CREATE YOUR OWN SCIENCE PLANNING TOOL IN 3 DAYS WITH SOA

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

ABSTRACT

Scientific discovery and advancement of knowledge has been, and continues to be, the goal for space missions at Jet Propulsion Laboratory. Scientists must plan their observations/experiments to get the maximum data return in order to make those discoveries. However, each mission has different science objectives, a different spacecraft and different instrument payloads, as well as, different routes to different destinations with different spacecraft restrictions and characteristics. In the current reduced cost environment, manageable cost for mission planning software is a must. Science Opportunity Analyzer (SOA), a planning tool for scientists and mission planners, utilizes a simple approach to reduce cost and promote reusability.

SOA, an object-oriented JAVA based tool that runs on Sun Solaris and PCs, has been built to allow users or projects to customize the software to their mission. The software has two elements – a project specific element and a core element (project-independent). Initially, a project or a user can perform the simple three-day adaptation and primarily use the core functionality. Building the project specific elements is done in three steps. The first step consists of obtaining and/or creating JPL navigation SPICE kernels. These kernels consist of planetary constants, spacecraft trajectory, planetary ephemeris, instrument field of view definition, and spacecraft/instrument coordinate frame definition. The development team can help with providing some of these kernels. The second step is to customize a configuration file. This file configures SOA to load in the project specific SPICE kernels as well as other core models. The optional last step is to create project specific geometric constraints (such as exclusion zones) using the SOA flight rule builder, a standard core feature. At this point the 3 Day project adaptation is complete and the user has an SOA that is specific to his/her mission.

Now the user has the ability to perform a set of tasks using five of the six major functional areas in SOA: Opportunity Search, Visualization, Observation Design, Constraint Checking, and Data Output. Opportunity Search allows scientists to search for interesting geometric opportunities including eclipses, flybys, etc. Visualization shows them a picture at the time of the opportunity or any other time of interest. Observation Design allows them to construct an actual observation at the time of the opportunity using SOA core observation types (i.e., mosaic, scan, roll scan, etc.), and then check for constraints, such as sun avoidance, with Constraint Checking. SOA also lets the scientist or planner check various aspects of the design using Data Output to obtain ancillary data. The data can be either plotted or presented in tabular form. This adaptation is sufficient for the early phases of a mission. Later in the mission after project specific spacecraft activities have been defined and the flight software has been built, SOA provides the flexibility to implement a more specialized project adaptation by adding these new elements to the software.

Now anyone can build his/her own sophisticated science-planning tool in a minimal amount of time at an extremely low cost with the Science Opportunity Analyzer.

1. INTRODUCTION

In the current reduced cost environment, manageable cost for mission planning software is a must. Science Opportunity Analyzer (SOA), a planning tool for scientists, utilizes a simple approach to reduce cost and promote reusability. SOA is a multi-mission Java-based software tool that runs on a Sun Solaris and a PC with either Windows or Linux. For a full description of SOA functionality see [4]. Since spacecraft missions are unique as far as destination, payloads, constraints and consumables, SOA is built so that it has two main divisions: the mission specific elements and the mission independent elements called "core". The 3 Day mission specific configuration requires a minimal set of mission specific elements. All of these elements are specified at

runtime. Currently, the SOA development staff has been creating these 3 Day mission specific versions of SOA at no charge.

2. THE 3 DAY SCIENCE PLANNER

The process to create any mission specific version is called adaptation. The required mission specific information includes spacecraft trajectory, celestial body information, instrument information, mission specific observations, spacecraft characteristics and spacecraft attitude control, and flight/mission rules. The core version of SOA contains the hooks for adding each of these types of data. So that a mission can use SOA in the early stages of the mission, the core contains general types of observations and a 3-axis stabilized turn model. When SOA is used for the early phases of the mission, these core elements plus the 3 Day adaptation are sufficient to produce enough information for studies and for trades. In later phases of missions more adaptation can be performed on SOA so that SOA replicates the mission characteristics more closely. With the 3 Day adaptation the early mission teams are able to perform their needed tasks easily.

