SHINE Virtual Machine Model for In-flight Updates of Critical Mission Software

This software is a new target for the Spacecraft Health Inference Engine (SHINE) knowledge base that compiles a knowledge base to a language called Tiny C — an interpreted version of C that can be embedded on flight processors. This new target allows portions of a running SHINE knowledge base to be updated on a “live” system without needing to halt and restart the containing SHINE application. This enhancement will directly provide this capability without the risk of software validation problems and can also enable complete integration of BEAM and SHINE into a single application.

This innovation enables SHINE deployment in domains where autonomy is used during flight-critical applications that require updates. This capability eliminates the need for halting the application and performing potentially serious total system uploads before resuming the application with the loss of system integrity. This software enables additional applications at JPL (microsensors, embedded mission hardware) and increases the marketability of these applications outside of JPL.

This work was done by Mark James, Ryan Mackey, and Raffi Tikidjian of Caltech for NASA’s Jet Propulsion Laboratory. Further information is contained in a TSP (see page1).

In accordance with Public Law 96-517, the contractor has elected to retain title to this invention. Inquiries concerning rights for its commercial use should be addressed to:

Innovative Technology Assets Management
JPL
Mail Stop 202-233
4800 Oak Grove Drive
Pasadena, CA 91109-8099
E-mail: iaoffice@jpl.nasa.gov
Refer to NPO-45959, volume and number of this NASA Tech Briefs issue, and the page number.

Mars Image Collection Mosaic Builder

A computer program assembles images from the Mars Global Surveyor (MGS) Mars Observer Camera Narrow Angle (MOCNA) collection to generate a uniform-high-resolution, georeferenced, uncontrolled mosaic image of the Martian surface. At the time of reporting the information for this article, the mosaic covered 7 percent of the Martian surface and contained data from more than 50,000 source images acquired under various light conditions at various resolutions.

The program geolocates, reprojects, and blends one source image at a time onto the mosaic. Geolocation and reprojection involve the use of a second-order polynomial based on coordinates of the source-image footprints. Images are stacked in the order of increasing resolution — higher-resolution images on top of lower-resolution images. The stacking order is also partly determined by the order of adding the source images to the mosaic. The mosaic-image data are stored in a custom file format that accommodates regional tiles and supports explicit representation of empty areas, image-data compression, and representation of localized changes.

This program was written by Lucian Plesea of Caltech and Trent Hare of the United States Geological Survey for NASA’s Jet Propulsion Laboratory.

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-45960.

Providing Internet Access to High-Resolution Mars Images

The OnMars server offers access to most of the available high-resolution Martian image and elevation data, including an 8-meter-per-pixel uncontrolled mosaic of most of the Mars Global Surveyor (MGS) Mars Observer Camera Narrow Angle (MOCNA) image collection, which is not available elsewhere. This server can generate image and map files in the tagged image file format (TIFF), Joint Photographic Experts Group (JPEG), 8- or 16-bit Portable Network Graphics (PNG), or Keyhole Markup Language (KML) format. Image control is provided by use of the OGC Style Layer Descriptor (SLD) protocol. The OnMars server also implements tiled WMS protocol and super-overlay KML for high-performance client application programs.

This program was written by Lucian Plesea of Caltech and Trent Hare of the United States Geological Survey for NASA’s Jet Propulsion Laboratory.

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-45959.

Providing Internet Access to High-Resolution Lunar Images

The OnMoon server is a computer program that provides Internet access to high-resolution Lunar images, maps, and elevation data, all suitable for use in geographical information system (GIS) software for generating images, maps, and computational models of the Moon. The OnMoon server implements the Open Geospatial Consortium (OGC) Web Map Service (WMS) server protocol and supports Moon-specific extensions. Unlike other Internet map servers that provide Lunar data using an Earth coordinate system, the OnMoon server supports encoding of data in Moon-specific coordinate systems.

