

# A WORLD WIDE WEB INTERFACE FOR AUTOMATED SPACECRAFT OPERATION

Christopher Kitts\* and Clemens Tillier†

## ABSTRACT

A ground based intelligent agent and operations network is being created to handle all aspects of spacecraft command and control. This system will have the dual purpose of enabling cost efficient operation of a number of small satellites and serving as a flexible testbed for the validation of space system autonomy strategies. The system is currently being targeted to include over a dozen globally distributed amateur radio ground stations and access to nearly ten spacecraft. The use of distributed computing systems and virtual interaction schemes are significantly contributing to the creation of this system. The Internet is used to link the network's control centers and ground stations. In addition, a World Wide Web (WWW) based user and operator interface is being developed to permit high level goal specification of spacecraft experiments and actions. This paper will describe the operating network being developed, the use of the Internet as an integral part of the system's architecture, the design of the WWW interface, and the future development of the system.

## KEY WORDS

Human computer interface, Visualization, Spacecraft operations, Goal/task level control, Automation.

## INTRODUCTION

Revolutionary breakthroughs in information technology and the rapid increase of affordable computational power are providing strong impetus for the automation of a new generation of services and devices. From automotive engine control to Internet based software agents to on-board satellite control, these developments are responsible for enhanced competitiveness by increasing features such as performance and by reducing costs in areas such as operations and maintenance.

---

\* Doctoral Candidate, Stanford University. Caelum Research Corporation, NASA Ames Research Center, CA.

† Engineers Candidate, Stanford University.

Providing advanced user interfaces to the users and operators of these systems is one particular technique for enhancing a system through the use of automation. Such interfaces can be used to permit convenient specification and delivery of services to external customers. Similarly, well designed interfaces for system status and control can assist operators in judging the efficiency of service provision, the health and status of the system, the location and cause of faults, the availability and demand for resources, etc.

Advances in Internet technology and tools provides a low-cost and ubiquitous opportunity for providing user interface resources in distributed systems. Internet connectivity can provide the communications backbone for these interfaces. The graphical nature and general familiarity of the WWW adds significantly to the ease of use of such interfaces. Finally, the ability to provide automated multimedia attributes such as animations, sound, and three dimensional visualization constitutes a powerful set of tools for creating easy-to-use, high level, interactive system interfaces.

## THE SPACE SYSTEM DEVELOPMENT LABORATORY RESEARCH PROGRAM

Stanford's Space Systems Development Laboratory (SSDL) is investigating the value of Internet based user interfaces for the purposes of operating its space systems. SSDL was chartered in 1994 to provide world class education and research in all aspects of spacecraft design, technology, and operation. To achieve this goal, SSDL members enroll in a comprehensive academic program composed of coursework, project experience and research investigations. With respect to studying user interface technologies, SSDL is creating Internet based command and telemetry interface systems in order to control university and amateur spacecraft as well as the ground equipment used to manage these spacecraft.

The Satellite Quick Research Testbed (SQUIRT) Microsatellite Program - The SSDL SQUIRT program is a yearly project through which students design and fabricate a real spacecraft capable of servicing low mass, low power, state-of-the-art research payloads.<sup>1</sup> By limiting the design scope of these satellites, the project is simple and short enough so that students can benefit educationally by seeing a full project life cycle and by being able to technically understand the entire system. Typical design guidelines for these projects include using a highly modular bus weighing 25 pounds, a hexagonal form that is 9 inches high by 16 inches in diameter, amateur radio communications frequencies, and commercial off-the-shelf components. Missions are limited to about one year of on-orbit operation. Because little money is available for mission operations, a highly automated mission control architecture is being developed.

The Automated Space System Experimental Testbed (ASSET) System - The ASSET system is a global space operations network under development within SSDL. This first goal of this system is to enable low-cost and highly accessible mission operations for SQUIRT microsatellites as well as other university and amateur spacecraft. The second goal of this system is to serve as a comprehensive, low inertia, flexible, real world validation testbed for new automated operations technologies. Figure 1 shows a high level view of the ASSET mission architecture. The basic components include the user interface, a control center, ground stations, communications links, and the target spacecraft. During the current development phase, a highly centralized operations strategy is being pursued with nearly all mission management decision making executed in the control center. These tasks include experimental specification, resource allocation throughout the ground and space segment, fault management, contact planning, data formatting and distribution, and executive control.<sup>2</sup>