3. NAVIGATION/CELESTIAL BODY DATA

The 3 Day configuration of SOA consists of obtaining and adding mission specific data in three main areas. First, the mission specific SPICE kernels must be obtained. SPICE stands for Spacecraft, Planets, Instrument, Constants and Events. The Navigation and Ancillary Information Facility (NAIF) at JPL often builds these files. See [1] for a description of the NAIF system. SOA requires at least one of each of the SPICE kernels except an events kernel. Spacecraft/planetary kernels are binary files that contain the spacecraft trajectory and the planetary positions. The Instrument kernel describes instrument apertures, for example, the size of a camera field of view. The Planetary Constants kernel has information about the planets or other celestial bodies such as rotation rate and size. In addition, SOA uses four more NAIF kernels: the Frames kernel, a miscellaneous text kernel, the leapseconds kernel and the spacecraft clock kernel. The Frames kernel provides information about the location of the instruments relative to the spacecraft. SOA's use of a miscellaneous text kernel is to provide the software with the spacecraft rate and acceleration maximum values. The leapseconds kernel contains information on the relationship between ephemeris time and universal time. Ephemeris time is a uniform time scale representing the independent variable of gravitational theories. Universal time is a time that is based on stellar motion as observed from Earth. The spacecraft clock kernel provides information on converting the spacecraft clock to universal time. The most challenging of the three steps is getting and setting up the SPICE kernels.

One of the 3 Day adaptations that the SOA development team has performed for a new mission named Dawn. The Dawn mission will visit two asteroids (Ceres and Vesta) that are thought to have existed since the beginnings of the solar system hence the project name "Dawn". The Dawn spacecraft has five instruments – a mapping spectrometer (mapspec), a gamma ray/neutron spectrometer (grns), a laser altimeter (lasalt), a magnetometer (mag) and a framing camera (ccd). The Dawn mission has spacecraft trajectory and celestial ephemeris kernels and those kernels have been provided to the SOA development team. The leapseconds kernel is a standard SPICE kernel that is used by everyone and is available to all projects. These kernels are placed in a directory that can be read by SOA. Next, the SOA development team must create or find the other kernels (planetary constants kernel, instrument kernel, instrument frame kernel and spacecraft clock kernel). In this case to get the needed data, SOA development team members have used various Dawn websites. In other cases, mission team members have provided the SOA team with the needed information. The planetary constants kernel defines Ceres and Vesta in a way that the information can be used by the various SPICE retrieval functions. The pole direction, rotation, size, names and id numbers are specified in this kernel. In the instrument kernel the various instrument fields of view (or apertures) are defined. Usually there is a separate instrument kernel for each instrument. The instrument field of view is defined by both size and shape. The frames kernel defines the coordinate frame of the instrument, and this coordinate frame defines the position of the instrument on the spacecraft. Finally, the spacecraft clock kernel is specified. If the project doesn't have a spacecraft clock kernel at this point in time, a pseudo clock kernel can be created and used until an actual spacecraft clock kernel is available. The spacecraft clock kernel defines the drift of the spacecraft clock and how the spacecraft clock divides time. Early in a mission this information is not crucial because time is not dealt with in terms of fractions of seconds.

