

ponents that identify anomalies or that search for past instances of similar behavior.

This software supports NASA's Software, Intelligent Systems, and Modeling element in the Exploration Systems Research and Technology Program by augmenting the capability of human flight controllers to make correct decisions, thus increasing safety and reliability. It was designed specifically as a tool for NASA's flight controllers to monitor the International Space Station and a future Crew Exploration Vehicle.

This program was written by Joseph Jacob, Michael Turmon, Timothy Stough, and Herbert Siegel of Caltech and Patrick Walter and Cindy Kurt of United Space Alliance 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 Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-43952.

➤ Mars Reconnaissance Orbiter Uplink Analysis Tool

This software analyzes Mars Reconnaissance Orbiter (MRO) orbital geometry with respect to Mars Exploration Rover (MER) contact windows, and is the first tool of its kind designed specifically to support MRO-MER interface coordination. Prior to this automated tool, this analysis was done manually with Excel and the UNIX command line. In total, the process would take approximately 30 minutes for each analysis. The current automated analysis takes less than 30 seconds.

This tool resides on the flight machine and uses a PHP interface that does the entire analysis of the input files and takes into account one-way light time from another input file. Input files are copied over to the proper directories and are dynamically read into the tool's interface. The user can then choose the corresponding input files based on the time frame desired for analysis. After submission of the Web form, the tool merges the two files into a single, time-ordered listing of events for both spacecraft. The times are converted to the same reference time (Earth Transmit Time) by reading in a light time file and performing the calculations necessary to shift the time formats. The program also has the ability to vary the size of the keep-out window on the main page of the analysis tool by inputting a custom time for padding each MRO event time. The parameters on the form are read in

and passed to the second page for analysis. Everything is fully coded in PHP and can be accessed by anyone with access to the machine via Web page.

This uplink tool will continue to be used for the duration of the MER mission's needs for X-band uplinks. Future missions also can use the tools to check overflight times as well as potential site observation times. Adaptation of the input files to the proper format, and the window keep-out times, would allow for other analyses. Any operations task that uses the idea of keep-out windows will have a use for this program.

This program was written by Teerapat Khanampornpan, Roy Gladden, Forest Fisher, and Pauline Hwang of Caltech 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-44222.

➤ Problem Reporting System

The Problem Reporting System (PRS) is a Web application, running on two Web servers (load-balanced) and two database servers (RAID-5), which establishes a system for submission, editing, and sharing of reports to manage risk assessment of anomalies identified in NASA's flight projects. PRS consolidates diverse anomaly-reporting systems, maintains a rich database set, and incorporates a robust engine, which allows tracking of any hardware, software, or paper process by configuring an appropriate life cycle. Global and specific project administration and setup tools allow lifecycle tailoring, along with customizable controls for user, e-mail, notifications, and more. PRS is accessible via the World Wide Web for authorized user at most any location.

Upon successful log-in, the user receives a customizable window, which displays time-critical "To Do" items (anomalies requiring the user's input before the system moves the anomaly to the next phase of the lifecycle), anomalies originated by the user, anomalies the user has addressed, and custom queries that can be saved for future use. Access controls exist depending on a user's role as system administrator, project administrator, user, or developer, and then, further by association with user, project, subsystem, company, or item with provisions for business-to-business exclusions, limitations on access according to the covert or overt nature of a given project, all with multiple layers of filtration, as needed. Reporting of metrics is built in. There is a provision

for proxy access (in which the user may choose to grant one or more other users to view screens and perform actions as though they were the user, during any part of a tracking life cycle — especially useful during tight build schedules and vacations to keep things moving). The system also provides users the ability to have an anomaly link to or notify other systems, including QA Inspection Reports, Safety, GIDEP (Government-Industry Data Exchange Program) Alert, Corrective Actions, and Lessons Learned.

The PRS tracking engine was designed as a very extensible and scalable system, able to support additional applications, with future development possibilities already discussed, including Incident Surprise Anomalies (for anomalies occurring during Operations phases of NASA Flight projects), GIDEP and NASA Alerts, and others.

This work was done by Don Potter, Charles Serian, Robert Sweet, Babak Sapir, Enrique Gamez, and David Mays of Caltech 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-40202.

➤ G-Guidance Interface Design for Small Body Mission Simulation

The G-Guidance software implements a guidance and control (G&C) algorithm for small-body, autonomous proximity operations, developed under the Small Body GN&C task at JPL. The software is written in Matlab and interfaces with G-OPT, a JPL-developed optimization package written in C that provides G-Guidance with guaranteed convergence to a solution in a finite computation time with a prescribed accuracy. The resulting program is computationally efficient and is a prototype of an onboard, real-time algorithm for autonomous guidance and control.

Two thruster firing schemes are available in G-Guidance, allowing tailoring of the software for specific mission maneuvers. For example, descent, landing, or rendezvous benefit from a thruster firing at the maneuver termination to mitigate velocity errors. Conversely, ascent or separation maneuvers benefit from an immediate firing to avoid potential drift toward a second body. The guidance portion of this software explicitly enforces user-defined control constraints and thruster silence times while minimizing total fuel usage.