Science Opportunity Analyzer - A Multi-Mission Approach to Science Planning

Barbara A. Streiffert
Jet Propulsion Laboratory/
California Institute of Technology
4800 Oak Grove Dr, MS 301-250D
Pasadena, CA 91109-8099
+1(818)354-8140
Barbara.Streiffert@jpl.nasa.gov

Carol A. Polanskey
Jet Propulsion Laboratory/
California Institute of Technology
4800 Oak Grove Dr, MS 301-250D
Pasadena, CA 91109-8099
+1 (818) 393-7874
Carol.A.Polanskey@jpl.nasa.gov

Taifun O'Reilly
Jet Propulsion Laboratory/
California Institute of Technology
4800 Oak Grove Dr, MS 301-250D
Pasadena, CA 91109-8099
+1(818)354-1170
Taifun.Oreilly@jpl.nasa.gov

Joshua Colwell
Laboratory for Atmospheric & Space Physics
University of Colorado, 392 UCB
Boulder, CO 80309-0392
+1(303)492-6805
Joshua.Colwell@lasp.colorado.edu

Abstract—In the past Science Planning for space missions has been comprised of using ad-hoc software tools collected or reconstructed from previous missions, tools used by other groups who often speak a different "technical" language or even "the backs of envelopes." In addition to the tools being rough, the work done with these tools often has had to be redone or at least re-entered when it came time to determine actual observations. Science Opportunity Analyzer (SOA), a Java-based application, has been built for scientists to enable them to identify/analyze observation opportunities and then to create corresponding observation designs for orbiting, astronomical (telescope), and flyby missions.

TABLE OF CONTENTS

1. INTRODUCTION 1
2. SCIENCE OPPORTUNITY ANALYZER 1
3. INITIALIZATION (GLOBAL/KERNELS)........... 2
4. OPPORTUNITY SEARCH 2
5. OBSERVATION DESIGN 5
6. CONSTRAINT CHECKING 8
7. DATA OUTPUT... 10
8. COMMUNICATION 11
9. SUMMARY 11
10. CONCLUSION.. 12

1. INTRODUCTION[1]

Spacecraft mission operations is comprised of diverse teams competing for the limited resources of a spacecraft. Each team has its goals and its "own language." Flight teams talk about spacecraft commands and command parameters. They're involved with hardware components and computer transactions. Science teams are concerned with science objectives, physical phenomena, celestial bodies and their instruments on the spacecraft. What has been missing is a software tool that allows scientists to plan and develop their observations using the terms that are meaningful to them. At the same time this tool must communicate with project software in terms of commands and command parameters. Finally, the tool must be able to support a wide-variety of missions. Science Opportunity Analyzer (SOA), a Java-based application, has been built to meet these needs.

2. SCIENCE OPPORTUNITY ANALYZER

SOA supports science mission operations for orbiting, astronomical (telescope), and flyby missions. At Jet Propulsion Laboratory (JPL) mission operations is divided into two basic categories: mission planning and mission sequencing. Mission planning starts with very high level plans having blocks of time set aside for spacecraft trajectory correction maneuvers, for observations, and for sending data to earth for an entire mission and ends with the building of a detailed set of spacecraft activities and science observations. Mission sequencing takes those activities and observations and turns them into spacecraft commands. SOA can be used for all aspects of the science portions of mission planning, but is primarily for use towards the end of that phase where the scientist is determining and developing science observations. SOA is the first multi-mission tool to allow the science user to enter those observations into the front-end of the spacecraft command pipeline using science friendly terminology and an intuitive graphical user interface and then send the observations to downstream software that will turn them into spacecraft commands. It has been built with communications to other "uplink" software tools in mind.

In addition to allowing the observation to be sent to downstream software, SOA permits scientists to look for opportunities of geometric interest (Opportunity Search). It also contains features to allow them to analyze their

[2]IEEAC paper #1448, Updated October 22, 2002

observation to see if the observation meets both the desired science objectives and with reality. Various views can be displayed at user-selected times to examine the geometry of a user-selected target (Visualization). The user can also view that information with one or more instrument fields of view projected onto the target. A field of view (FOV) is one of the apertures of a given instrument. If the display of the selected time is promising, the scientist can begin to create an observation and then view the observation (Observation Design). If further investigation of the observation is needed, the user can get observation or trajectory related data to help in making the decision whether to keep this observation or to change to another time (Data Output). The data regarding the spacecraft trajectory can be retrieved at any point in the process. Finally, the scientist can check the observation to make sure that it does not violate any constraints such as "too much bright light exposure" on a sensitive instrument (Constraint Checking). Along with the ability to communicate with other tools (Communication), SOA provides a convenient package of functional capabilities in one generic multi-mission tool.

In the following paragraphs an example of one path that a user will typically take is presented along with pictures of the screen displays that the user will see. This path can have variations based on the user's needs and desires, but the path is ultimately one that will end in a science observation that is placed into the plan of activities to be sent to the spacecraft. The path will be called the Science Observation Planning Process in this paper.

SOA is configurable to a specific mission. This process is called "adaptation". Adaptation adds the mission specific data for SOA to work with. This mission specific data will include flight rules, observation types, physical phenomena models, celestial body data, instrument data, spacecraft-specific data and a spacecraft trajectory. Much of this data can be provided through the use of JPL navigation files.

3. INITIALIZATION (GLOBAL/KERNELS)

Step 1—

(Required) The user runs an SOA configuration file.
(Note: The configuration file is a Java Bean shell script that is executed by the Java Runtime Environment.)

Even though initializing SOA isn't part of the Science Observation Planning Process it is a necessary step for SOA to have data to work with. The user can select, edit or run a configuration file. The configuration file selection is under the Global/Kernels tab. The configuration file determines the spacecraft, spacecraft data and spacecraft trajectory to use, instrument data, planetary constants information, opportunity searches that are available, many of the defaults in SOA including the seeding of the default time windows, and other initialization components. The configuration file is a Java Bean shell script and as such is executed by the Java Runtime Environment. Once the configuration file is run, the color-coding of the SOA main screen changes from red to various other colors so that the user can tell visually if a necessary item is missing. Figure 3.1 shows SOA prior to running the configuration file (all buttons and tabs are red) and Figure 3.2 shows SOA after running the configuration file (all buttons and tabs have changed color and none are still red).

Often this configuration file will be built by the project as part of the adaptation of SOA. In this case, the user will simply run the file that is delivered with the software. The SOA User's Guide that is delivered with SOA at each release describes in detail how to modify the configuration file. The SOA User's Guide has been written by a scientist to ensure that the science user community can easily understand it.

4. OPPORTUNITY SEARCH

Step 2—

(Required) The user finds one or more windows of opportunity for an observation based on specified geometric criteria.

Initially, in the Science Observation Planning Process, the scientist wants to find observation opportunities that meet specific geometric criteria. The Opportunity Search tab allows the user to find when the desired geometric criteria are met. On this display (Figure 4-1) the scientist either selects to build a search query or to load previously built queries from a file. If the search query is new the user is presented with the search query builder window (Figure 4-2). The list of available queries is loaded at runtime and appears in the upper right hand side of the query builder window (Figure 4-2). The list has over 30 query types that include such things as eclipses, distance from a specified celestial body, periapse events, transits, etc. The user selects the desired query from the list and simply drags it into the graphical query window. Now the query's properties (such as search time window, celestial bodies involved, etc.) can be customized for the selected search. Once the search query is built the user selects it and presses a button for the search to begin.

