stored to a file at the local workstation. The SMART disk farm typically contains telemetry from current and past IUS missions. Telemetry from the Host recall contains every sample of the requested measurand over the specified period.

Data File Manipulation Tools - Once telemetry is written to the local disk by the Host Telemetry Recall or Local Archive, a set of file manipulation tools can be used to search for events or combination of events, merge and overlay telemetry from different files, and edit parameter names and start/stop times. Using these tools allows, for example, construction of a file combining measurements from the current mission with those from a previous mission.

Expert Message Log - The message log provides a database of all messages received from the expert system. Associated with selected anomaly messages is a *diagnostic help option* that instantly configures ISTAR tools to investigate and justify the identified anomaly. By clicking on an icon next to the message, the user can rapidly view telemetry and information related to the specific anomaly.

On-line Hypermedia Vehicle Documentation - Several important mission support manuals have been placed on-line. These manuals are *hyper-linked* to allow rapid navigation to access needed information. The expert system may also bring up pertinent sections of these manuals in response to a detected anomaly.

KNOWLEDGE BASES

Knowledge bases are invoked by the user as needed from a menu of knowledge bases. Knowledge bases can be driven with real-time telemetry, or with past telemetry using the ISTAR playback utility. Knowledge bases may generate status, warning, anomaly

detection, or anomaly diagnosis messages. Anomaly detection and diagnosis messages may contain an anomaly explanation, justification, recommended action, hints on further investigation, and automatic configuration and branching to ISTAR tools such as graphs, schematic diagrams, and relevant sections of on-line manuals, that are pertinent to the anomaly investigation. The knowledge bases in ISTAR are of several types:

Status Knowledge Bases - These knowledge bases provide general vehicle status information to the user. For example, knowledge bases were developed that generate status messages based on vehicle events, and significant changes or limit violations of voltage, current, temperature, and pressure telemetry.

Monitoring Of Specific Mission Phases - Specific knowledge bases were developed for monitoring various phases of the IUS mission. For example, knowledge bases for monitoring deployment, stage separation, solid rocket motor burns, and separation were developed. These knowledge bases monitor specific events and attempt to detect and diagnose based on anomalous conditions that are most likely to occur, or most critical to detect. They are based largely on the heuristic knowledge acquired from vehicle experts.

Monitoring Of Specific Subsystems - Specific knowledge bases were developed to monitor subsystems or components of IUS. For example, knowledge bases to monitor the Power Distribution Unit subsystem, and the Signal Conditioner Unit Multiplexer component were developed. These knowledge bases typically run the entire duration of a mission, and are configured to detect predetermined anomalous conditions that are most likely to occur, or most critical to detect. They are based largely on the heuristic knowledge acquired from vehicle experts.

Automaton Of Handbook Procedures - Knowledge bases were developed which implement procedures from the IUS orbital operations handbooks. These procedures include uplink command sequences to accomplish various vehicle configurations, and redundancy restoration. For example, a procedure is available for restoring redundancy to an avionics channel after it has been lost due to an anomaly. These procedures typically monitor uplink commands to verify correct execution.

ISTAR DEPLOYMENT

The ISTAR project was initiated in May of 1988, by the Air Force Space and Missile Systems Center IUS Office, to be performed by The Aerospace Corporation. A rapid prototype was developed on an IBM PC in four months as proof of concept. The prototype utilized a simple COTS expert system development tool and was driven with simulated telemetry. A rapid prototype was then developed on the target machine, a DEC VAXstation III workstation. This prototype was completed in six months, and used a more sophisticated expert system development tool. In May of 1990, an operational prototype was completed and installed at the Air Force Consolidated Space Test Center (CSTC) for evaluation during simulations.

Prior to evaluation, ISTAR underwent extensive testing at CSTC to insure that it would reliably operate on current SMART workstations, yield results consistent with SMART, and function without adversely affecting the operation or performance of the workstation, the SMART Host Computer, or the SMART network. The groundrule for use of the

system was that no expert system results would be used without verification from an official Air Force telemetry data system. After completion of several simulations, it became apparent that the ISTAR system provided valuable, timely information to the support team. The excellent performance of ISTAR generated support from management at Air Force SMC, Aerospace, and the IUS prime contractor (Boeing Aerospace & Engineering) for its continued development and use.

Over the next three years, the system was used at CSTC to support five IUS missions, and countless mission simulations. During this time, the system often played a key role in detecting or verifying the presence or absence of anomalies. It was even used as a primary tool in the investigation of an inertial measurement unit anomaly, which led to a launch abort. Enough confidence was placed in the system to allow it to assist in determining a go/no-go launch decision. Due to the fact that it was a prototype, ISTAR was allowed to continually evolve based on lessons learned during operation. This resulted in a much more effective system than would have otherwise been possible.