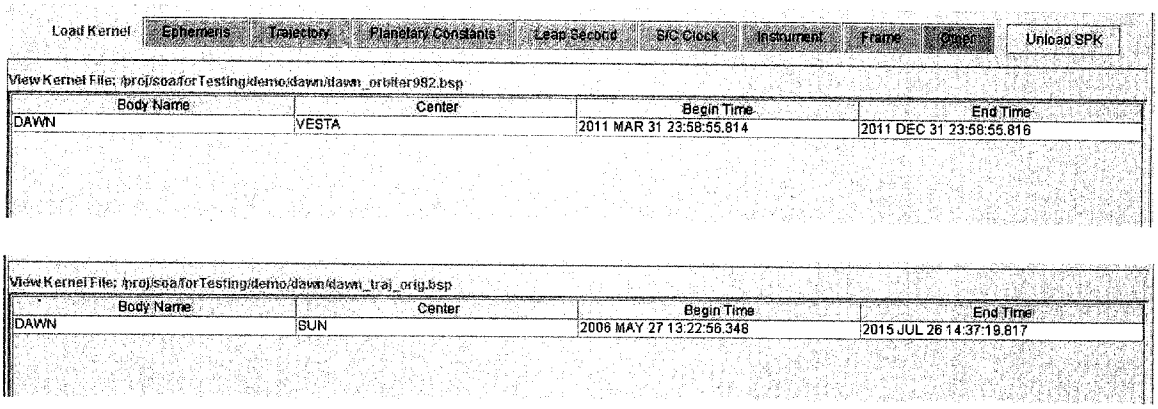

Figure 3.1 shows 2 different trajectories for the Dawn mission. One is centered on Vesta and the other is centered on the Sun.

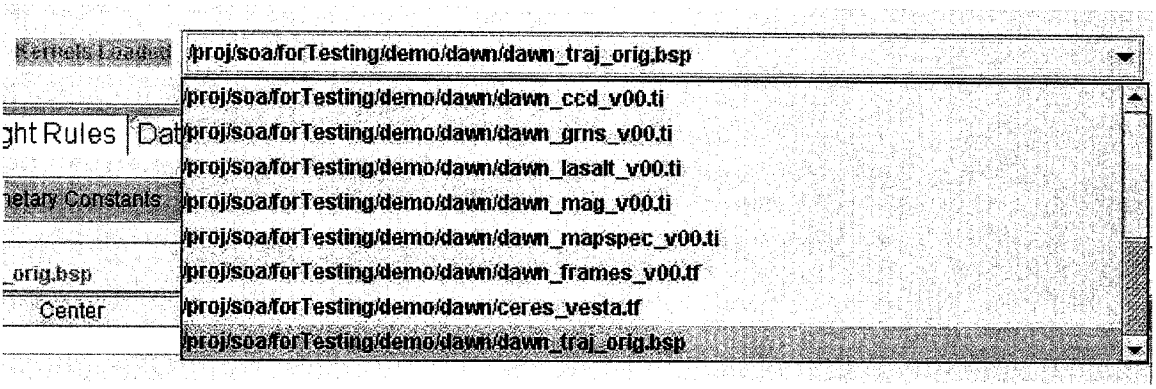

Figure 3.2 shows a list of the Dawn SPICE files

4. SOA CONFIGURATION FILE

The next step in adaptation is to provide the names and locations of the mission specific elements so that SOA can load them. SOA uses a configuration file for this purpose. This configuration file tells SOA the name of the spacecraft, the names and locations of the various SPICE files as well as other information used to configure the core SOA. Currently, the configuration file is written using Java Beans. The advantage of using Java Beans is that is very powerful and Java does the interpretation of the file. The disadvantage is that it looks like "code". For most projects the adapters of SOA create the mission specific configuration file and end users do not need to change it. In the case of the 3 Day adaptation the SOA development team creates the configuration file and the end users will not need to change it until the mission decides to do more adaptation of SOA.

For Dawn the configuration file has been created to specify the spacecraft name as "DAWN". All of the relevant SPICE kernels are entered into the file by their full path names. The start and end times of the mission are entered into the configuration file. This time span should be reflected in the time span covered by spacecraft trajectory kernel and the planetary ephemeris kernel. The geometric data to be used to search for observation opportunities, the core observation types and the core constraint types have been entered. Now the configuration file is ready to be loaded into SOA.