SOA allows the user to create either a simple search query or a complicated one. The user can use the relational operators of "and," "or," or "not" to create complex queries. The graphical query window helps the user to keep track of the relationships by building a tree so that the user can see their relationship. The scientist doesn't have to remember complicated command line syntax for either type of search query. That complexity is handled by SOA. SOA creates the correct command line syntax for the search engine specified by the scientist and sends the information directly to that search engine via inter-process communication.

SOA has two search engines that process the search queries. These search engines work as root finders on continuous functions. Some queries result in a single time (like a periapse) and others result in time windows (like an occultation). The search engine returns the resultant time(s) when the search criteria are met. The windows of opportunity are displayed for the user. The user now has a choice of selecting to see a display of that time, save the results to a file, see data related to a result window, begin an observation design or do any combination of these four. SOA always allows the user to save his/her work to files for future recall.

The delivered SOA currently contains all of the search criteria types. Generally, a mission will not have to perform any adaptation for this area of SOA. If a search doesn't apply to a given project, it can simply be removed. A project can also use a different search engine, but then more adaptation work will have to be done. That work is described in the SOA Adaptation Guide delivered with the software.

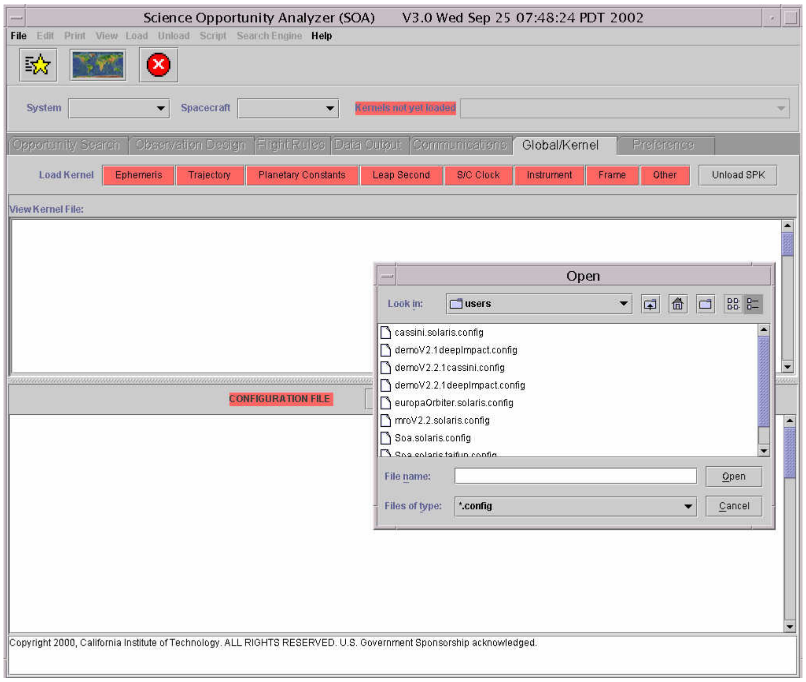

Figure 3-1. Initial SOA Global/Kernals Tab display with Configuration File Selection Box open.

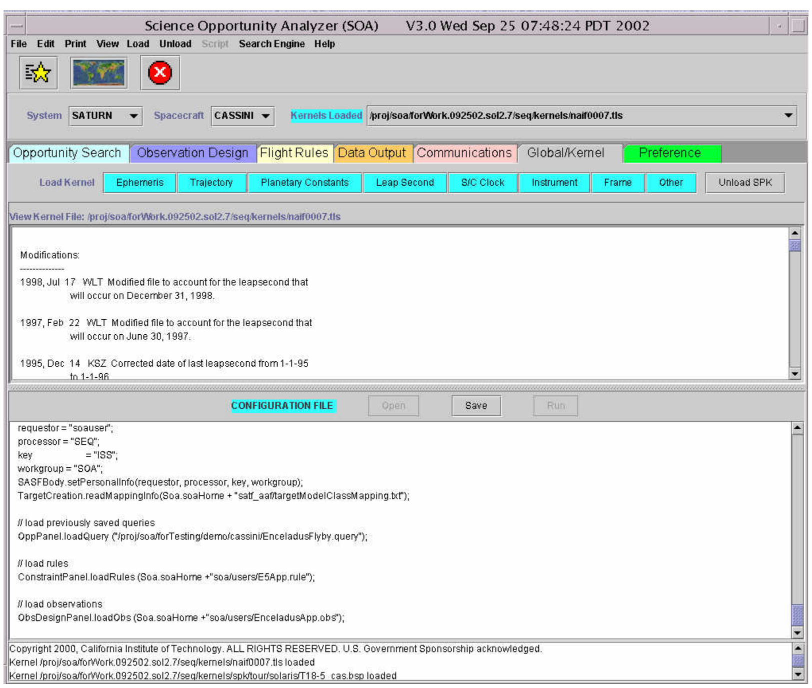

Figure 3-2. Global/Kernals Tab display after configuration file has been run.

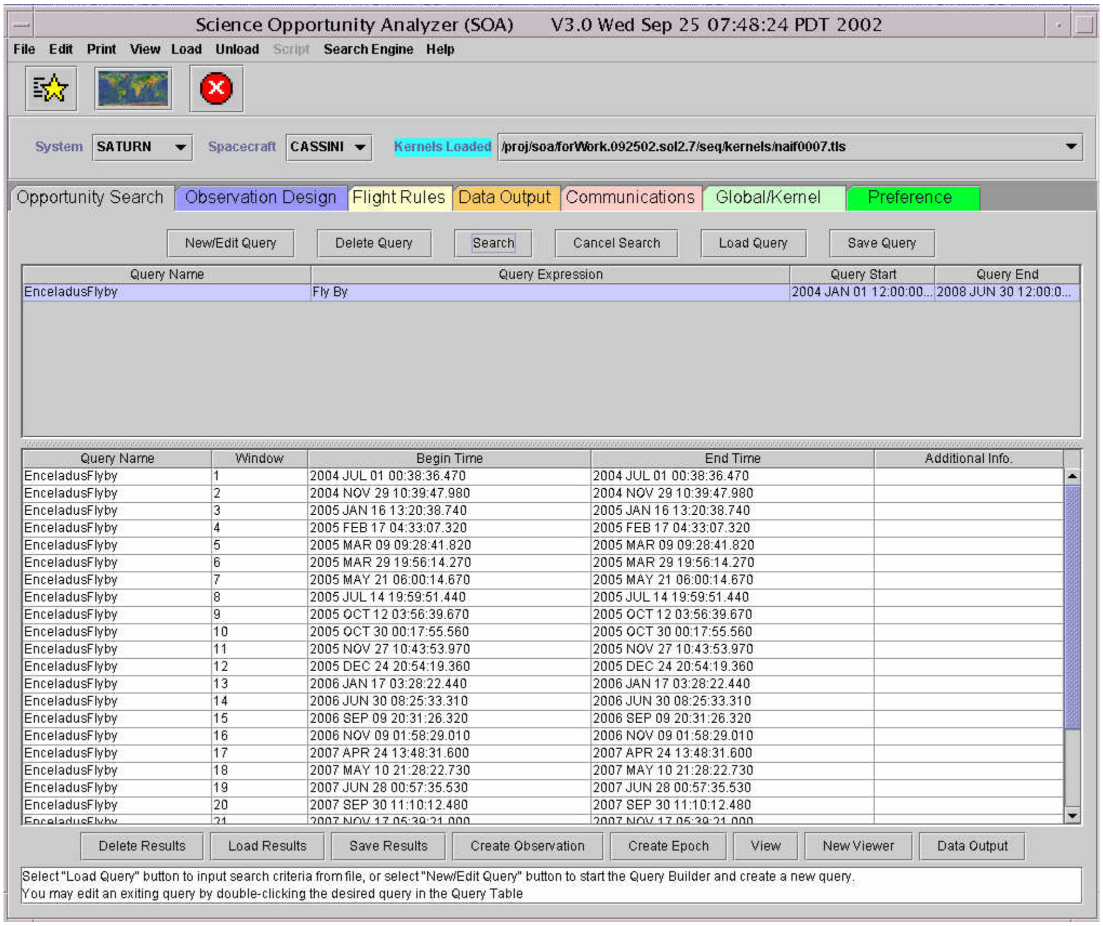

Figure 4-1. Opportunity Search Tab display with the Enceladus Flyby query selected and the resultant search windows displayed.