LESSONS LEARNED

The ISTAR system represented the first significant attempt to acquire an expert system capability for the Air Force SMC Space Launch Operations Program. Over the five years of development and operation, many lessons were learned. The following outlines the most significant lessons learned (in no particular order).

Need champions at the top.

A significant lesson learned was that it is very important to have the support of upper management. The new technology had to be sold to management both initially and continually. Through presentations and demonstrations, expert systems combined with advanced graphics were shown as a way to address the current problems of reduced staffing, and more reliable anomaly detection and diagnosis. Frequent demonstrations were given to management over the development period showing progress; this kept them interested in the project. Unwavering management support allowed continuous funding over the five year development period, and made insertion of the prototype into the operational environment possible.

Need a user who is a champion of the new technology.

In addition to champions at the top, a champion in the field is also needed; that is, a member of the user community who understands and can sell the new technology to other users. Such a person makes technology infusion a lot easier. A vehicle expert was available who possessed a global knowledge of the vehicle, anomaly detection and diagnosis, and who had enthusiasm for developing a support system. The expert served as the primary source for initial knowledge base development. Perhaps of most value was his ability to generate enthusiasm for the new capability among IUS management, and other mission control team members, and to provide direct use and evaluation of the system during mission support.

System must be of maximum benefit to users.

To gain acceptance by IUS mission controllers, the ISTAR system had to provide capabilities that addressed their needs. To build just an expert system would have addressed one particular need, but perhaps not their strongest need. Therefore, the ISTAR system was designed with the philosophy of providing an integrated set of tools, expert systems being one of those tools. Other important tools included graphical subsystem browsing, real-time and historical graphing, local data archiving, Host data recall, data file search & manipulation, and on-line vehicle documentation. This proved to be a wise decision. After the users decided they liked the system, it was easy to add more expert system capability.

There are benefits to making the system transparent to the operational environment.

There was much resistance to allowing ISTAR into the operational environment. There was fear that system errors might negatively impact the existing operations. Because of these fears, strict configuration management, and political considerations, it was decided that no dedicated hardware would be allowed into the operational environment for the ISTAR effort. In addition, no operational code could be modified to accommodate ISTAR. Thus, the only way to gain access was to run transparently on existing workstations. Fortunately, ISTAR was designed to do just that. No software or hardware changes to the SMART workstation or Host were required, just a simple change to the system startup file. Once installed, the user was given the option to run the ISTAR system alone, the SMART system alone, or both concurrently. The fact that the ISTAR system could simply be switched off if not desired, without any impact to SMART capability, was key to gaining acceptance. Moral: If you can insert the new technology in a transparent, non-threatening way, it is best to do so.

Start with simple, rapid prototypes, and proof of concepts.

A rapid proof of concept prototype proved to be a valuable exercise. Within four months, a prototype was built showing a solution of a very limited subset of the entire problem. These four months proved to be a great learning experience for the customer as well as the developers. The developers learned that there were going to be difficult problems to overcome, such as data and knowledge acquisition. The customer learned that what they thought they wanted in the beginning was not exactly what they wanted or needed. The result of the prototyping effort was a new perspective on program needs, and problems to be solved. It helped immeasurably in defining requirements for the next phase.

Make sure all users are involved from the start.

It was learned that getting all potential users involved in the beginning is key to gaining user acceptance. In the initial development phases, only Aerospace Corporation mission controllers were involved in the system definition. During later development phases, the IUS prime contractor Boeing Aerospace & Engineering was funded to participate in the project. After only a short period of time, Boeing engineers made important recommendations on additional system requirements, which resulted in greater user acceptance.

Most effort spent in developing and integrating core system.

About two-thirds of the entire ISTAR project effort was spent in developing the core system. Less than one-third of the effort was spent on knowledge base development. ISTAR development tasks included designing the basic architecture (10%), interfacing the core architecture with the existing environment (25%), interfacing the core system with COTS products (25%), and development of special functions and tools to augment the capabilities of the various COTS products (40%). After the core system was built, the knowledge base development effort was increased. It should be noted that COTS products are now available with many of the features that had to be custom developed for ISTAR.

Need strong software developers.

Even though COTS products were used, custom software had to be developed to interface the COTS products to data sources and to each other. In addition, software had to be written to augment the built-in capabilities of the COTS products. The system ended up with over 30,000 lines of source code, excluding COTS products. The software was developed by programmers with many years of experience in the target language, operating system, and target machine architecture. In general, timely, reliable software products were created. Due to the many subtle coding and integration challenges, schedule impacts would have undoubtedly occurred without experienced programmers. It should be noted that COTS products are now available with many of the features that had to be custom developed for ISTAR. If a similar effort were started today, much less custom software development would be necessary.