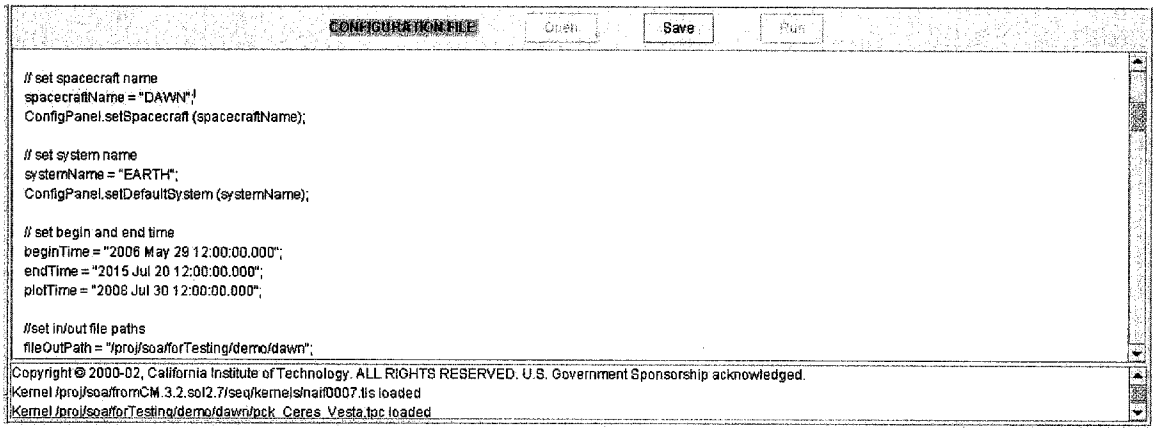

Figure 4-1 shows the Dawn Configuration File. The spacecraft name is set to be Dawn. The time span is set to the time span of the Sun-centered trajectory.

5. CONSTRAINTS

The final adaptation step is to optionally create mission specific constraints. This step is optional because checking constraints is not a required element in SOA. It is a user-selected option. Some spacecraft constraints are usually known even early in a mission. Often sensitive instruments or their heat-reducing radiators can't be in view of the sun under certain conditions. Sometimes an entire side of the spacecraft can't be illuminated. SOA has the capability to check that selected previously defined spacecraft hardware or a selected spacecraft axis is not in a specified exclusion zone. In order to perform this check, the SOA development team creates a rule that identifies the exclusion zone and the spacecraft hardware item using the Flight Rule Builder portion of SOA. The builder uses Java's drag-and-drop functionality and a list of exclusion zone types. It is very easy to use. Even end users can perform this portion of the adaptation. It is fully described in the SOA User's Guide, see [5].

The Dawn mission has a framing camera. Most camera lenses are sensitive to sunlight. For this mission the SOA development team created a rule stating that the camera cannot be within an angle of 20 degrees of the Sun. When the end user selects constraint checking, this rule will be activated whenever the camera field of view (aperture) of the camera is in that exclusion zone. The rule has been made for Dawn users as an example. Dawn has not provided the SOA development team with an actual exclusion zone to check.

6. SOA FUNCTIONS

At this point SOA is ready to run for the Dawn mission. The end user can use five of the six major functional items: Opportunity Search, Visualization, Observation Design, Flight Rules and Data Output. Opportunity Search allows the end user to check for geometric opportunities. The user can check for the times when the spacecraft is within a certain distance of either Ceres or Vesta as well as for 16 other types of geometric opportunities. It is important to make sure that the trajectory file that is being used supports the search that is being made. The spacecraft trajectory kernel must be consistent with the planetary ephemeris kernel that is used. SOA also has the idea of "epoch" times. Epochs tie a specific time to a user-specified symbol name. These times are generally tied to a specific event such as launch or closest approach to a body. The idea behind using the symbolic name is that if the time of the event changes, the new time only needs to be changed one place. Once the user finds an opportunity, visualization will show the user a picture of the celestial bodies at that time. Next the user can try out sample observations using Observation Design. At this point, one of the core SOA observation types can be selected. SOA has a "scoping" level observation that simply shows the celestial geometry and a single sample field of view (aperture). More complex types such as a mosaic, a scan or roll scan are also provided. Next the user can check for constraint violations using SOA. It is even possible to rotate the spacecraft around an axis to try to locate a constraint free zone. Finally, Data Output provides graphing capabilities as well as tabular data output. This data can be related to opportunity search, the spacecraft trajectory or an observation. The most important aspect is that SOA provides multiple ways of looking at the same data.