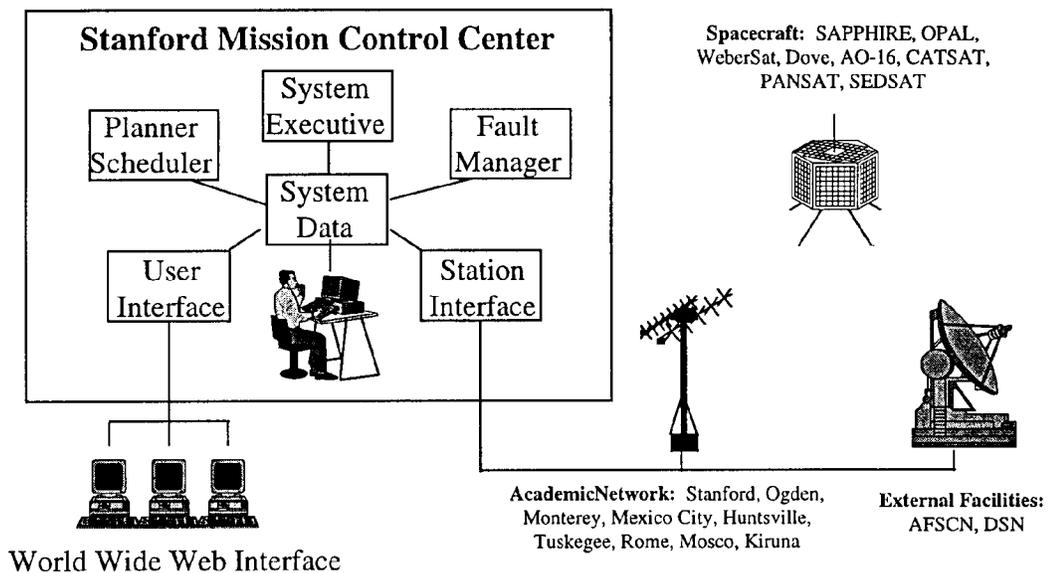


Figure 1 - The ASSET Space System Architecture

WeberSat, a Weber State University and Amateur Satellite Organization spacecraft launched in 1990, and SAPPHERE, the first SSDL SQUIRT spacecraft being completed at this writing, are currently being integrated into the ASSET system. In addition, controllers for five satellites have expressed interest in becoming part of the system. As for ground stations, the Weber State and Stanford ground stations are the first two facilities to be included. Seven other stations throughout North America and Europe have been identified for future integration.

## DESIGN GOALS FOR A SPACECRAFT OPERATIONS INTERFACE

The design of the ASSET user interface is one of the current projects being pursued. The functions of this interface include accepting experimental parameters from users, providing experiment constraint checking, loading parameters into the system database, notifying users on the status of experiments and system status, and making data available for display and download. In performing these functions, three primary design objectives are being targeted.

First, the interface should permit decoupled user interaction with the mission control center with respect to time and space. From a historical perspective, spacecraft mission operations have often required collocated and/or synchronized interaction for users and operators due to a required level of personal interaction, the use of specialized equipment, security considerations, or other applicable reasons. Automation of many other mission management tasks, such as resource scheduling, is relaxing the requirement for such coupling. The ASSET system meets this design requirement through the use of the WWW and automated request processing. This feature should increase the convenience and accessibility for system users.

Second, the interface is being developed to interact with the user at a level appropriate to their experience. To implement specific user requests, the system must have execution level knowledge of the desired task. The ASSET system permits experiments to be specified at this level. The generation of this execution level knowledge, however, may not be of interest to or within the capability of the user. For some users, a very high level description of the task may be more suitable. As an example, to broadcast a message from the SAPPHIRE spacecraft, the vehicle requires the date and time of transmission. Most users, however, think of specifying the experiment in terms of who or what area they'd like to have hear the message. The ASSET interface compensates for this by permitting users to click on a world map at the target location for the broadcast. Using orbital dynamic models, the automated system then converts the map location to a suitable date and time that will be used in the execution level command sent to the spacecraft. In this way, users with little knowledge or inclination can easily use the system; in addition, experienced users will still retain the ability to control low level activity in the system.

Third, the interface seeks to guide the user to a request that meets the stated objectives while being as general as possible. As an example, consider the alternative methods of specifying the previously described SAPPHIRE broadcast function. If, for instance, that the user wants somebody in Washington, D. C. to hear a message anytime during the next week. At the low level, the user calculates one particular time and date when this can be achieved and submits an execution level command. At the high level, the user clicks on the map and adds a deadline; this submission generates a number of applicable times and dates

that meet the overall goal of the user. Clearly, the later method is preferable since this lends flexibility to the system. Fundamentally, the nature of goal level direction helps to prevent the generation of overconstraints that ultimately prevent system optimization.