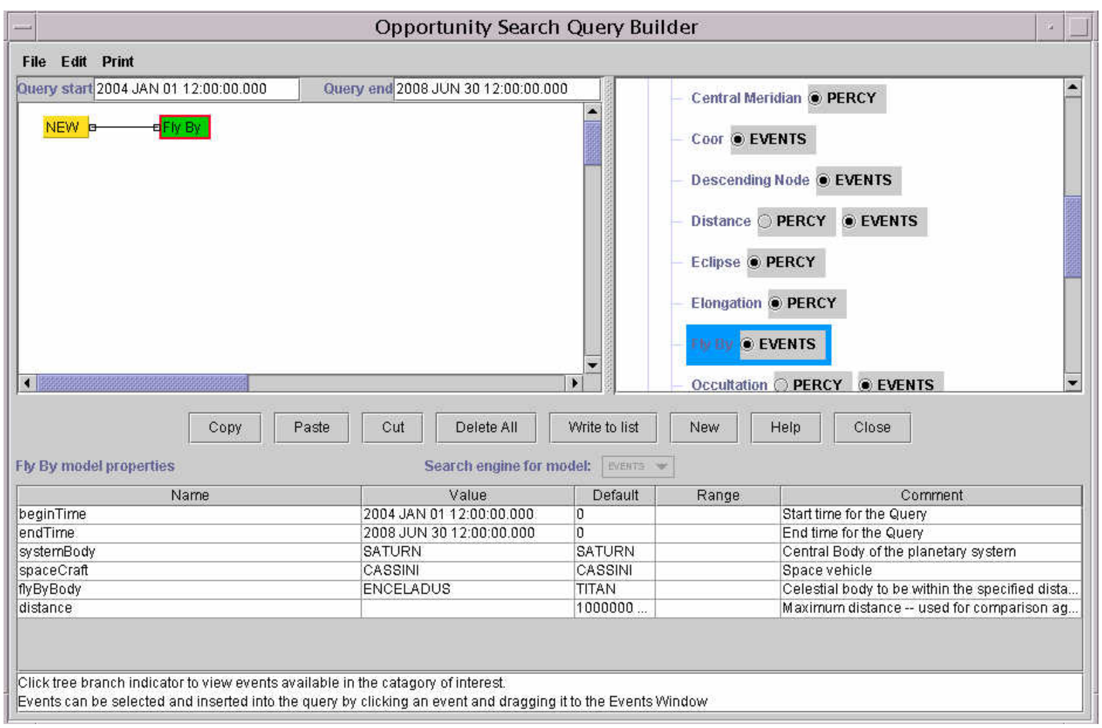

Figure 4-2. Opportunity Search Query Builder display with an Enceladus flyby query.

5. Observation Design

Step 3—
(Required) The user selects a time window and begins to design an observation.

Step 4—
(Optional) The user views the design in a visualization window.

Prior to this step the user may have selected a time from the Opportunity Search Results and selected to view an SOA display at that time (as stated in the previous section), but the user may have created the search criteria in such a way that any resultant window is a candidate to use to perform the observation design. The Observation Design tab allows the scientist to begin the design.

The scientist can select to simply look at a time with a single instrument field of view (FOV) to get a visual idea of the "landscape" or he/she can select to create one of the four different types of observations. The simple view is called a "scoping" activity and shows the user a view from a specific spacecraft attitude. This type of observation usually doesn't contain enough information to be sent to downstream software, but it can give the scientist an idea of whether or not this observation is worth pursuing. It has very few properties (parameters) and requires very little of the user's time to specify while providing the user with a significant amount of visual information. Figure 5-1 shows the Observation Design display with a scoping-level activity specified.

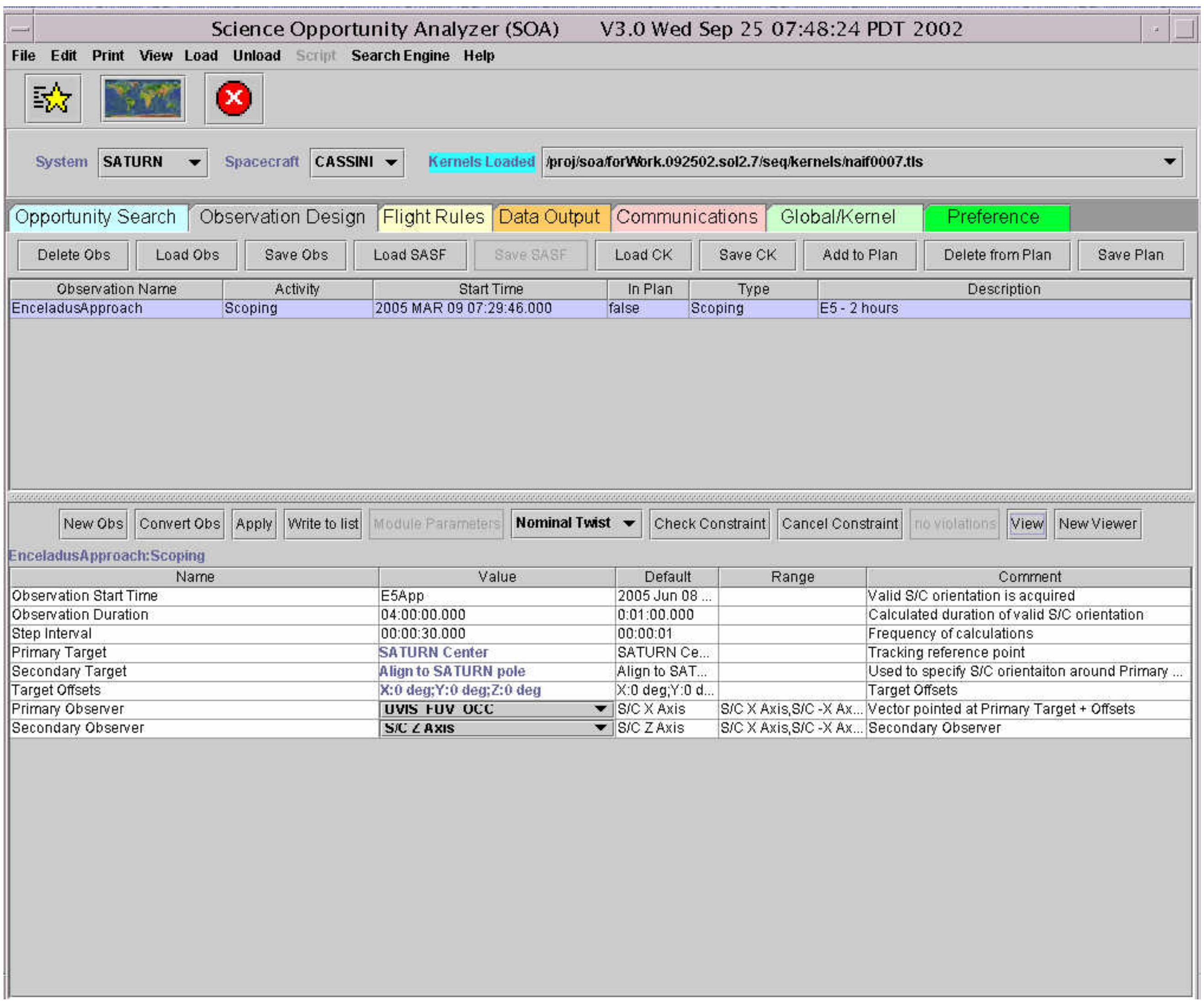

Figure 5-1. The Observation Design Tab display showing an Enceladus scoping level observation with constraint violations (the orange button with the date in the center of the display).