The following series of figures shows how the Dawn mission can use each of the functional areas of SOA after a 3 Day adaptation. For Dawn two sample search queries have been created. One search query looks for the closest approach to Vesta and the other one looks for all times when there is a local minimum in distance from the spacecraft to Vesta. Once the results from the search have been returned, the user can now look at a display of Vesta

at that time using SOA Visualization. Visualization has four different views: 3-D perspective projection, 3-D arbitrary observer, 2-D sky map and 2-D trajectory plot. The 3-D perspective projection view displays the target as seen from the spacecraft. The 3-D arbitrary observer view displays the target as seen from an arbitrary point in space. The 2-D sky map view is a 2-D equidistant map projection of the solar system. The 2-D trajectory plot displays a projection of the spacecraft path around a target as viewed from either the equatorial or the ecliptic plane. Next several observations have been created to give planners and scientists an idea of what they would be able to see from this spacecraft attitude. At this point the user can check the constraint that was built as a sample. Finally, Data Output was used to compare two of the trajectories that have been created for orbiting Vesta. The plots compare the spacecraft distance to the center of Vesta and the spacecraft altitude above Vesta. The SOA development team has used this 3 Day adaptation for an actual demonstration of SOAs capabilities to Dawn team members. Also, similar adaptations have been created for the Deep Impact and Mars Reconnaissance Orbiter missions.

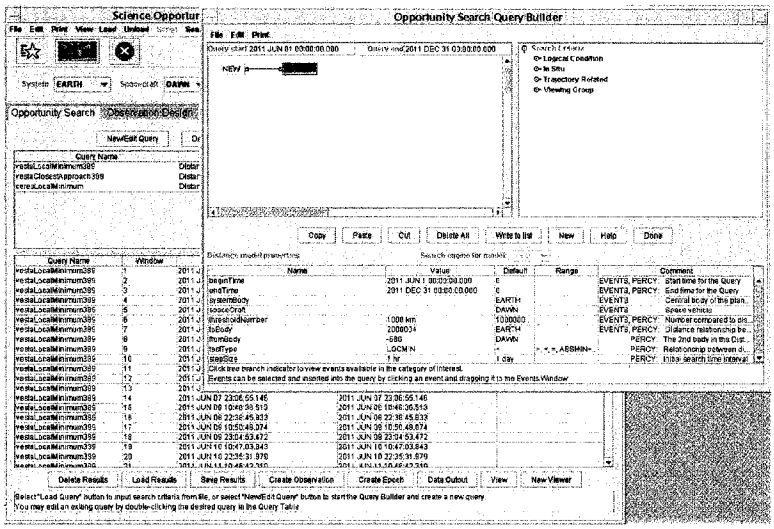

Figure 6.1 shows the opportunity search builder window with a local minimum distance search query and its associated parameters. Behind the builder window is the opportunity search display showing the results of that search. The highlighted result is a time that has been used to build the observation in Figure 6.3

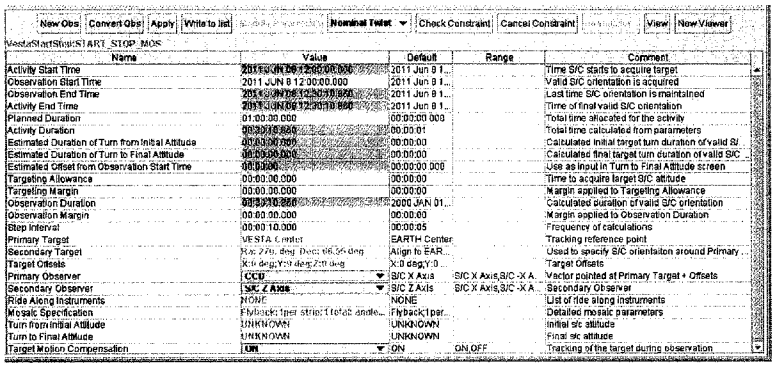

Figure 6.2 shows the observation design parameters that have been used for the observation in Figure 6.3