To meet these design objectives, the ASSET system is being developed with a World Wide Web (WWW) based interface capable of high level interaction. Use of the Internet provides a widely available, convenient, low cost communications link between users and mission control. Goal level design of the interface allows 1) simple and general experimental requests And 2) ease of use for the operator.

## IMPLEMENTATION OF THE WORLD WIDE WEB INTERFACE

The ASSET user interface is hosted on an existing web server which serves the entire Department of Aeronautics and Astronautics at Stanford. The server, a Sun SPARC 5 running httpd software, is capable of executing programs through the Common Gateway Interface (CGI) standard. The language used to implement the user interface is Perl. Perl features powerful string manipulation capability, and is therefore ideally suited for processing text input from the user as well as processing output from satellites linked to the ASSET system. The user interface is built following the traditional client-server architecture, shown in Figure 2. Server security issues have not been explored at this stage.

User Input - The input segment of the ASSET interface consists of a set of hypertext markup language (HTML) web pages and their associated Perl scripts. The user specifies an experiment request at several different levels, in a progression from general input (e.g. name, address, choice of experiment) to specific input detailing the user's request. As shown in Figure 3, this results in a tree structure of web pages, through which the user progresses from root (top-level) to leaf (specific request). At each stage in the input procedure, the user fills out an HTML form, the contents of which are processed by that page's associated Perl script. Simple error checking is performed on all inputs and the user is served back the appropriate web page, depending on the outcome. If the input passes the initial error screening, the data is appended to a temporary text file in a standardized format suitable for later use by the ASSET system. Should the user have made an error, a dynamically generated error message is served back with information on how to correct the error and links to help files.

One of the challenges with the tree structure for HTML pages and Perl scripts is the need to track many simultaneous users without mixing their inputs. There is no consistent means of identifying users in hypertext transfer protocol (HTTP), so a solution needs to be devised in order to achieve this. Identification is often accomplished through the use of a registration giving each user a separate name and password. In the case of the ASSET interface, it was decided to use a simpler solution: every user beginning a request is

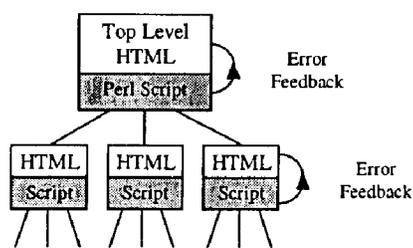
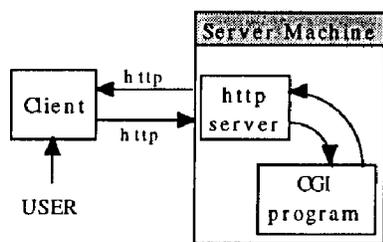


Figure 2 - Client-server architecture      Figure 3 - Tree input structure

assigned a unique identifier (UID). The UID is actually the server's process identification number for that particular HTTP connection, and is therefore unique for each simultaneous user. The UID is passed along everywhere the user goes, hidden in the HTML code. Whenever an HTML page is served back to the user, a Perl script dynamically inserts the assigned UID into every form (as a "hidden" text input) and every uniform resource locator (URL) (as a query string). The next time the user follows a link or submits a form, he/she is uniquely identified and can thus be tracked reliably through a request without regard to other simultaneous users.

A typical user's interaction with ASSET's input interface might go as follows. At first the user is greeted with the top level web page, shown in a screen shot in Figure 4. In this example, the user has entered her name and address, selected a satellite telemetry dump for her experiment, and specified "any" in the satellite choice menu. When the form is submitted, its associated Perl script writes this information to a new temporary file and serves back a page which contains a form to be used for specifying a telemetry request. In this case, the form served back, shown in Figure 5, concerns the SAPPHIRE satellite and prompts the user for dates and times and just what is to be sampled. Once the user has submitted this data, it is appended to the appropriate temporary file and a thank you message is served back. The interface is designed to forward this request to the ASSET database for scheduling and execution.

User Output - The output segment of the ASSET interface serves experiment results back to the user, and allows browsing of data already archived in the ASSET database. Upon arrival, raw satellite data files are pre-processed and stored in the database. The user is automatically notified and directed to the results of her particular experiment. Different telemetry subsets can be dynamically generated from the masterframe and served to the user using Perl scripts as shown in Figure 6. These scripts not only present the data in a clear and organized fashion but are capable of performing operations on the data such as converting raw values into meaningful units and performing statistical calculations.

