Jswitch/Jsat: Real-time and Offline World Wide Web Interface

Abigail H. Maury
Anna Critchfield
Jim Langston
Computer Sciences Corporation

Cynthia K. Adams
Goddard Space Flight Center

Abstract: The Java-based spacecraft web interface to telemetry and command handling, Jswitch, is a prototype, platform-independent user interface to a spacecraft command and control system that uses Java technology, readily available security software, standard World Wide Web protocols, and commercial off-the-shelf (COTS) products. The Java-based science analysis and trending tool, Jsat, is a major element in Jswitch. Jsat is an inexpensive, Web-based, information-on-demand science data trend analysis tool. Both Jswitch and Jsat were developed in Java for NASA's Goddard Space Flight Center (GSFC). Jsat provides Web interface user access to science data from different satellites. Historically, satellite science data trend analysis has been an additional time-consuming operational consideration in which logistics of the science analytical parameter requests/responses are cumbersome and restrictive. Jsat removes this middle layer of access by sending the processed science satellite data (user-specified graphics, tables, and reports) as Java applets via the Web to the user upon request. Jsat reduces ground science data processing costs and decreases processing time by eliminating the need for special operations personnel assigned to process science data requests, run trend analysis software, and collect and distribute results. Jswitch reduces operational costs by providing access to spacecraft telemetry and command from any standard web browser.

Introduction

NASA's Goddard Space Flight Center (GSFC) is funding the Jswitch/Jsat system as a research effort in the use of the latest technology to reduce satellite ground system operations costs. The result of this research is the Jswitch/Jsat system. This system enables a spacecraft engineer and flight operations engineer to monitor and control a spacecraft or enables a scientist to select and analyze the science data. This system also enables the user to communicate over the open Internet using the standard World Wide Web (WWW) tools, browsers, and commercial-off-the-shelf (COTS) products in conjunction with the Java programming technology. Security is provided by the current encryption and certification, firewall, and intrusion detection technology.

Jswitch interfaces with the Epoch 2000 system, which provides the core telemetry and command control functions. Jsat interfaces with the Archive Browser Extractor (ABE), a COTS spacecraft non-real-time telemetry data analysis package that provides the statistical subset data. Epoch 2000 and ABE were developed by Integral Systems, Inc. (ISI). ISI provided both products to NASA/GSFC for interfacing with and prototyping the Jswitch/Jsat system.

The Jswitch/Jsat system demonstrates how this combination of current technologies can be used in mission support systems today. The Jswitch system reduces operations costs through operational scenarios that enable remote diagnosis of problems detected during lights-out operations periods. The Jsat system removes the necessity for operators to provide data to scientists by giving scientists direct access through the Web. In addition, the system allows distributed operations teams to use the inexpensive open Internet for wide area access. These identical protocols can be used for internal closed networks.

The flexibility of the Jswitch/Jsat system provides for usability across many mission profiles, thus reducing development costs. The Jswitch/Jsat system fits into the typical GSFC control center that consists of three redundant operational strings to support primary, backup, and offline spacecraft health and safety, because the Jswitch/Jsat system can be run as one of these strings. The net result is the first opportunity to standardize user interfaces to missions, to provide flexible networking, and to reduce operations, significantly decreasing life-cycle costs.
Jswitch/Jsat Advantages

The Jswitch/Jsat work is based on a prior study and a small prototype that demonstrated using Java applets with the legacy Transportable Payload Operations Control Center (TPOCC) system. Figure 1 depicts the TCP/IP data connections between the Jswitch and the TPOCC system through the Web server. Jswitch added security and substituted a COTS command and control.

Because other research activities were being performed with most of the other ground system products and because it was already mission proven, the Jswitch team selected the Epoch 2000 product from ISI as the COTS command and control product. A team member from this vendor was included in the initial prototype version.

The Jswitch prototype Java software interfaces with the Epoch 2000 application programming interface (API) through C code. Using a distributed architecture, the control center software can be replaced relatively easily (TPOCC vs. Epoch 2000) while the Jswitch user interface stays the same.

The Jswitch/Jsat prototype gives the user the ability to view events and telemetry data, send commands, and analyze selected statistical science data—all through the use of a web browser as the user interface. From our experience, this approach offers a number of advantages:

1. The workload is distributed. Once the Java applets are downloaded to the client machine from the connected host, the applets run locally on the client's machine.
2. Portability. The Java applets will run on any machine that has a web browser that supports Java—without any required source code changes for a specific platform (one version for all).
3. Easy user access. There is no complicated setup involved or additional software to buy for clients—the only requirement is a web browser that supports Java and an Internet connection.
4. Easier software maintenance. Clients always connect through the central web server, which automatically downloads the applets, making it easier to install updates to the software.
5. Scientific community direct access to the data. Using Jswitch/Jsat security mechanism, the user privileges can be set up to allow direct access to the data for analysis without jeopardizing the command and control security.