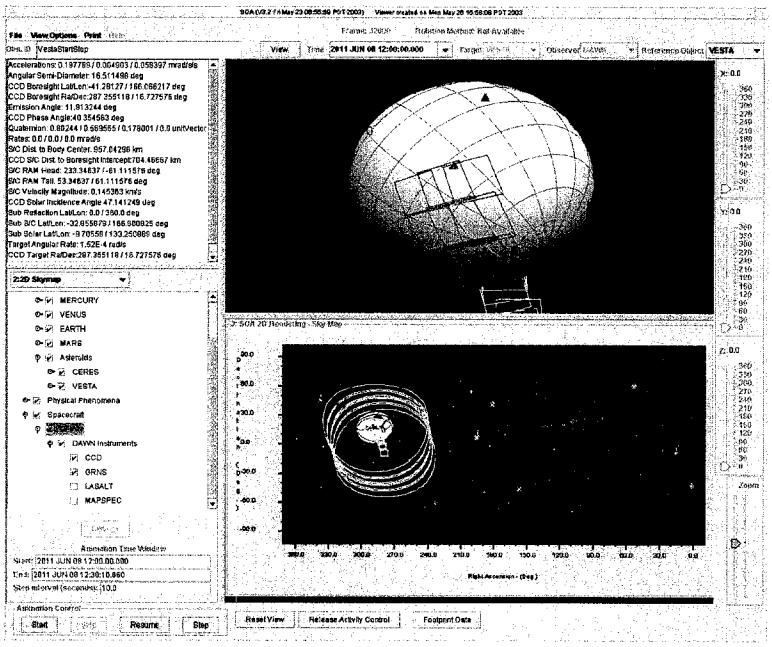

Figure 6.3 shows two views (a 3-D perspective projection and a 2-D sky map) of one of the local minimums discovered using opportunity search. This time was converted to a mosaic observation that shows the camera and the gamma ray/neutron spectrometer fields of view

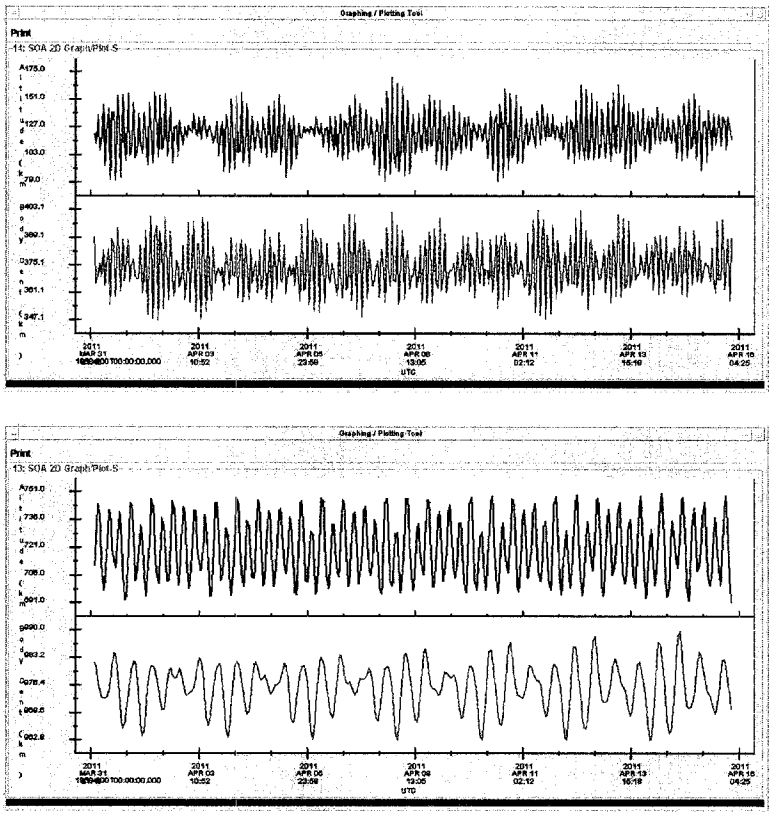

Figure 6.4 shows the Data Output plots for the spacecraft altitude and the spacecraft distance to the center of Vesta for two different trajectories. Mission planners can use these types of comparisons to evaluate candidate trajectories.

.

7. MISSION PHASE

The 3 Day configuration or adaptation of SOA to a particular mission is especially useful during mission definition and formulation phases, called Phases A and B, of the mission life cycle although the functionality provided with this adaptation is needed during other phases as well. Phase A provides enough of a design to be able to cost the mission. Phase B continues the design work to assure that cost estimates continue to be achievable. During these phases of the mission studies are performed to help determine the most optimal trajectory, to evaluate the achievability of science goals, to evaluate geometric constraints on the spacecraft such as areas to avoid, to evaluate potential science return, and always to determine costs. Based on these studies trades are made. These trades are made for a variety of reasons. Some trades are to reduce cost. Some are to increase science return. Some are to make sure that science goals are met. Most often the trades involve multiple factors. It is important during these phases that mission planners and scientists are provided an easy-to-use software tool to help them determine which trades to make. The additional stipulation is that the tool has to be low cost. Generally, these phases of the mission have a small staff and a small budget. SOA has all of the needed functionality and saves costs as well.