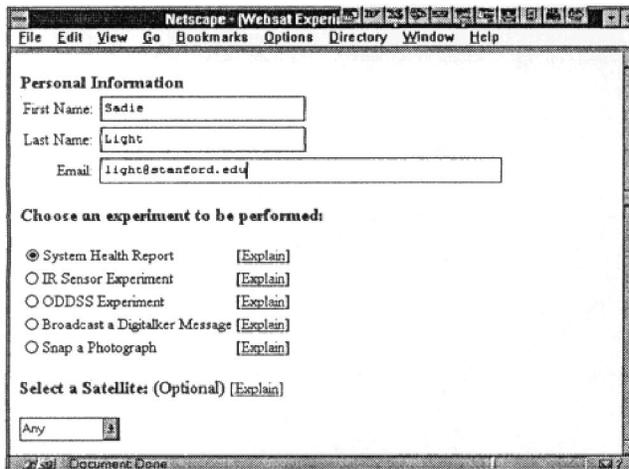


Figure 4 - Selection of an experiment

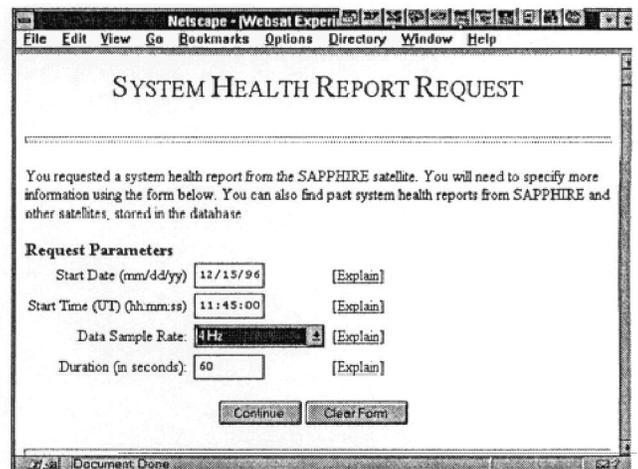


Figure 5 - Specification of experimental parameters

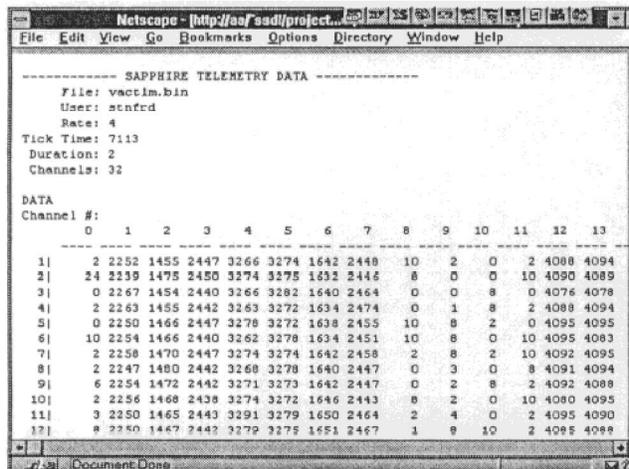


Figure 6 - Display of experimental results

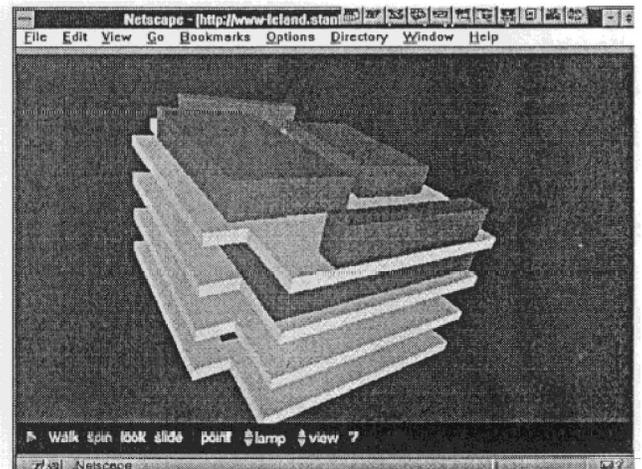


Figure 7 - 3-D Display of SAPHIRE components

The output interface is also designed to allow high-level goal specified browsing of the experiment database. For example, if a user would like to see the typical behavior of a particular satellite as it goes into eclipse, telemetry sets that exhibit this feature can be extracted from the database. Long-term trend analysis can be accomplished as well, through automated parsing of a large number of past data files.