At any point the scientist can begin or continue working on a previously saved observation design. The four basic types of observations are:

1. Start Stop Mosaic
2. Continuous Scan
3. Roll Scan
4. Stare

A Start Stop Mosaic observation is one in which the spacecraft or scan platform moves to a series of locations, stops (or dwells) at each location and performs an observation at each of these locations. Then it moves to the next location and performs the next observation. A remote sensing instrument such as a camera normally uses this type of observation. The Start Stop Mosaic has also been called a box scan or an N by M mosaic. The Continuous Scan observation is similar to the Start Stop Mosaic, but for this observation the spacecraft or scan platform is continuously moving while observations are being performed. Remote sensing instruments such as spectrographs or spectrometers often use this type of observation. The Roll Scan observation is an observation where the spacecraft generally rolls about a single axis. However, it is conceivable that the rolling could be about some sort of platform or other mounted device that allowed an instrument the freedom to spin. Fields and Particles In Situ instruments extensively utilize this type of observation. Finally, the Stare observation is one where the spacecraft or scan platform tracks a single target direction for the duration of the observation. Instruments of all types use this category of observation.

Each of these observations has its own set of properties (parameters). However, some of the properties are common to all such as the start time of the observation, the planned duration of the observation, the target, the observer and any time margins used to make sure that the observation fits into its allotted time frame. Other properties are specific to the type of observation. Some examples of properties for specific observation types are:

1. Start Stop Mosaic: the number of footprints per scan and the total number of footprints.
2. Continuous Scan: the number of scans (or rows).
3. Roll Scan: the roll axis.
4. Stare: the actual stare duration.

Projects may have their own variations of these activities or maybe even an activity type that is completely different from the ones provided. SOA has been built using a hierarchical structure so that new or variations of existing observation types can be created. Performing observation adaptation is also described in detail in the SOA Adapter's Guide.

After the scientist has filled in the properties, he/she will want to view a display of the celestial bodies and the observation. SOA has four types of display: 3-dimensional perspective projection, 3-dimensional arbitrary observer, 2-dimensional sky map and 2-dimensional trajectory view. The 3-dimensional perspective projection (Figure 5-2) is a 3-dimensional view of the target from the point of view of the observer. The 3-dimensional arbitrary observer is a 3-dimensional view that is a parallel projection of the target from an observer that can be arbitrarily placed in space. The 2-dimensional sky map is an equidistant cylindrical projection of the celestial sphere as viewed from the spacecraft. Finally, the 2-dimensional trajectory view is a view of the spacecraft trajectory around a target in either the ecliptic or the equatorial plane. If the target has satellites their orbits can also be shown. Each of these displays can be animated over the time span of the observation. The scientist can choose to see all of the fields of view that form the observation at one time or to see them as they are performed during the animation. The scientist can also look at more than one view at a time on a single screen (Figure 5-3) or have multiple screens displaying the same or different views (Figure 5-4). Each viewer window has data to allow the user to obtain more information about the observation quickly, and also has controls to allow the user to customize the view of the observation as well as to animate it.

On the left-hand side of the visualization window the data display gives the scientist information such as the range to the target and the right ascension and declination of various points of interest. Below the data display, the user can use the Settings Panel to make display objects visible or invisible. Additionally, the user can also change properties of display objects such as color, line width, font, etc. as they are applicable for that object. The last items on the left hand side are the animation controls. The user can control the animation through the use of these text entry slots and buttons.

On the right hand side the user has a series of sliders. These sliders allow the scientist to control the amount of rotation in x, y and z of the current view. The bottom slider allows the user to zoom in or out on the current display. If the sliders are not available for use in a particular view (such as the 2-dimensional trajectory plot) they are grayed out.

Most of the needed basic graphics primitive objects are provided currently in SOA. A hierarchical structure similar to the one provided for other adaptable items is available for the graphics primitive objects as well. The low-level graphics primitive objects can be combined to form most any display object that is needed by a project. Currently, meshes and 3-dimensional volume rendering objects are not available, but those objects are to be developed for future versions of the software.

A scientist can remain in the observation design area until the observation is completed and all of the observation objectives are met or he/she can proceed to the next step at any point while creating the observation and then return to observation design as needed. The next step in the process is constraint checking.

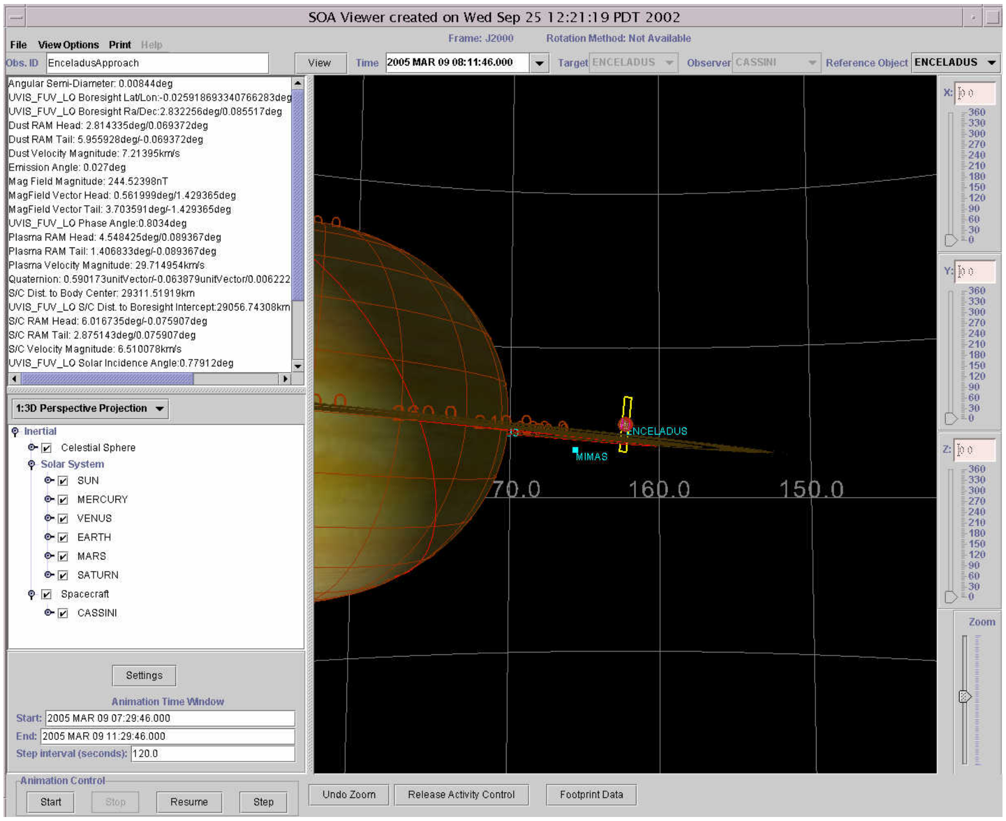

Figure 5-2 A 3-D perspective view of the Enceladus observation.
The yellow rectangle around Enceladus is the field of view of a spectrograph.

