

Onboard Short Term Plan Viewer

Lyndon B. Johnson Space Center, Houston, Texas

Onboard Short Term Plan Viewer (OSTPV) is a computer program for electronic display of mission plans and timelines, both aboard the International Space Station (ISS) and in ISS ground control stations located in several countries. OSTPV was specifically designed both (1) for use within the limited ISS computing environment and (2) to be compatible with computers used in ground control stations. OSTPV supplants a prior system in which, aboard the ISS, timelines were printed on paper and incorporated into

files that also contained other paper documents. Hence, the introduction of OSTPV has both reduced the consumption of resources and saved time in updating plans and timelines. OSTPV accepts, as input, the mission timeline output of a legacy, print-oriented, UNIX-based program called "Consolidated Planning System" and converts the timeline information for display in an interactive, dynamic, Windows Web-based graphical user interface that is used by both the ISS crew and ground control teams in real time. OSTPV en-

ables the ISS crew to electronically indicate execution of timeline steps, launch electronic procedures, and efficiently report to ground control teams on the statuses of ISS activities, all by use of laptop computers aboard the ISS.

This work was done by Tim Hall and Troy LeBlanc of Johnson Space Center and Brian Ulman, Aaron McDonald, Paul Gramm, Li-Min Chang, Suman Keerthi, Dov Kivlovitz, and Jason Hadlock of United Space Alliance. Further information is contained in a TSP (see page 1). MSC-24335-1

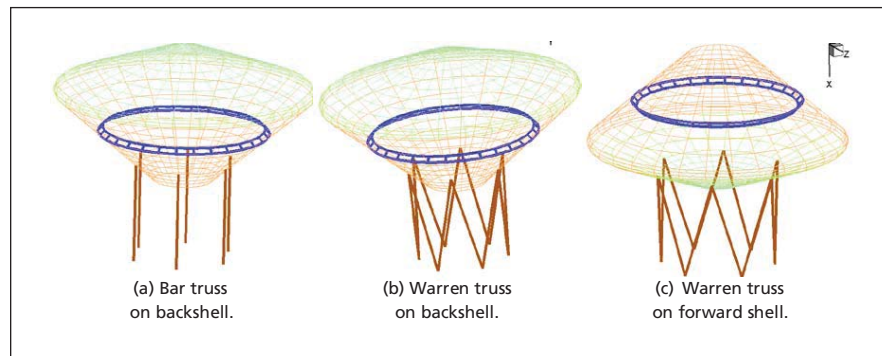
Multidisciplinary Tool for Systems Analysis of Planetary Entry, Descent, and Landing

Langley Research Center, Hampton, Virginia

Systems analysis of a planetary entry (SAPE), descent, and landing (EDL) is a multidisciplinary activity in nature. SAPE improves the performance of the systems analysis team by automating and streamlining the process, and this improvement can reduce the errors that stem from manual data transfer among discipline experts.

SAPE is a multidisciplinary tool for systems analysis of planetary EDL for Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune, and Titan. It performs EDL systems analysis for any planet, operates cross-platform (i.e., Windows, Mac, and Linux operating systems), uses existing software components and open-source software to avoid software licensing issues, performs low-fidelity systems analysis in one hour on a computer that is comparable to an average laptop, and keeps discipline experts in the analysis loop.

SAPE uses Python, a platform-independent, open-source language, for inte-



SAPE is capable of analyzing and sizing certain classes of planetary probes, as demonstrated by sample structural topologies for Pathfinder probe.

gration and for the user interface. Development has relied heavily on the object-oriented programming capabilities that are available in Python. Modules are provided to interface with commercial and government off-the-shelf software components (e.g., thermal protection systems and finite-element analysis). SAPE cur-

rently includes the following analysis modules: geometry, trajectory, aerodynamics, aerothermal, thermal protection system, and interface for structural sizing.

This work was done by Jamshid A. Samareh of Langley Research Center. Further information is contained in a TSP (see page 1). LAR-17821-1

Bundle Security Protocol for ION

NASA's Jet Propulsion Laboratory, Pasadena, California

This software implements bundle authentication, conforming to the Delay-Tolerant Networking (DTN) Internet Draft on Bundle Security Protocol (BSP), for the Interplanetary Overlay Network (ION) implementation of

DTN. This is the only implementation of BSP that is integrated with ION.

The bundle protocol is used in DTNs that overlay multiple networks, some of which may be challenged by limitations such as intermittent and possibly unre-

dictable loss of connectivity, long or variable delay, asymmetric data rates, and high error rates. The purpose of the bundle protocol is to support interoperability across such stressed networks. The bundle protocol is layered on top of a

“convergence layer” of adapters that encapsulate bundles in the protocol data units (PDUs) of the underlying networks’ native protocols for transmission and also extract bundles from the PDUs of those protocols as they are received. This convergence-layer encapsulation enables an application in one network to communicate with an application in another network built on entirely different native protocols, both of which are spanned by the DTN.