Jswitch/Jsat Web Security

Security over the open Internet is recognized as the major risk in satellite operations, particularly in commanding the satellite. Jswitch employs a number of readily available security tools and procedures to provide a secure interface over the open Internet. The risks and mitigation associated with commanding are an ongoing topic for debate to provide a comfort level that balances the reduced cost of operations with a quantifiable risk.

Figure 1. Jswitch TCP/IP Connection to TPOCC

Abigail H. Maury
The only totally secure interface is to have no connection at all to the outside world. This would not provide the capability we are prototyping, which is to have a simple access capability from all over the world, via a common web browser, to specified ground system functions with practically no prior setup. With the utilization of several tools, many of them free, and a few simple setup procedures on the host system, a reasonably secure interface can be achieved at low cost. The gauntlet of security tools consists of:

- A web server that transmits information in encrypted format
- Secure/changing passwords to authenticate the user at the web browser
- A "digital id" to verify the authenticity of the web site
- Firewall software to restrict access to the host machine
- Intrusion detection software to automatically detect hacking attempts
- Simple scripts to check the integrity of the host system
- Probing software to look for security weaknesses on the host machine

These tools are also combined with specific system software alterations on the host machine to make the site more secure and less vulnerable to hacking attempts. The host machine itself, a typical Unix workstation, is designed to sit connected to an open Internet, with the ground system interface software and the actual ground system software on the same box. The only connection to actual data will be through a socket connection to a different device, which will likely be behind another firewall or other closed network. Identical or equivalent software and procedures are available for Windows NT servers.

The heart of the interface consists of the Stronghold web server, which is a Secure Socket Layer (SSL) implementation of the Apache web server, the most common web server in the world. Non-SSL web servers transmit information over the Internet in “clear text”, which is easy to read. Netscape’s SSL, a de facto standard now, encrypts all transmissions, making it difficult for the data sent between the user’s web browser and the web site to be monitored in transit. The encryption strength used depends on the web browser version installed by the user. Most web browsers have 128-bit encryption available (although these are restricted for use only in the United States). Browsers that are for export outside the United States use 40 bits for encryption, which is reasonably secure. The web server’s function in the prototype is to serve Java applets to the user. These Java applets provide for a graphical dynamic interface to the ground system. They connect to data servers to send/receive requests and data to interface programs that talk to the actual ground system. The Java applets, once downloaded on the client’s machine, will run locally—only connecting back to the host for data and/or command information. The Stronghold product is easy to use, which is important because errors in complicated setups could compromise use of the product.

Several options are available to encrypt data that travels between the Jswitch data server and the applets. In the original configuration, this data traveled over socket connections without encryption. Only the web pages were encrypted by the web server and decrypted by the web browser. One approach is to use a package of Java classes that encrypt data (i.e., the data assembly classes inherit encryption from the encryption classes). This approach is being pursued with our study using the COTS Phaos SSL Java class library/package.

The other approach is to pass the data packets through the web server and use the web server’s encryption capability. This capability was installed in the Jswitch Epoch prototype in early June and demonstrated to work. The key to using the server/browser encryption is to provide the data to the applet in a standard output (for telemetry) or standard input (for commands) stream via a cgi script. For data moving from the server to the browser, the applet asks to get a stream object (i.e., creates a connection to the standard output of a cgi script) using a statement such as:

```
stream = URL "https://jswitch.nascom.nasa.gov/NPH_event.cgi ? sim1"
```

The standard output stream is established and comes from the NPH_event.cgi script. The web server encrypts because of the use of the https:// directive. The parameter sim1 is used in our configuration to pass to Epoch 2000 to ask for the data source sim1. The NPH_event.cgi script contains a call to event.c that in turn connects to Epoch 2000.

After the applet establishes the stream, it iteratively references stream.getline() to retrieve each packet of data; encrypted by the server and decrypted by the browser. Figure 2 illustrates this architecture.
The SSL web server also provides support for digital identification, which can be used to help assure that clients are connecting to the actual web site and not a "bogus" site. The Jswitch web site uses a digital certificate from Verisign for this purpose. The digital certificate consists of several fields of information, such as a certificate serial number, signature algorithm, and company information (such as the name of the host machine and the project associated with it). Web browsers recognize this piece of information installed in the web server. Clients accessing the site can view the information in this certificate (in Netscape, for example, the user would select "Page Info" from the View menu). This is a useful tool because it is possible to "spoof" web sites, which basically involves forging a source's address. Clients can use this to display the page info to help assure them they are actually accessing the specific site they requested.