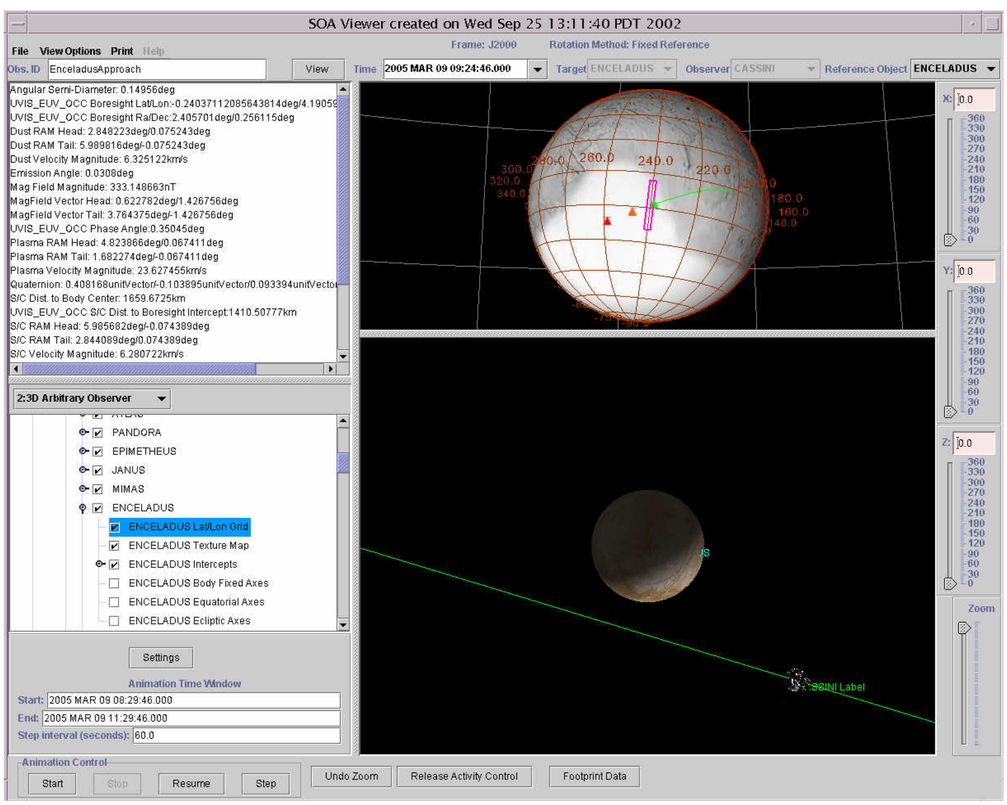

Figure 5-3. A 3-D perspective view and a 3-D arbitrary observer view of closest approach to Enceladus. Note that the Cassini spacecraft appears in the 3-D arbitrary observer view as well as the spacecraft trajectory.

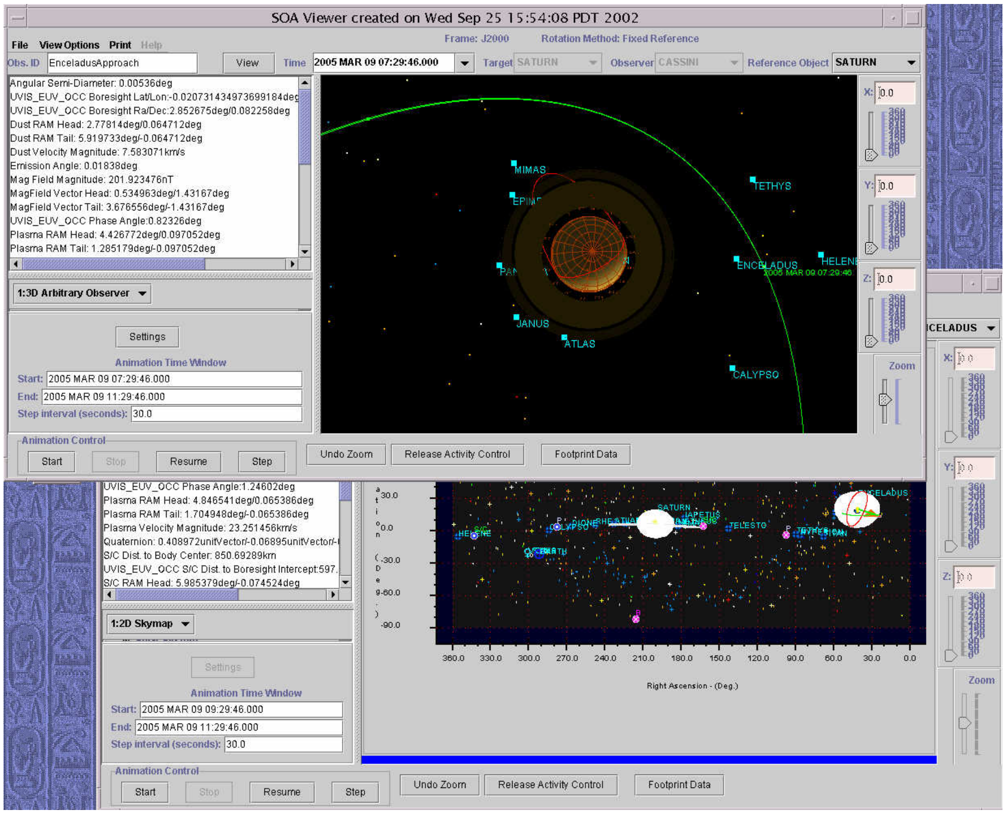

Figure 5-4 Two-viewers are displayed. The top one shows a 3-D Arbitrary Observer view with Saturn as the target. It shows the spacecraft trajectory by Enceladus. The bottom one shows the 2-D SkyMap view at this time.

6. CONSTRAINT CHECKING

Step 5—
(Required) The user checks to see if any constraints have been violated.

Now the scientist wants to make sure that the observation meets with the rules that the project has provided. These rules are often called flight or mission rules. Rules are checked at the observation level. In downstream software used for sequencing the rules are checked at a command level. In SOA rules come in two varieties: geometric or dynamic. The geometric rules are generally ones that have to do with exclusion zones (such as the avoiding pointing a delicate camera lens at the Sun). The dynamic type has to do with illegal spacecraft states (such as requiring the spacecraft to exceed its maximum turn rates).

The exclusion zone rules have several flavors. The zone is generally an angle that forms a region where there's a bright body that must be avoided in order not to damage an instrument. Sometimes if the instrument is far enough away from the bright body, then the light is not harmful. In this case the rule can specify both the angle and the distance that are to be avoided. Finally, the instrument may only be harmed if it is exposed for a given period of time. In this case exposure times less than the specified time are not harmful. These rules can be created using any combination of the three items: angle, distance, and exposure time.

The dynamic rules deal with placing the spacecraft in an illegal state. For SOA generally that means exceeding maximum spacecraft rates and/or accelerations or not meeting minimum spacecraft rates and/or accelerations. These rates can apply to the spacecraft, a scan platform or an instrument. Individual rules and rates can be created for each of the pieces of hardware that must be checked.

Before actually selecting to check the rules, they have to be built. The user can select to build new rules or use the ones that the project has supplied through adaptation. The rules are created using a drag-and-drop builder screen (Figure 6-1) similar to the search criteria builder in Opportunity Search. The rule type is selected and dragged to the graphical rule-building window. To enter the rule values, the user just clicks on the graphical rule box and the specifics for that rule can be entered at the bottom of the screen. In addition to creating simple rules, SOA also allows the user to create complex rules using the logical operators of "and" and "or."