8. COST SAVINGS

In the past Phase A and B software tools have been built that generally expect the end user to be knowledgeable of navigation and orbital mechanics. These tools often are difficult for scientists and mission planners to use and so ad hoc tools, pencil and paper or other less than optimal ways of making trades have been used. In this environment it takes longer to generate the data and often, less data are used as the basis for the trade. SOA (at no cost to the mission) promotes a cost effective way of giving mission planners and scientists the data they need to make trades. One mission planner said that SOA gave him more ways of visualizing trajectories than other tools. Another one said that SOA enables him to find a trajectory that minimizes a rather formidable Sun constraint. Several scientists have indicated that presenting data in multiple ways allows them to find observations that will "work". After running SOA one engineer said that the work that took him several days to complete, he was able do in hours using SOA. As a mission moves into other phases, further adaptation can be performed to include project specific activities, formal flight and mission rules, and a higher fidelity model of spacecraft turning capability. SOA, at no initial cost, is an extremely cost effective tool that provides significant functionality and extensive information at no initial cost.

9. CONCLUSION

Now in a minimal amount of time at an extremely low cost (free), any mission can have its own sophisticated planning tool -- the Science Opportunity Analyzer.

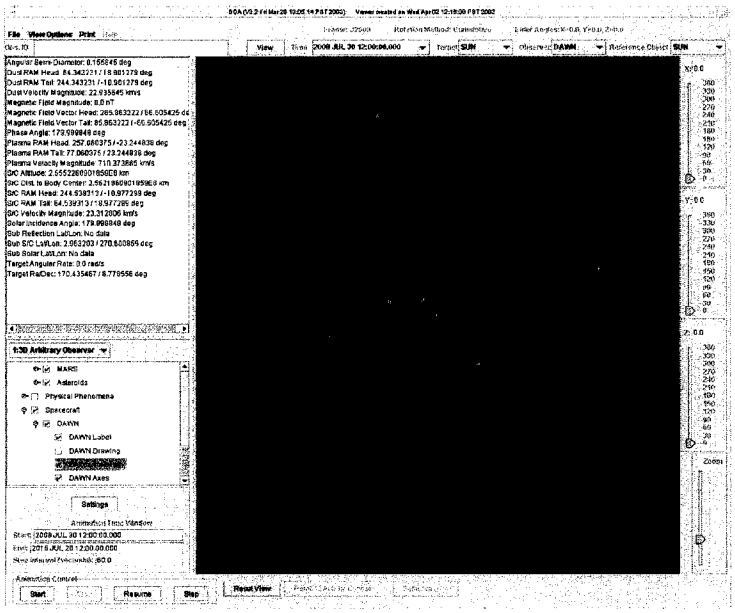

Figure 9.1 shows the trajectory of Dawn and all of the planets around the Sun. It closely resembles the drawing that is on one of the Dawn websites.

10. REFERENCES

[1] Charles Acton, Nat Bachman, Lee Elson, Boris Semenov and Ed Wright, "SPICE: A Real Example of Data System Re-Use to Reduce the Costs of Ground Data Systems Development and Mission Operations," *5th International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations,* July 8–11, 2003.

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

[3] 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.

[4] Barbara Streiffert, Carol Polanskey, Taifun O'Reilly, and Joshua Colwell, "Science Opportunity Analyzer -- A Multi-Mission Approach to Science Planning," *2002 IEEE Aerospace Conference Proceedings,* Mar. 8–15, 2003.

[5] Joshua Colwell, "Science Opportunity Analyzer User's Guide V3.0" Jet Propulsion DSMS Mission Services and Applications Library, No. 887-000059, Feb. 17, 2003

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 Dawn Project for their support.