Firewall software has been traditionally regarded as a standard security tool to isolate an organization's internal network from the Internet at large. Once organizations have them in place, they tend to feel secure. Many security experts tend to de-emphasize the standard firewall approach now. The problem has been that many companies have relied upon this as their sole security strategy. Once hacked into, the entire internal network may be exposed. Firewalls tend to distract from internal host security problems and give their owners a false sense of security while only improving security by a small amount. We employ a form of "firewall" software on our Jswitch machine called TCP Wrapper. This freely available tool acts as a "wrapper" for common TCP services. It provides for more detailed logging and allows for a host access and deny list for TCP services (ftp, telnet, etc.), all of which are traditionally used for remote access of a machine. The access lists can be set to include ranges of allowable Internet addresses or domain names. More complex and more secure firewall software is available.

The Jswitch machine employs a form of intrusion detection software called SWATCH, which is freely available on the Internet. Often neglected, intrusion detection software forms a valuable part of any secure system, because hacking attempts are indeed sometimes successful, and having this kind of tool can help automate any alert that someone is attempting to hack into the system. The SWATCH tool can be set to monitor the system logs and watch for specified patterns.

Figure 2. Encrypted_Streaming_Objects Architecture
that indicate some potential foul play (such as login attempts from unauthorized users). The software can be configured to send an email to warn the system administrator, beep a pager, or even shut down the server.

Probing software to check the host machine for weaknesses consists of a variety of tools. A freely available tool called SATAN was installed on another host to scan the targeted Jswitch machine for security weaknesses. SATAN is a highly effective scanner tool that can probe a number of potential weaknesses in a host machine. Hackers also use this tool to find vulnerabilities. Another free public domain program, called CRACK, is used to check for password weakness. This tool can scan the current list of users and their passwords and warn of passwords that can be easily guessed by automated tools. Users can then be instructed to switch to a more rigorous password. A couple of simple scripts were also written to check for hidden files on the system and to check for files that have the setuid bit set (a favorite trick employed by someone who has gained illegal control over the system to possibly allow potential superuser-like execution of commands). These are tools that can be run from time to time to do a quick “sanity” check of the system for possible weaknesses and illegal alteration.

Alteration of some system configuration files and services on our Unix machine has led to a considerable improvement in system security. These services can easily be altered by the system administrator. The primary modifications involved disabling services that are enabled in the Internet configuration file (/etc/inetd.conf on most systems). By default, Unix has numerous ports available for various functions, such as querying the network status of the machine, generating random characters, mail, etc. The problem is that some of these ports offer hackers potential access into the system. The solution is relatively simple in that many of these services are not needed for the typical web server, so they are simply disabled. Mail, a frequent source of security holes and denial of service attacks (by flooding a machine with large amounts of mail, thus slowing it down), is effectively dealt with by being disabled. The machine cannot receive mail (although it can send out mail). This is not really a problem for, say, a web page that has a mail address, because the mail address can always reside on a different machine. There are simply very few access points into the system. The limited number of messages produced by probing by hacker tools such as SATAN has borne this out.

We are currently implementing a secure password system from Security Dynamics. This is a standard product in use by operational systems. Jswitch/Isat users will be issued a card that recognizes passwords that are changed every 30 seconds from a password server that is synchronized with time. The card guarantees the identification, and the password server recognizes the user’s identification from the card.

Finally, the configuration of the Jswitch machine itself offers a simple, easily maintainable solution. All the software and hardware resides on one standard Unix machine. It is connected to the open Internet, but the host machine could easily be set up to have no connection to any other internal network. The only connection it needs is for the ground system software to access a socket, which has the connection to data or the actual front end of a ground system. In this way, the data or front-end machines can still be kept behind whatever firewall or other closed network system they currently reside in.

The security techniques and research are ongoing, but the preceding represent the basic steps that we have implemented in the Jswitch prototype and, as such, represent reasonably good security. Any security setup should be kept up to date in terms of awareness of computer bulletins that inform subscribers of newly detected security problems in software and hardware. This is easily accomplished by subscribing to a mailing list of computer security alerts, such as the one maintained by the Computer Emergency Response Team (CERT).

**Operational Scenarios**

By providing a uniform user interface and security mechanism, the Jswitch/Isat architecture supports all typical operational scenarios scripted for users with different levels of operational privileges. Jswitch/Isat naturally integrates into GSFC’s use of multiple redundant strings to support satellite contacts. The Jswitch concept supports important operational scenarios that are needed to move closer to reduced operations staffs and lights-out operations during some shifts. The use of dedicated operations areas can be eliminated for small missions in which operators work from their offices on desktop computers. No special client hardware or software is required beyond a current Web browser.
The following paragraphs describe some typical operational scenarios.