In the main Rule tab (Figure 6-2) the user has the option of enabling all of the rules to be checked, none of the rules to be checked or some but not all of the rules to be checked. After the rules have been built and the ones that are to be checked have been enabled, the user can save these rules to an XML formatted file for future recall. The project may have delivered a project set of rules and the user may not need to build or enable any rules.

If the user wishes to create additional rules, the user may do so and create his/her own file that can then be loaded in addition to the project's rules . The user simply adds this new set to be loaded using the configuration file.

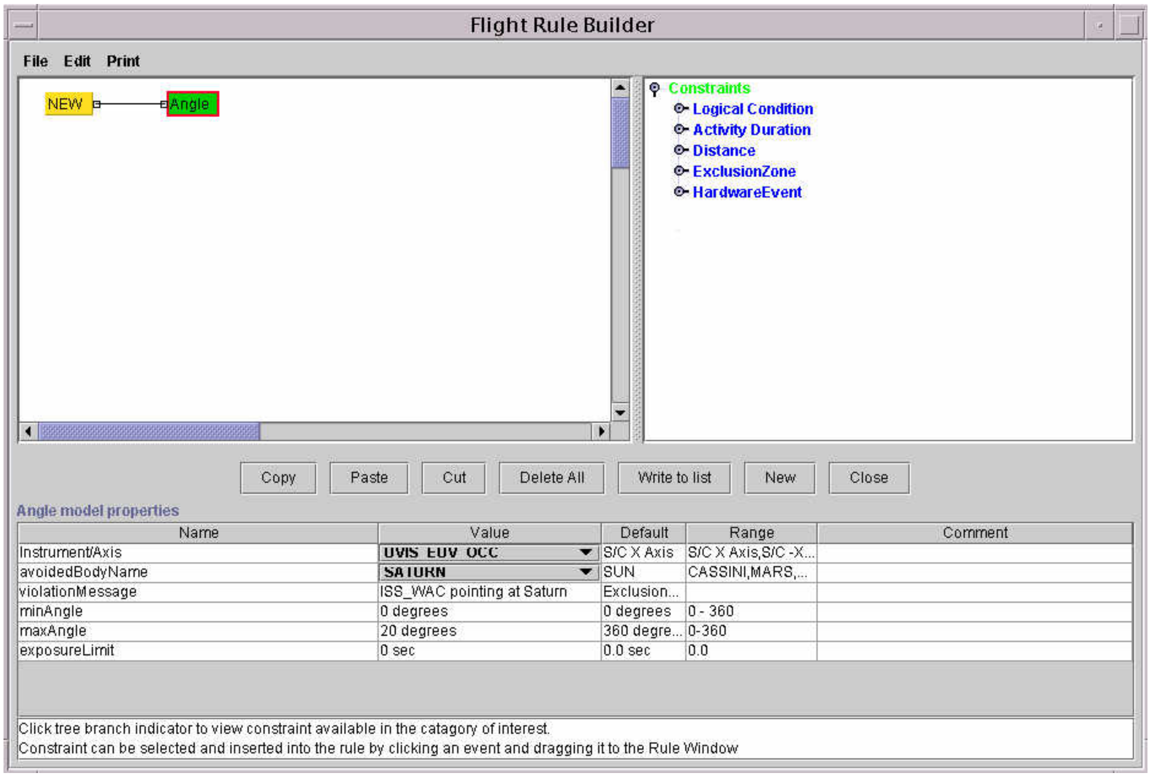

Figure 6-1. The Flight Rule Builder display shows an Angle Rule being built.

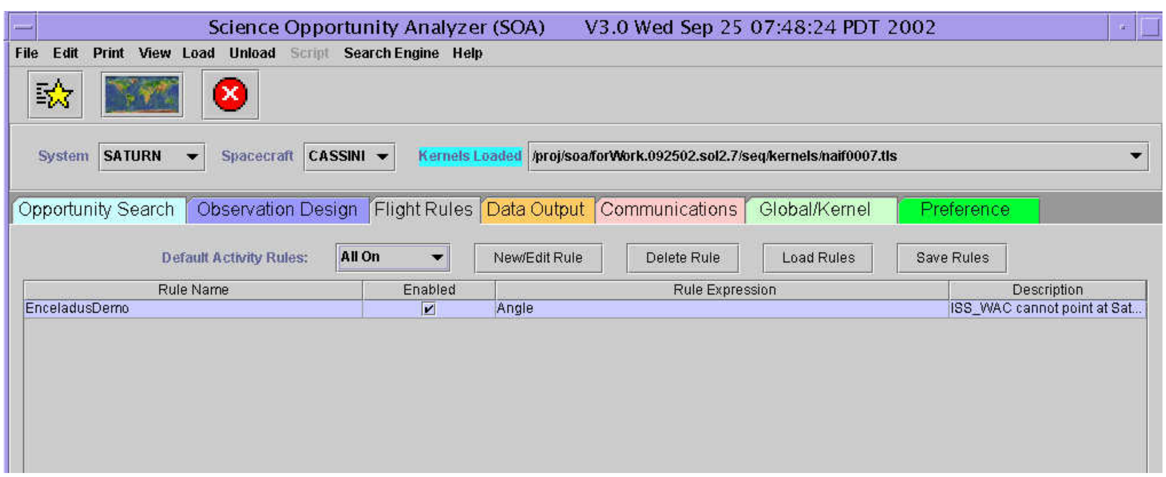

Figure 6-2 The Flight Rule Tab display shows the Enceladus Angle Rule selected to be checked when the Check Constraint button is selected. Figure 5-1 shows the results of checking this flight rule.

Once the rules have been loaded, the user simply returns to the Observation Design tab and presses the constraint check button. SOA checks all of the enabled constraints and returns with a message that either violations were or were not found. If violations are found, they are listed in the constraint log that pops up when the orange constraint violation button is pressed (see Figure 5-1 in the previous section). If no violations are found the constraint violation button stays gray and says "no violations." In addition, if the user displays a visualization of the observation, the footprint(s) that are in violation are indicated in red.

In most cases projects can use the rule builder screen to do all of their adaptation. One project using SOA has been able to create 20 rules in one hour with the rule builder. If the project has rules that are not supported by the rule builder, SOA provides a similar hierarchical structure in order for project adapters to add new rule types or new rule building blocks.

7. Data Output

Step 6—
(Optional) User selects and views output data to ensure observation viability

At any point in the process the scientist can view output data produced by SOA. The Data Output tab (Figure 7-1) allows the user to select the time, the type and kinds of output data to be viewed as well as allowing the selection of viewing the data in tabular or graphical form. There are three types of data output: (1) spacecraft trajectory related data, (2) opportunity search results data, and (3) observation related data.

The Spacecraft Trajectory Related Data option allows the user to select output that relates to data that can be determined without the user having to specify an instrument field of view (FOV). These data can be determined by simply knowing the spacecraft path. Certain physical phenomena model information, the sub-spacecraft point and other information that doesn't require FOV information can be requested.

Opportunity Search Results Related Data option allows the user to select data that pertains to the results of a particular opportunity search criteria. Again this information doesn't require knowing a particular instrument FOV to obtain.

The Observation Design Related Data option allows more information to be requested. In this option the user can specify data that is calculated based on the instrument FOV such as emission angle. The emission angle (or viewing angle) is an angle calculated between a FOV intercept with the target and the spacecraft's intercept with the target. In order to calculate this information, the observer field of view data must be known.

All three types of data can be requested in a tabular file format. The resultant file can be placed in a commercial spreadsheet and analyzed. The data can also be presented in a graphical format (Figure 7-2). In this case the scientist has a choice of how the graphs are to be arranged. The choices are: stack plot (one or more plots per window), x/y or x/y/z plot (a single plot per window) or all-in-one plot (all data items appear in a single plot in a single window).