Use COTS products whenever possible.

One significant lesson learned was the advantages of using COTS products over custom development. First, support costs are lower for COTS products than for custom software because the maintenance costs are shared by all users. Second, COTS products are more adaptable to changing requirements over a project's life span due to support of multiple platforms, multiple applications, and multiple users. Third, the features of a COTS product are often thoroughly tested by a large community of users. The advantages of COTS over custom could be clearly seen in experiences with the SMART system, which was a totally custom software development. After DEC announced that they would no longer support the VWS user interface, it was desired to move SMART to the new DECWindows interface, based on the X-Windows standard. However, due to the large amounts of custom code developed around VWS, and low-level calls into the graphics hardware, such a transition would have required virtually a total rewrite of the SMART System-90 display software. However, such a transition would have been virtually automatic for ISTAR, which relies heavily on COTS, since the COTS vendors had already worked out the port to DECWindows.

Choose COTS products carefully.

It is important to choose the right COTS products. Since the time that COTS products were chosen for ISTAR, many similar products have come and gone. Had the system been built around a failed product, it would have been a disaster for the project. It was also learned that the COTS product must be able to handle current as well as unanticipated future requirements. It often makes sense to choose a product that is more

robust than needed. As requirements change, the system may eventually grow into those capabilities. It is very important that the COTS product provide an open design to allow custom extensions if needed. Also, financial stability of the vendor should be a consideration. New and better COTS products are being offered at an ever increasing rate. In fact, if a COTS decision were made today for this effort, totally different packages would be selected. Therefore, it is important to keep the overall system architecture generic enough to allow upgrade to a different COTS product when necessary, without extensive redevelopment.

Start with simple, well-defined knowledge bases first.

The first knowledge bases developed for ISTAR were for vehicle redundancy management anomaly detection, diagnosis, and resolution. These knowledge bases were rather complicated, and usually required user input of certain information before diagnoses and restoration could be made. While these knowledge bases worked well with simulated cases, they represented situations that were very unlikely to occur during missions. A decision was then made to develop broader, more simplistic knowledge bases with emphasis on vehicle anomaly state determination and simple anomaly detection. These knowledge bases proved to be valuable assets for mission controllers in detecting vehicle developments that might have otherwise been missed -- one of the major goals of the system. After early successes, these simple knowledge bases were enhanced, and more sophisticated knowledge bases were attempted. It was also learned that anomaly isolation and diagnosis are much more difficult than detection. Automated diagnosis would be nearly impossible in many failure situations. In these situations, it was best to give the users pertinent information that would assist them in performing their own anomaly isolation and diagnosis (such as relevant graphs, diagrams, and vehicle documentation).

Automate diagnostic information in spacecraft operations manuals.

Knowledge acquisition from a human expert is clearly a difficult task, and the major bottleneck in developing an expert system. However, much of the knowledge necessary to develop simple knowledge bases may be contained in vehicle operations handbooks. Several IUS handbooks were identified that provided valuable information for ISTAR knowledge base development.

Make sure the expert can read and understand the knowledge base.

For ISTAR, the knowledge acquisition process was typically an iterative task performed by an expert and a knowledge engineer. The knowledge engineer interviewed and/or obtained written requirements from the expert, generated an initial knowledge base, reviewed the knowledge base content and performance with the expert, and made agreed upon changes. It was very helpful to have an expert who could understand the basic concepts of a knowledge base, and the syntax of the rules. After only a short period of time, knowledge bases were given to the expert for visual verification. Eventually, the expert was able to make simple changes to knowledge bases. This accelerated the knowledge base development effort. It also helps to have an expert system product that presents rules in a readable, English-like syntax.

Tackle problems that are best suited for expert systems.

One of the lessons learned was that not every problem is suitable for an expert system. It worked out best to tackle problems that were declarative in nature, rather than procedural. Detecting combinations of vehicle events that indicate a problem was a very appropriate use of the expert system tool. However, trying to implement procedures in a rule based expert system proved to be difficult. If procedures are to be implemented, it is best to use an expert system shell with procedural capability. In general, the approach taken here was to try to mimic the declarative techniques that the human expert used in monitoring and detecting anomalies.

Develop a reasonable verification and validation standard.

Validation of expert systems is always difficult, and sometimes impossible. The approach taken by the NASA RTDS project was adopted here [1]. Their approach was to validate expert systems based on use. In order to qualify for mission use, a knowledge base had to work flawlessly in a specified number of mission simulations. Our adoption of this rule was that no new knowledge bases, or any other ISTAR capability, could be used during a mission without correct performance in at least two mission simulations and one mission dress rehearsal.