Anomaly Resolution Scenario

- Ground system activities:
  - Receives data during scheduled pass during nonstaffed off shift
  - Detects anomaly from analysis of data
  - Phones and pages operations personnel and/or engineers

- Operations personnel activities:
  - Connects to ground system from home or office or other remote PC using local, personal ISP, or network Internet connection
  - Diagnoses problem by examining telemetry and events from recent or current pass
  - Prepares and issues a real-time spacecraft command or directive that will result in spacecraft command during next pass resolving the anomaly
  - Maintains spacecraft health and safety

Trend Analysis Operational Scenario

Intended user is the scientific community or engineers/analysts. Privileges are limited to playback and analysis of history data on the offline string.

- Activities:
  - User logs in the Jswitch system
  - Jswitch recognizes limited privileges and connects to offline Epoch/ABE string
  - User requests to generate a certain statistical analysis for a specified set of mnemonics
  - Jsat passes the request to ABE, and ABE sends results back displayed in form of textual or graphical (future) dynamic displays or formatted reports

This example shows how easy trend analysis becomes when using Jsat, compared to a legacy control center that involves time-consuming subset generation by the operations team and then distribution of correct subsets to multiple users.

Distributed Operations Scenario

Ground system servers are located in a closet at a convenient location for configuration control and hardware maintenance. Backup system(s) could be located elsewhere for geographical security. Operations personnel and spacecraft engineers perform operations duties from their desktop PCs in their offices during normal work situations and hours. Operations can be performed from remote or other sites during nontypical situations (e.g., conferences or meetings at other centers). Scientists and other researchers can perform science planning, receive science observations, and analyze this information from their own university offices and laboratories. Operations personnel, spacecraft engineers, and scientists share in operations through standard Internet-based groupware (e.g., Microsoft Netmeeting).

Jsat Web Interface Architecture

Figure 3 illustrates the relationship between Java applets, the Java-enabled browser, and the Web. The user accesses a web page, which contains an HTML tag that refers to the Java applet that can be located anywhere on the Web. If the Web browser is Java enabled, the applet is pulled down into the user's client computer and executed in the browser environment.
Figure 4 shows the Jswitch/Jsat architecture. The architecture was derived from the TPOCC web interface (Figure 1) that allows users to access and view selected data over the Internet using a Netscape browser. The Epoch 2000 interface was later substituted for the TPOCC interface. ABE was the most recent addition to Jswitch to create a Jsat component that conceptualizes direct user science data processing and analysis.

This web interface basically works as follows:

1. A Web server runs on the application host machine for remote clients to access.

2. Data server applications running on the application host machine receive socket requests from the applets and send the response back, via a socket, to the connecting Java applet. Another version uses the server encryption to return data packets via a cgi script/standard output interface.

3. The user interface is a Web browser (like Netscape Navigator) in which the user enters the URL for the Web site. This will produce the Web page and download the Java applets from the server to the Web browser on any platform. (Note: applets are not reloaded when they are already in the browser cache.)

4. The Java applets pop up in their own frames, allowing users to enter commands, view real-time event data, monitor telemetry data, and talk to each other.

The ISI Epoch 2000 system and the ABE data extractor are integrated with other COTS products as a mission operations control server running on the Unix Workstation. Epoch 2000 is a flight-proven satellite command and control system that can be used to operate satellites from any manufacturer. It offers a modern graphical interface and is completely database driven. Epoch 2000 supports automated operations using the built-in Satellite Test and Operations (STOL) language. Epoch 2000 can provide integrated services for telemetry processing, commanding, ground equipment monitoring and control, alarm/event processing, and archive/retrieval. ABE is usually associated with Epoch, allowing access to spacecraft databases for the retrieval of telemetry data, events, and trend data files.

Figure 4. Java-Based Spacecraft Web Interface to Telemetry & Command Handling
The Epoch 2000 system provides real-time playback or simulated telemetry and events in the spacecraft downlink to the Jswitch applets. The commands for uplink are received by Epoch 2000, which performs the criticality and constraints checking and subsequently uplinks the commands to the spacecraft. The Jswitch prototype system has demonstrated the command link to Epoch 2000, but no real-time commands have actually been sent to a spacecraft.

The Jswitch Java code interfaces with the Epoch 2000 API (C interface routines) for telemetry, command, and events. The Jsat system will use a standard text editor to generate a profile file that contains all of the information that ABE needs to generate an ABE trend table and store the data in a file for retrieval by the Jsat system. A Jsat applet retrieves the data for display on any platform with a Java enabled browser.