Engineering Data Visualization - In addition to servicing experimental requests, the ASSET interface is also being developed to provide enhanced engineering analysis of spacecraft telemetry. Early spacecraft telemetry display systems typically scrolled information across the operator's screen as information was decoded from the telemetry masterframe. The current state-of-the-practice in telemetry display technology typically uses fixed frame formatting in which pages of information are organized by spacecraft

function. This, for example, is the current technique employed by the Air Force Satellite Control Network, the NASA Pioneer project, etc. Cutting edge interface techniques are beginning to employ advanced visualization techniques ranging from iconic representations to immersive environments.

Given the use of the WWW based interface, the telemetry display system for the ASSET project is experimenting with the use of the Virtual Reality Modeling Language, a new three dimensional modeling specification compliant with many of the advanced Web browsers. Through the use of this language and previously described techniques such as CGI scripting and animation, three dimensional representations of the spacecraft, its environment, and its components can be generated and used for health analysis. As an example, Figure 4 shows a tray level view of the SAPPHIRE spacecraft; in this particular display, the color of each box represents the temperature of the component. Other displays are being developed to show the vehicle's attitude, orbital position, environmental conditions, component status, system schematics, etc.

## FUTURE WORK

Development of the ASSET system is an ongoing research activity within the SSDL. Work is being performed in all functional areas in order to incorporate more advanced and cost-efficient capabilities. These capabilities will enable the system to include more users, missions, spacecraft, and ground stations within its architecture. Improvements to the user interface include upgrading its goal level experimental specification capability, providing an easy access to the database of spacecraft data, and enhancing the content and display of spacecraft health information.

The work performed to date has attracted significant attention and has laid the groundwork for a variety of initiatives with other agencies and universities. First, as part of the NASA New Millennium Program (NMP), much of the user interface work performed on the ASSET system is providing technology feedback to NMP's autonomy development team. Second, as part of the U. S. Air Force Satellite Control Network's (AFSCN) new commercialization initiatives, SSDL will attempt to apply its interface strategies to the AFSCN mission architecture in order to enhance the provision of services to the commercial sector. Finally, SSDL hopes to build upon previous cooperative activities with the University of Umea in Sweden in order to create operator interface software that will enable telepresence operation of remotely stationed amateur radio ground stations.

## CONCLUSION

Advanced command and telemetry interfaces have proven to be useful in the operation devices and the provision of services due to their ability to provide convenient interaction at a level appropriate to the user. This is especially true in the field of spacecraft operations due to the distributed nature and complexity of the systems involved. Design of the ASSET user interface is showing how high level, automated, multimedia, Internet based interfaces can be used to enhance the operation of these systems for both customers and system operators. This work is directly contributing to the space industry and is paving the way to achieving NASA's goal of providing ways for the general public to conduct science from their desktop.

## ACKNOWLEDGMENTS

The authors wish to acknowledge the outstanding work and commitment of other members of the current ASSET design team: Mike Swartwout, Andrew Miller, and Tim Caro-Bruce. In addition, special thanks is given to NASA's Jet Propulsion Laboratory and Ames Research Center for their support and assistance with this research project. Finally, appreciation is extended to SSDL's SAPPHIRE design team as well as Weber State's WeberSat operations team for their willingness to participate in, contribute to, and provide feedback concerning various aspects of the ASSET system.

## REFERENCES

- [1] Kitts, Christopher A., and Robert J. Twiggs, "The Satellite Quick Research Testbed (SQUIRT) Program." In Proceedings of the 8th Annual AIAA/USU Conference on Small Satellites in Logan, Utah, September 16-22, 1994, by the Utah State University. Logan, UT: Utah State University, 1994, 80-83.
- [2] Kitts, Christopher A., "A Global Spacecraft Control Network for Spacecraft Autonomy Research." Pending publication in the Proceedings of Spaceops '96: The Fourth International Symposium on Space Mission Operations and Ground Data Systems in Munich, Germany, September 16-20, 1996, by the Electronic Publishing System. Munich, Germany: German Aerospace Research Establishment, 1996.
- [3] Net.Genesis, and Devra Hall. Build a Web Site. Rocklin, CA: Prima Publishing, 1995.
- [4] Schwartz, Randal L. Learning Perl. Sebastopol, CA: O'Reilly & Associates, Inc., 1993.
- [5] Pesce, Mark. VRML: Browsing & Building Cyberspace. Indianapolis, IN: New Riders Publishing, 1995.