Common user interfaces needed.

The screen layout and functionality of ISTAR interface are much different than that of the SMART system. Therefore, users of SMART had to be trained on how to use the ISTAR interface. A common screen layout and functionality for ISTAR and SMART would have resulted in reduced training requirements and increased user acceptance. Future adherence to emerging standards for spacecraft command and control user interface design will address this problem [8].

Choose a powerful workstation.

One lesson that was quickly learned was not to underestimate the amount of computing power needed for the application. As more COTS products, knowledge bases, and system features were added to the ISTAR prototype, the computing capacity of the workstation was quickly reached. Midway through the development it was necessary and permitted to upgrade to a more powerful hardware platform that maintained compatibility with the SMART system.

Deal with real-time data and control issues early.

One of the technical challenges was determining how to connect to and process the real-time data stream. The problem was more difficult than was originally anticipated. However, it was important to establish this connectivity in order to demonstrate the capabilities of the system in the operational environment. Demonstrating successful real-time processing of live vehicle data greatly increased the credibility of the system.

It is important to have support for derived measurands.

Capabilities for derived telemetry measurements proved to be important for a number of reasons. There were many situations where it was desired to have telemetry parameters

combined in some predetermined fashion. For example, taking a voltage and a current to produce a wattage. The result of such calculations was often needed by the expert system, as well as the user interface. Rather than burden the expert system with the overhead of such calculations, a separate derived process was created. This process allowed derived parameters values to be written to the SMART current value table. The derived parameters could then be displayed by the interface process, incorporated by the expert system process, and even archived by the local data archive process.

It is important to have local telemetry archiving and playback capability.

Local telemetry archiving at the user workstation proved to be a very valuable capability. The prime advantage was that once the telemetry was archived to the workstation disk, it could be accessed freely and rapidly. There were several situations where the network had gone down during mission operations. Those workstations relying on SMART network distribution of playback data were unable to perform analysis during the down time. However, users of ISTAR were able to analyze locally archived telemetry during this period. Also, this analysis could be done at a speed superior than SMART network access would have allowed.

It is desirable to have an integrated on-line documentation system.

ISTAR provides on-line hypermedia access to important IUS mission support documentation. The hypermedia links allow rapid access to needed information in text and graphical form, by mouse-clicking. A valuable capability was the ability to perform inter-document referencing. In many cases, while browsing a document, the user can request and branch to information on a subject that is contained in a separate document. Another valuable capability is the ability of the expert system to recommend and allow branching to relevant sections of the documentation, based on the anomaly detected. The hypermedia capability helped the mission controller decrease reliance on paper documents, and improved access to critical information during IUS missions and simulations.

SUMMARY & PLANS

This paper has presented an overview and lessons learned from the ISTAR expert system prototype developed by The Aerospace Corporation for the Air Force Space and Missiles Systems Center, for support of IUS booster missions. In 1993, the ISTAR system was turned over to Boeing Aerospace & Engineering, the IUS prime contractor, for continued development and maintenance. The existing ISTAR system is currently being ported to the UNIX operating system and Sun workstations, to allow easier integration into the future IUS ground system, which will be based on the Loral Open System 90 telemetry system.

REFERENCES

1. Heindel, T.A., "Shells for Mission Control," Aerospace America, October, 1991

2. Atkinson, D.J., et. al., "SHARP: Automated Monitoring of Spacecraft Health and Status," Space Operations Applications and Research Conference Proceedings, June, 1990
3. Schwuttke, U.M., et. al., "MARVEL: A Knowledge-Based Productivity Enhancement Tool for Real-Time Multi-mission and Multi-Subsystem Spacecraft Operations Status," Jet Propulsion Laboratory Technical Report, June, 1991
4. Simmons, C.B., "Intelligent System for Telemetry Analysis in Real-time (ISTAR) Part I: Overview & Project Legacy," The Aerospace Corporation Technical Operating Report (3464)-4, September 1993
5. Simmons, C.B., "Intelligent System for Telemetry Analysis in Real-time (ISTAR) Part II: User's Guide," The Aerospace Corporation Technical Operating Report (3464)-4, September 1993
6. Simmons, C.B., "Intelligent System for Telemetry Analysis in Real-time (ISTAR) Part III: Programmer's Guide," The Aerospace Corporation Technical Operating Report (3464)-4, September 1993
7. Lee, J.S., Simmons, C.B., "Intelligent System for Telemetry Analysis in Real-time (ISTAR) Part IV: ISHOW Hypermedia Reference Manual," The Aerospace Corporation Technical Operating Report (3464)-4, September 1993
8. Air Force Space Command, *Integrated Satellite Control Human Computer Interface Standard Draft Version 1.0*, ISC HCI Working Group, August 1993