The Jswitch/Jsat system has a current set of tools for encryption, certification, and authentication to protect client-side security, server side security and document confidentiality.

**COTS Products**

**Jswitch Software**

The current COTS products in the Jswitch prototype suite are described below.


2. Digital id by Verisign (http://www.verisign.com) (Further details can be found at http://digitalid.verisign.com/Payment.htm.). Helps assure the client that the web site connected to is the legitimate one.


4. Jchart library by KLM group (http://www.k1g.com/jclass/overview.html. Components are used to provide graph widgets (such as plots, bar charts, etc.) for our Java source code.

5. Jswitch Java source code (free, http://moca.nascom.nasa.gov). Source for the Java applets the user exercises to download and display via the user’s web browser. Also employed to provide for a base for the data servers. (They serve as a conduit to get data from the ground system and funnel it to the Java applet running on the client’s machine.)


7. Ground system interface software (free). Code called by the data servers to actually get the requested information from the specific ground system. There are specific formats for this software to use to work with our Java source, which will vary depending on the ground system. Ground system interfaces are written in C. Freeware C compiler (such as GNU C) can be used.

8. Software and configurations to make the client machine more secure. (Assumes a Unix workstation, although similar products are available on an NT system, for example.) All these products are freely available to download and/or involve modifying some system files. They are downloadable from various sources on the Internet; one central source is ftp://coast.cs.purdue.edu/pub/tools/unix/

- TCP Wrapper—Provides a wrapper around traditional TCP utilities, such as rlogin, telnet, and ftp. Gives more detailed logging information in terms of who is connecting to the system and allows the administrator to create lists of allowed/denied users of the system.

- SWATCH—Provides a monitoring capability of the system log files. Can be set up to look for patterns of breaking or illegal access in the system log files and alert in the form of email or even a call to a pager.

- SATAN—Should be set up on a different host machine and can be used to probe a system for security weaknesses.

- SSH—Provides a “secure” shell for rlogin, telnet, and ftp, by encrypting the data that goes back and forth (standard remote logins, for example, go across the net as clear text). This is not in use yet, but is planned.

- Modify /etc/inetd.conf services—Can be modified to disable a host of ports and services that are on by default but typically not needed
and often serve as ports of entry for hacking into the system.

- Secure ID by Security (http://www.securitydynamics.com)—Provides a strong form of user authentication in that users use a "card" that gives them a password to type in. The password changes every 30 seconds and is therefore unlikely to crack. This requires a password server to be running (see http://www.securitydynamics.com/solutions/products/asvrdata.html) for more details, Secure ID cards (see http://www.securitydynamics.com/solutions/products/tokens.html for more details), and web server agent to be running (see http://www.securitydynamics.com/solutions/products/agent4ns_index.html for more details). Note that, at this time, the product only works with Microsoft's web server and the Netscape series of servers.

**Jswitch Hardware**

Clients can theoretically use any Web browser that is Java aware. Because of differences in some vendors' Web browsers, we find that Netscape Navigator 4.0 seems to work the best. This is free and can be downloaded from http://www.netscape.com.

The host machine, needed to serve the Java applets, run the web server, interface to the ground system, and run the interface servers, has been Sun's UltraSparc 2, running the Solaris 2.6 for the operating system.

The host machine does not need a Sun box as long it is running some form of Unix. It could very well be some Hewlett Packard (HP) workstation, for example. The only caveat here is that it should have an adequate amount of memory and processing speed to handle all the connecting clients. For that matter, with some extra work, the host machine could even be an NT box.

**Staff Estimates to Install a Jswitch Configuration**

These estimates are very rough and are provided only to give an idea of the ease of installing and configuring a Jswitch capability. One to two staff-weeks is estimated to install the OS and configure the Unix host, install the freeware security tools, and set up and configure the web server and Java source.

The effort for ground system interfaces can vary. It depends upon whether there is any existing code in the ground system that would provide for an interface. It also depends on how much this interface varies from the current format Jswitch uses. It could be anywhere from one to several weeks. There are two areas to consider here:

- Modification of the Java source that deals with the data server formats (these are confined to one module). This can vary according to the format that will be received (or it could be used "as is" if the data format being sent by the ground system interface program produces output in the same format Jswitch currently uses).

- The ground system interface routines (for telemetry, events, and commanding). These will typically be non-Java utilities (written in C, for example) that interface to the internals of the ground system (and are invoked the Java application socket servers to send/receive data from the ground system to the Java applet running in the client's web browser).

Finally, the applet code can be used as is, but it is expected that several weeks would be needed to customize the displays, and the same effort would be set aside as part of ongoing maintenance to further enhance the displays. Note that an upcoming version of the Jswitch suite will have configurable display palettes using Java bean technology.