Security will be important for the bundle protocol. The stressed environment of the underlying networks over which the bundle protocol will operate makes it important that the DTN be protected from unauthorized use, and this stressed

environment poses unique challenges on the mechanisms needed to secure the bundle protocol. Furthermore, DTNs may very likely be deployed in environments where a portion of the network might become compromised, posing the usual security challenges related to confidentiality, integrity, and availability.

The BSP encompasses four mechanisms that are designed to provide this security. The technology currently being reported implements one of those mechanisms, the Bundle Authentication Block (BAB), and provides a framework for implementation of the remaining mechanisms: Payload Integrity Block, Payload Confidentiality Block, and Extension Security Block.

The ION system runs on Linux, OS/X, Solaris, FreeBSD, RTEMS, and VxWorks, and it should port readily to other POSIX-based operating systems. No special hardware is required. RAM (random access memory) requirements depend on the volume of DTN traffic that must be handled.

This work was done by Scott C. Burleigh of Caltech and Edward J. Birrane and Christopher Krupiarz of the Johns Hopkins University Applied Physics Laboratory for NASA’s Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47211.

Visual PEF Reader — VIPER

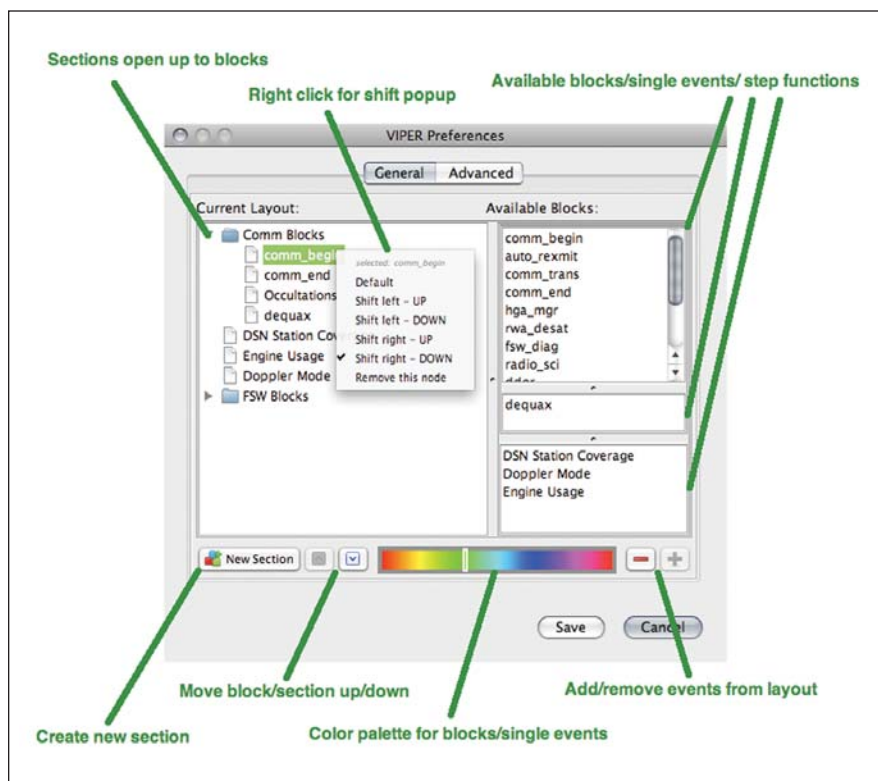
NASA’s Jet Propulsion Laboratory, Pasadena, California

This software graphically displays all pertinent information from a Predicted Events File (PEF) using the Java Swing framework, which allows for multi-platform support. The PEF is hard to weed through when looking for specific information and it is a desire for the MRO (Mars Reconnaissance Orbiter) Mission Planning & Sequencing Team (MPST) to have a different way to visualize the data. This tool will provide the team with a visual way of reviewing and error-checking the sequence product.

The front end of the tool contains much of the aesthetically appealing material for viewing. The time stamp is displayed in the top left corner, and highlighted details are displayed in the bottom left corner. The time bar stretches along the top of the window, and the rest of the space is allotted for blocks and step functions. A preferences window is used to control the layout of the sections along with the ability to choose color and size of the blocks.

Double-clicking on a block will show information contained within the block. Zooming into a certain level will graphically display that information as an overlay on the block itself. Other functions include using hotkeys to navigate, an option to jump to a specific time, enabling a vertical line, and double-clicking to zoom in/out.

The back end involves a configuration file that allows a more experienced user to pre-define the structure of a block, a single event, or a step function. The individual will have to



VIPER GUI general preferences.

determine what information is important within each block and what actually defines the beginning and end of a block. This gives the user much more flexibility in terms of what the tool is searching for. In addition to the configurability, all the settings in the preferences window are saved in the configuration file as well.

This work was done by Victor Luo, Teerapat Khanampornpan, Rudy A. Boehmer, and Rachel Y. Kim of Caltech for NASA’s Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47509.