The OnMoon server offers access to most of the available high-resolution Lunar image and elevation data. This server can generate image and map files in the tagged image file format (TIFF) or the Joint Photographic Experts Group (JPEG), 8- or 16-bit Portable Network Graphics (PNG), or Keyhole Markup Language (KML) format. Image control is provided
by use of the OGC Style Layer Descriptor (SLD) protocol. Full-precision spectral
arithmetic processing is also available, by use of a custom SLD extension. This server
can dynamically add shaded relief based on the Lunar elevation to any image layer.
This server also implements tiled WMS protocol and super-overlay KML for high-
performance client application programs.
This program was written by Lucian Plesa of Caltech and Trent Hare of the United States
Geological Survey for NASA's Jet Propulsion Laboratory.
This software is available for commercial li-
censing. Please contact Karina Edmonds of the California Institute of Technology at (626)
395-2322. Refer to NPO-45052.

Virtual Satellite
Virtual Satellite (VirtualSat) is a computer program that creates an environment
that facilitates the development, verification, and validation of flight soft-
ware for a single spacecraft or for multiple spacecraft flying in formation. In this
environment, enhanced functionality and autonomy of navigation, guidance, and
control systems of a spacecraft are provided by a virtual satellite — that is, a
computational model that simulates the dynamic behavior of the spacecraft.
Within this environment, it is possible to execute any associated software, the
development of which could benefit from knowledge of, and possible inter-
action (typically, exchange of data) with, the virtual satellite. Examples of as-
associated software include programs for simulating spacecraft power and ther-
mal-management systems. This environment is independent of the flight hard-
ware that will eventually host the flight software, making it possible to develop
the software simultaneously with, or even before, the hardware is delivered.
Optionally, by use of interfaces included in VirtualSat, hardware can be used in-
stead of simulated. The flight software, coded in the C or C++ programming
language, is compilable and loadable into VirtualSat without any special mod-
fications. Thus, VirtualSat can serve as a relatively inexpensive software test-bed
for development test, integration, and post-launch maintenance of spacecraft
flight software.
This program was written by Stephan R. Hammers of the Hammers Co., Inc. for God-
dard Space Flight Center. Further information is contained in a TSP (see page 1). GSC-
14824-1

Small-Body Extensions for
the Satellite Orbit Analysis
Program (SOAP)

An extension to the SOAP software al-
low users to work with tri-axial ellipsoid-
based representations of planetary bod-
ies, primarily for working with small
natural satellites, asteroids, and comets.
SOAP is a widely used tool for the visual-
ization and analysis of space missions. The
small body extension provides the same
visualization and analysis constructs for
use with small bodies. These constructs
allow the user to characterize satellite
path and instrument cover information for
small bodies in both 3D display and
numerical output formats.
Tri-axial ellipsoids are geometric
shapes the diameters of which are differ-
ent in each of three principal x, y, and z
dimensions. This construct provides a
better approximation than using
spheres or oblate spheroids (ellipsoids
comprising two common equatorial di-
ameters as a distinct polar diameter).
However, the tri-axial ellipsoid is consid-
ernably more difficult to work with from a
modeling perspective. In addition, the
SOAP small-body extensions allow the
user to actually employ a plate model for
highly irregular surfaces. Both tri-axial
ellipsoids and plate models can be as-
signed to coordinate frames, thus allow-
ing for the modeling of arbitrary changes to body orientation.
A variety of features have been ex-
tended to support tri-axial ellipsoids, in-
cluding the computation and display of
the spacecraft sub-orbital point, ground
trace, instrument footprints, and
swathes. Displays of 3D instrument vol-
umes can be shown interacting with the
ellipsoids. Longitude/ latitude grids, con-
tour plots, and texture maps can be
displayed on the ellipsoids using a vari-
ety of projections. The distance along an
arbitrary line of sight can be computed
between the spacecraft and the ellipsoid,
and the coordinates of that intersection
can be plotted as a function of time.
The small-body extension supports the same
visual and analytical constructs that are
supported for spheres and oblate sphero-
ids in SOAP making the implementa-
tion of the more complex algorithms
largely transparent to the user.
This work was done by Robert Carnright of
Caltech and David Stodden and John Coggi
of The Aerospace Corporation for NASA's Jet
Propulsion Laboratory.
This software is available for commercial li-
censing. Please contact Karina Edmonds of the California Institute of Technology at (626)
395-2322. Refer to NPO-45054.

Scripting Module for
the Satellite Orbit Analysis
Program (SOAP)

This add-on module to the SOAP soft-
ware can perform changes to simulation
objects based on the occurrence of spe-
cific conditions. This allows the software
to encompass simulation response of