**Jswitch/Jsat Implementation**

A web server runs on a host machine to provide access to clients. From the server's web page, users, by using their web browsers and connecting to the host site, can select which Jswitch/Jsat functions to invoke (command, events, telemetry, or trend analysis).

Selecting a function will invoke a Java applet, which will be downloaded to the client machine and then run locally. The Java applet will pop up a window, providing a user interface for the specified function. The applet will make a socket connection back to the host machine to send and receive data from a data server that is running on the host machine.

Abigail H. Maury
Figure 5 shows the initial command, telemetry, and events applet running on a PC. Figure 6 shows a screen snap of the command, events, and telemetry applets after a command has been sent and a telemetry chart and text display have been selected.

The command applet provides the user with a text entry field for entering commands to the ground system, as well as a menu selection of common commands that can be inserted by clicking. Once a command has been entered, clicking on the submit button will send the command to the host machine, where the data server will send the command to the ground system, get the results back, and then send it back to the command applet. The command applet will display the original command, along with the result, in a scrolling text area.

The event applet provides the user with a list of events from the ground system. Once activated, it will connect to a data server, which fetches events from the ground system. These events are displayed in the event applet in a scrolling text area. Event messages are color coded (red, out of limits; yellow, starting to exceed limits; green, normal limits; and white, spacecraft events). The user can filter on events via entering the desired filter in a text field on the applet.

The telemetry function provides the user with a list of "pages" to select (see Figure 7). These pages are ASCII text files residing on the host machine that describe the appearance of a telemetry page. The pages include a list of data points to display and instructions on how to display them. Currently, the displays consist of basic text displays and bar charts for more graphical displays (see Figure 8). Additional graphical displays are available by utilizing COTS products, which provide Java graphical libraries. Once the user selects a desired page, this information is sent to a data server, where data updates are sent to the telemetry applet, which displays the telemetry information in a window. The user may define their own mnemonics for graphical display (see Figure 9).

Jsat interfaces with ABE, the COTS spacecraft telemetry data analysis package from ISI that provides the user with the following set of browsing and data analysis functions:

- Retrieval and decoding of actual telemetry data
- Creation of tables and graphs of the decoded telemetry data

Figure 5. Command, Telemetry, and Command Applet
Figure 6. Applet After Command Has Been Sent and Telemetry Selected for Display

Figure 7. List of Pages To Select
Figure 8. Telemetry Displayed in Applets

Figure 9. Create Text Page Applet
• Creation of tables and graphs of statistical information for user-defined telemetry data
• Retrieval and display of spacecraft events
• A profile feature that allows "canned" setup for analysis
• Creation of postscript files and printing

The ABE package is hosted on the server and is usually associated with Epoch 2000, allowing access to spacecraft databases for the retrieval of telemetry data, event, and trends data files. The ABE package uses PV-WAVE, a commercial general-purpose visual analysis package from Visual Numeric, Inc. PV-WAVE is widely used in the technical community as a foundation for data analysis.

The current Jsat prototype accesses the trend data files. The directory and files are user selectable in an applet (see Figure 10). The file may be viewed before downloading. Once "OK" is clicked, the data is downloaded and displayed in a Java applet window. The data continuously updates until the end of the dataset is reached (see Figure 11).

Work is ongoing to command ABE remotely from the browser and to send the data to a chart in a chart library commercially supplied by the KLM group. The remote interface consists of a simple ABE command interface routine written in C. This remote interface processes a profile file that can be generated with any text editor that will tell ABE to write the data to a text file that we want to process. If the user has a limited access only to an offline string, the applets later can be replaced by applications to enable the user not only view but to store and manipulate data locally on user computer.

Platform Independence

JsWind/Jsat client has been tested on PC, Macintosh, HP, and Sun platforms. Netscape Navigator Gold 3.0, Netscape Navigator Professional Edition 4.04, and Internet Explorer 3.0 browsers have been used. Browser differences were noted in each instance with the rendering of the applets. Differences consisted mainly of font size, widget sizes, and colors. We have standardized by using Netscape Navigator 4.04 for development purposes. On the PC, the primary development platform, these differences were traced to individual preferences in control panel settings.

![Figure 10. Trend GUI To Select Trend File](image-url)
The colored events applet rendered true red, yellow, green, and white events as foreground colors when the Color Palette was set to “256 colors” for Netscape Navigator Gold 3.0. However, on the same PC (Compaq Deskpro), with the Color Palette set to “256 colors,” Netscape Navigator Professional Edition 4.04 rendered the foreground colors incorrectly: red was orange, yellow was pastel yellow, and green was teal green. When the color definitions were reversed to background in Netscape 4.04, the colors were correct. Further investigation revealed that if the PC Color Palette was set to “high color 16 bit,” the foreground colors were rendered correctly in Netscape Navigator Professional Edition 4.04.