As with all of SOA the project can choose to use the data output items that SOA has provided or the project adapters can create additional types of data output and hook those into the hierarchical structure that SOA uses. The data output that can be specified currently in SOA is generally sufficient for most projects.

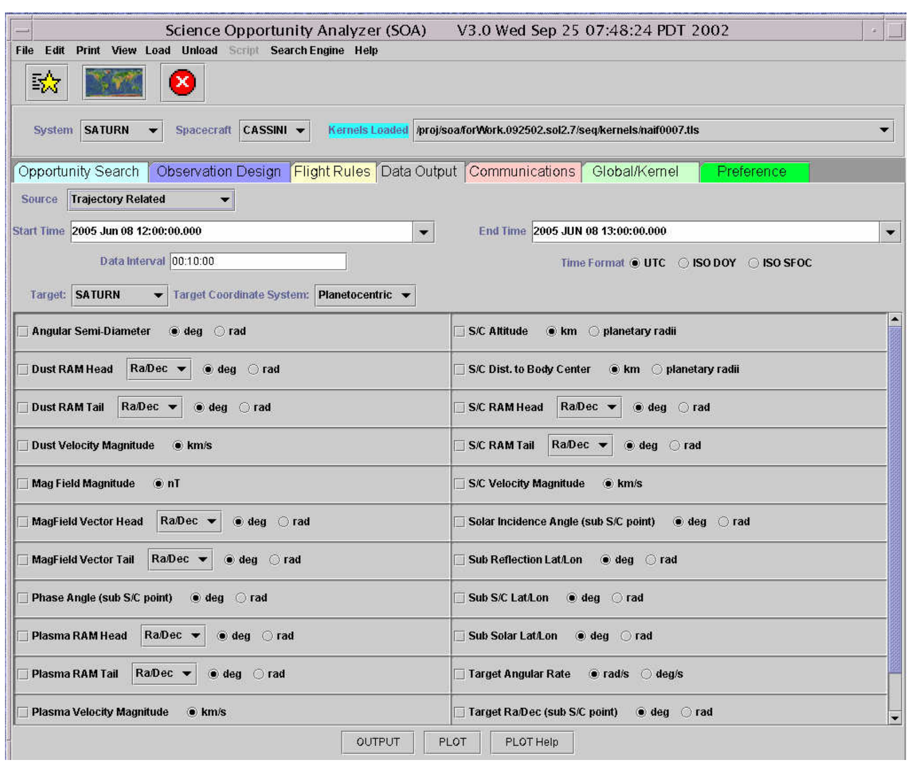

Figure 7-1. The Data Output Tab display shows the data items available for selection in the area of Spacecraft Trajectory Related Data.

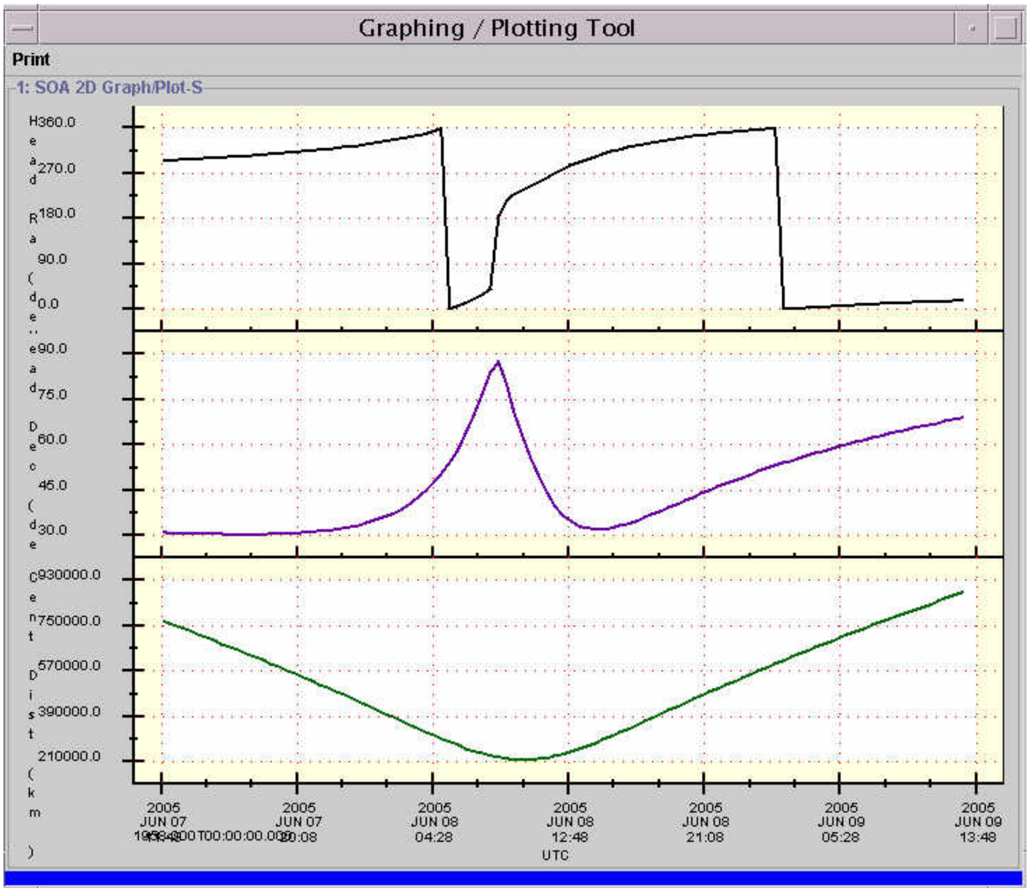

Figure 7-2. The Graphing/Plotting tool shows x/y data in a stacked graph.

8. COMMUNICATION

Step 7—
(Required) The user saves the observation for further recall and sends it to downstream software to be added to the plan of spacecraft activities.

At this point the observation has been designed and checked against both the user's requirements and against constraints. It meets the science objectives and is ready to be saved for further recall. SOA allows the scientist to save the observation in multiple formats. The choices are:

1. SOA format
2. Downstream software format that has been specified in a JPL interface control document.
3. Pointing information format

SOA can read and write all three formats. SOA can, of course, process its own file data. The downstream software format saves information to a file that downstream legacy software can change into commands. The pointing information format is a binary file that is widely used called a C-kernel. This file can be used to exchange information among scientists.

After the observation has been saved, it is time to add it to the plan of activities that are going to the spacecraft. SOA is one of the first software tools to be created with communication to downstream software as part of its design. SOA uses both interprocess communication and files to communicate with downstream planning and sequencing software. The user simply uses the Communications tab (Figure 8-1) to set-up an interprocess communications link with the planning software or creates a file that can be read by the legacy sequencing software.

This area requires the most adaptation. In order for the information to be sent to the downstream software, adaptation methods need to be written so that all the tools understand the data that is being sent. An SOA observation will correspond with a specific project "spacecraft activity" that the planning and sequencing software understands. A correspondence between an SOA observation and this project-specific spacecraft activity is created using the same hierarchical structure that SOA has used throughout for adaptation. This area is explained in detail in the Adapter's Guide.

