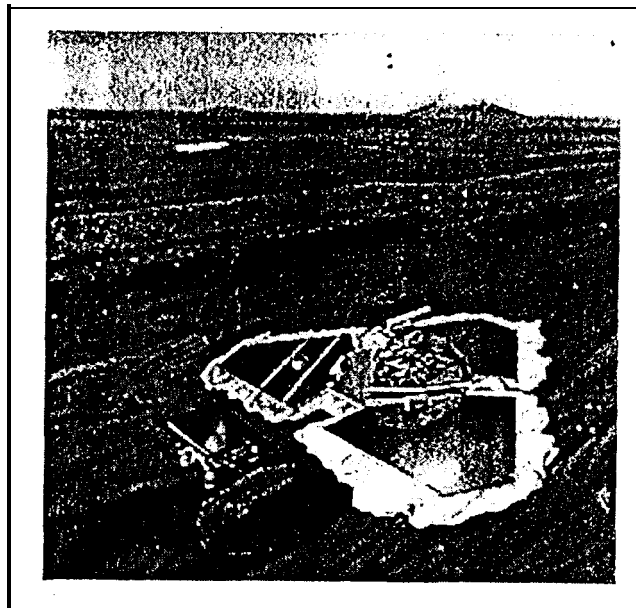


Title: Solar System Calculator: A Mission-independent Tool for Science and Mission Operation

Author: Jacqueline Kim
Science Integration Engineer
Mars Pathfinder Project, GDS
818-393-0780
jackie@delta3.jpl.nasa.gov

Affiliation: ^{CI-1 /} Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109

Contact: Jacqueline Kim



Conceptualization of Lander on Surface of Mars

ABSTRACT

The Solar System Calculator (SSC) is a portable, multi-mission tool designed to compute mission-critical observation geometry such as directions, ranges and lighting angles. SSC is being developed in support of NASA's Mars Pathfinder Project. Pathfinder is the first Discovery-class mission implementing the "cheaper, faster, and better" approach.

In order to successfully develop this mission, a paradigm shift was necessary, both for scientists and developers of the ground system. Instead of the traditional approach of redundancy and specialization, the main focus was automation and integration. Also, by imbedding re-usability in the design, software is no longer mission-dependent, or as costly in the long run.

To provide an efficient operation system to the Pathfinder science teams, the Ground Data System team identified core functional areas required for planning, operating and analyzing the mission. One important area identified is providing ancillary planetary information. Currently, all ancillary information is provided in the form of SPICE kernels, as defined by the NAIF (Navigation Ancillary Information Facility) group at JPL. Although SPICE kernels are easy to use, and NAIF provides adequate user-support, SSC can be used to improve SPICE kernel's usability by eliminating redundant programming. Often, several science teams duplicate their programming efforts to get the same information, such as planetary quantities or spacecraft related parameters. On any project, it is common to find several engineers generating independent programs to access similar data, mostly related to Navigation information. The SSC can help reduce this redundancy.

When completed, the SSC is expected to consist of 3 independent modules: Generator, Plotter and Mapper. Generator performs calculations and outputs the parameters. Plotter takes the parameters and plots in a specified coordinate system when a user request it. Mapper can overlay necessary information on images such as maps of the surface of Mars. Each module is functionally separated for efficiency. Other output functions could also be added, such as a spread sheet or database.

The SSC is mission-independent software, re-usable by any mission or task. As long as SPICE kernel files are defined, SSC can manipulate the necessary information to calculate a large assortment of scientific quantities. It reads SPICE kernels and

interprets NAIF Tool kit subroutines, thus bypassing normal programming. The SSC has two interfaces: command line and GUI. The command line version is easily ported to any platform which supports ANSI FORTRAN 77 and C. The GUI version requires MOTIF.

The prototype of SSC was recently tested and received favorable reviews from science and **mission operations teams. For Pathfinder**, the SSC not only serves the science community, but also will be used to solve several problems in navigation and mission operations. Further, the ground system team and NAIF group are investigating a way to incorporate the SSC into the existing uplink system. The demand for this software is growing, and many projects find the SSC to be a suitable solution for their problems.

The new era of cheaper, faster and better missions requires reusable, mission-independent software rather than traditional, mission-specific programs. Of course, the initial development of reusable software is difficult. However, the SSC program demonstrated that it is possible to develop high quality, mission-independent and reusable software at a reasonable cost.

This paper describes the design approach and characteristics of SSC now being developed by NAIF, and includes example displays of the prototype.

(This abstract has been submitted to the Computing In Aerospace 1.0 Conference, 1995.)