The trend analysis applet was rendered with the last column in a second row underneath the label column on the developer’s PC and correctly on a second PC with all columns in one row. Turning the screen size on and off corrected this rendering on the developer’s PC.

One exception was noted in the execution of code between the Java Virtual Machine on Internet Explorer 3.0 and Netscape Navigator Professional Edition 4.04 (see Figure 12). On the Internet
Explorer 3.0 Java Virtual Machine, the data values were updated correctly in the labeled text fields; however, the time was incorrectly updated in the Time text field. The time appeared correctly while the first four labeled text fields were updated under the AMN statistical type. When the next set of data was retrieved for the last three statistical types (MED, MIN, and MAX), the time was incorrect and would correct itself at the beginning of each new set of statistical data beginning with MIN.

Netscape 4.04 always displayed the time correctly. The Internet Explorer 3.0 Java virtual machine has incorrectly interpreted the following statement:

\[
\text{if(datapoint.time} \neq \text{"XTIME")}
\]

At this point, further investigation is needed to determine the root cause of the problem. However, if the virtual machine is not standard in each browser, thus...
leading to execution errors, the ramifications are quite extensive. Spacecraft cannot be reliably monitored and controlled in our application. The cost of extensively testing different browsers is prohibitive. We need to test once and run anywhere. To decrease risk until this problem is further researched, we recommend running only on the browser on which the code has passed acceptance test. The final solution to this problem probably lies with the Java virtual machine as an add-in to the browser. This product is under development by Sun.

Jswitch/Jsat: Spacecraft Control Center Future Applications

Jswitch/Jsat is an ongoing demonstration of emerging technologies to support lights-out operations at reduced costs. The growing maturity of technology (communications, security, etc) and the spread of the Web will be an integral part of all of our lives in the next century. The Jswitch concept is currently being integrated with extending Internet Protocol to satellites, such that the spacecraft is a node on the network. This concept of a spacecraft as a node on the network involves defining a standard format for mapping IP over RF links.

GenSAA, a Government-off-the-shelf (GOTS) product, is being integrated into Jswitch/Jsat. This product contains a data server that is already integrated with other command and control systems in existing mission systems. GenSAA also contains a graphical user interface (GUI) definition tool and GUI definition language that we are translating into Java by a preprocessor under development as part of the Jswitch prototype suite. This Java preprocessor will deliver spacecraft mission displays to a Web browser using Java bean technology.

Summary

Jswitch/Jsat fits very well with inexpensive small satellite rapid development

Jswitch/Jsat can quickly provide reusable ground software, with the user evaluating the software from their location and providing immediate feedback to the developers. Scientists have a quick, simple access to process their science data.

Building ground systems faster is achieved first by using existing software and second by integrating the system using an iterative development process. In the Jswitch prototype systems, several existing COTS products were used. Core command and control functions are readily available in several COTS products (e.g., ISI Epoch 2000, STI OS/Comet, Altair MCS, LM SCS 21) and GOTS products (TPOCC, ASIST, ITOS), to name a few. Epoch 2000 represents a flight-proven technology available as a COTS product. To make use of the WWW, COTS web servers and browsers were examined that include security features and support the Java virtual machine. The security features needed include encryption and user authentication through passwords. These features are available in the C2net (Apache) Stronghold COTS web server. This product is used in the Epoch 2000 version of the prototype. Several other products provide various security features, and many of these are free in the public domain.

The Jswitch concept enables a better ground system by providing the definition of standard GUI elements that are based on the Java language and applets served up through the Java virtual machine under standard web browsers. This is a powerful combination, because virtually every work location already has recent, free web browsers from Microsoft or Netscape already installed. These products are available for Windows 95, Macintosh, and most versions of UNIX operating systems. In summary, the user benefits from the following capabilities:

- Scientist can communicate with instrument using a Web browser.
- Engineers and operators can work from office, home or on the road using only a Web browser.
- Client computers are platform independent (Web browsers run on PC, Macintosh and Unix platforms).
- Anomaly resolution is facilitated, with remote access making lights-out operations during off-shifts realistic.
- Costs are lower: no special hardware and no special client software are required beyond a current Web browser.
The use of standard COTS products for major functions provides not only a quicker development cycle but also a less expensive option. This is sometimes mistaken for a panacea. Some COTS products are expensive, and some do not perform as advertised. Obtaining a free trial license for evaluation mitigates this risk. Despite the problems, however, COTS products that can be used in the spacecraft support domain have matured over recent years and present a very good value compared to custom development or even to modification of existing software. The reasons for this include the following:

- Development costs are amortized over many customers.
- The product is enhanced to keep pace with evolution in supporting technologies to remain competitive.
- New features are added as part of product development, also to remain competitive.
- Maintenance costs are usually quoted as a known percentage of initial license costs.
- Special vendor support is usually available for short periods as needed rather than having to maintain dedicated software maintenance staff.