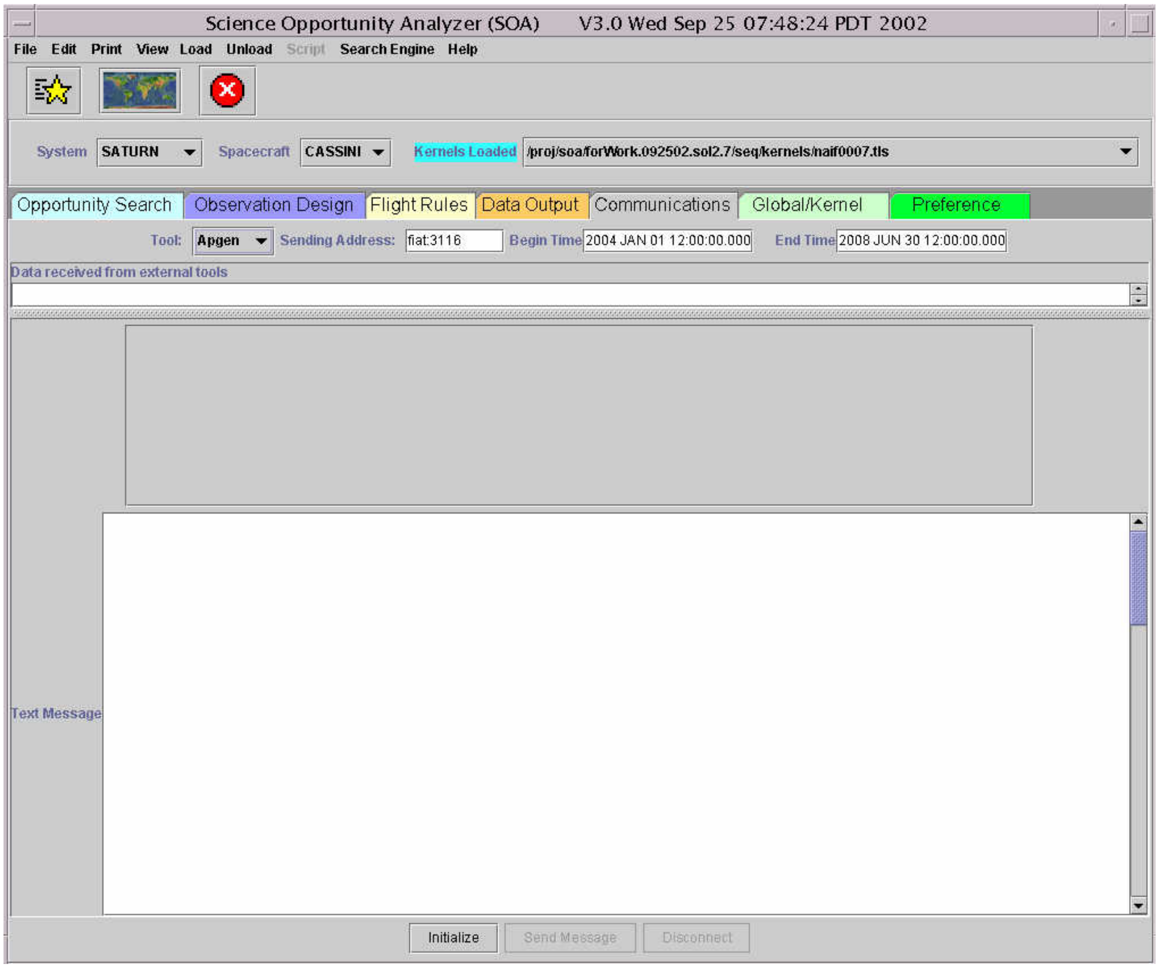

Figure 8-1. The Communications Tab display shows the communications connection any message being sent to downstream software using inter-process communications.

9. SUMMARY

SOA provides a full suite of tools to support scientists in building their observations. To recap the typical Science Observation Planning Process consists of the following steps:

1. The user runs an SOA configuration file to select the spacecraft and spacecraft trajectory data to be used.
2. The user finds one or more windows of opportunity for an observation based on geometry criteria using SOA's Opportunity Search capability.
3. The user designs the observation using the SOA Observation Design capability. At this point the user will often want to see Visualization displays. The user may even want to view an animation of the observation.
4. The user can also check to see if any constraints have been violated with this observation using the SOA Constraint Checking capability. The constraints are checked at the observation level against either geometric (i.e., Sun exclusion zones) or dynamic (i.e., spacecraft rates) rules.
5. At any point in this process the user can view associated data such as a phase angle using the Data Output capability. Data can be viewed in a tabular format, graphically or both.
6. The scientist saves the observation for future recall. Once the observation meets the science criteria and doesn't violate constraints, it can be sent downstream to planning software and command-level geometric checking software using SOA's communication capability.

SOA allows scientists to work in each of these areas emphasizing the data that are important to them and hiding details that are necessary to make the software work correctly, but are irrelevant to scientists.

In addition, each of the above areas of SOA can be adapted to meet the needs of a specific project. However, support in the software is provided so that adaptation can be minimal from project to project.

10. CONCLUSION

From the very beginning SOA was designed with the user in mind. Extensive surveys of the potential user community were conducted in order to develop the software requirements. Throughout the development period, close ties have been maintained with the science community to insure that the tool maintains its user focus. Although development is still in its early stages, SOA is already developing a user community on the Cassini project that is depending on this tool for their science planning. There are other tools at JPL that do various pieces of what SOA can do; however, there is no other tool which combines all these functions and presents them to the user in such a convenient, cohesive and easy to use fashion.

REFERENCES

[1] Carol Polanskey, Barbara Streiffert, Taifun O'Reilly, and Joshua Colwell, "Advances in Science Planning," *2002 AIAA Space Operations Conference Proceedings,* Oct. 8–11, 2002.

[2] Barbara Streiffert, Carol Polanskey, Taifun O'Reilly, and Joshua Colwell, "Science Opportunity Analyzer A Multi-Mission Tool for Planning," *2002 Core Technologies for Space Systems Conference Proceedings,* Nov. 19–21, 2002.

ACKNOWLEDGEMENTS

The work described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

We would like to thank the Cassini Project for their support.

Barbara Streiffert *is a Senior Systems and Software Engineer at Jet Propulsion Laboratory specializing in the development of innovative software approaches for use in ground data systems for spacecraft missions. She has worked in all aspects of systems and software development for commercial, military and aerospace projects. She has a B.S. in Applied Mathematics from Metropolitan State College, a B.A. in English from the University of Colorado and 20 years experience in software development.*

Carol Polanskey *is a Senior Science Systems Engineer and Technical Manager at Jet Propulsion Laboratory specializing in science planning for interplanetary spacecraft missions. In addition to her work on SOA, she has been a co-investigator on the Galileo project (an orbiting mission to Jupiter) where she and her team were responsible for science data acquisition and health of the magnetometer and dust detector (two of eleven) science instruments on the Galileo spacecraft. She has 14 years experience in spacecraft operations. She has a B.S. in Physics and Astronomy from Pennsylvania State University an M.S. in Geophysics from California Institute of Technology and a Ph.D. in Planetary Science from California Institute of Technology.*

Taifun O'Reilly *is a Senior Software Engineer at Jet Propulsion Laboratory specializing in new software solutions to existing ground data system problems. She has created the intuitive, easy to use graphical user interface for SOA as well as much of the object oriented hierarchical structure of SOA. She has worked in all aspects of software development for commercial, military and aerospace projects. She has a B.S. in Mathematics from California State University and over 20 years experience in software development.*

Josh Colwell *is a planetary scientist specializing in the origin, evolution, and dynamics of planetary rings, comets, and asteroids. He has flown two experiments on the space shuttle and has an ongoing microgravity research program on collisions in the solar system. He led the development of the Cassini Sequence Planner software (CASPER) for the Cassini mission to Saturn that served as a prototype for SOA. In addition to software consulting for JPL he teaches astronomy at the University of Colorado. He has a B.S. in Physics from Steston University and a Ph.D. in Astrophysical, Planetary, and Atmospheric Sciences from the University of Colorado, Boulder.*