Given the acceptance of product enhancement and maintenance costs, there are many GOTS software products, which are also excellent candidates to include in mission ground support systems.

The Jsswitch/Jsat prototype system has demonstrated that we can receive telemetry, send commands, process events, and analyze statistical data over the Web. The Jsswitch system can be tailored to interface the Java language through native code to any control center. The Jsat component can easily download science data that has been written to a file. The FOT and the scientific community can easily be cross-trained on an uniform interface. Mission-specific applications are easily written and downloaded as applets to the user with a standard Java-enabled Web browser. The user always has the latest software updates, and there is no special hardware to buy.

**Biographies**

Ms. Abigail H. Maury has a BA in Chemistry from Emory University, a MS in Microbiology from The University of Georgia, an MS in Computer Science from The Johns Hopkins University, and a MS in Technical Management from The Johns Hopkins University. At Computer Sciences Corporation, Ms. Maury developed real-time control center software systems for many NASA missions, including the International Sun Earth Explorers (ISEE-A, B, C), Solar Maximum Mission (SMM), Landsat-D, and others. Ms. Maury was the project manager for the software development of the Earth Radiation Budget Satellite (ERBS) and Compton Gamma Ray Observatory (GRO) Control Centers and deputy project manager for the installation of the HST orbital verification system at Marshall Space Flight Center. Ms. Maury served as project manager on an ESA project funded by the Italian Space Agency to develop the control center telemetry and command systems for the X-SAR instrument that flew successfully on STS 59 and 68 in 1994. Recently, Ms. Maury elected to augment her software management skills by bringing her technical knowledge and background more up to date with applied programming courses in C, Unix, and Java. She has enjoyed Java so much she volunteered to become a Java developer and subsequently developed the command, colored events, and trend analysis Java software for the Jsswitch/Jsat system.

Ms. Anna Critchfield received her MS in Computer Science in Moscow, USSR. Ms. Critchfield emigrated to the United States in 1978. Since 1981 at Computer Sciences Corporation, she has worked on NASA and ESA projects, including Landsat, HST, GRO, EUVE, ISTP, SMEX, X-SAR, and ACE. She was a team leader of the Control Center software development tasks for GRO, WIND, POLAR, and ACE satellites and for the X-SAR instrument. Currently, Ms. Critchfield is involved in the reengineering of the GSFC control centers using COTS and Web solutions.

Mr. Jim Langston received his MS in Physics from Florida Atlantic University. Mr. Langston has 20 years of experience engineering and developing large software systems, primarily for the aerospace domain. For the last 5 years, Mr. Langston has worked to insert new technology into spacecraft ground support systems as part of the GSFC Renaissance team. Mr. Langston participated in the IMACS and BIOS prototype systems and has been involved in the Jsswitch prototype suite development from the point of conceptualization. Mr. Langston is also involved in a concurrent activity to demonstrate the use of Internet protocols in space communication and onboard spacecraft.

Ms. Cindi Adams has a BS in Mathematics from Bowie State College and a Masters in Engineering Management (MEM) from George Washington University. Ms. Adams has been managing spacecraft software development at Goddard Space Flight Center.
since 1985. At GSFC, Ms. Adam’s career milestones have included Control Center Systems Manager (CCSM) for the Solar Heliospheric Observatory (SOHO) Project and the Compton Gamma Ray Observatory (CGRO) Project and System Architect for the Earth Observing System (EOS) AM-1 Flight Operations Segment (FOS). Currently, Ms. Adams ensures that all interface documentation for AM-1 FOS is baselined and that changes are tracked and incorporated into the interface documentation. Ms. Adams is the Landsat 7 Mission Operations Center (MOC)/Landsat 7 Simulator (LSIM) Integration Manager. Ms. Adams is the GSFC task lead for the Java-based Spacecraft Web Interface to Telemetry & Command Handling (Jswitch) task. Ms. Adams has been recognized for many outstanding accomplishments. The most current recognitions have included the National Technical Association Top Women Scientists & Engineer Honoree (1997) and a Group Achievement Award in recognition for dedication, management, and technical expertise in defining, developing, integrating and making fully operational the large-scale international SOHO Ground Data System (1997).