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 flies 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.

This program is currently specialized to small-body proximity operations, but the underlying method can be generalized to other applications.

This program was written by Behçet Açikmeşe, John Carson, and Linh Phan 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-44291.

DSN Scheduling Engine

The DSN (Deep Space Network) Scheduling Engine targets all space missions that use DSN services. It allows clients to issue scheduling, conflict identification, conflict resolution, and status requests in XML over a Java Message Service interface. The scheduling requests may include new requirements that represent a set of tracks to be scheduled under some constraints. This program uses a heuristic local search to schedule a variety of schedule requirements, and is being infused into the Service Scheduling Assembly, a mixed-initiative scheduling application.

The engine resolves conflicting schedules of resource allocation according to a range of existing and possible requirement specifications, including optional antennas; start of track and track duration ranges; periodic tracks; locks on track start, duration, and allocated antenna; MSPA (multiple spacecraft per aperture); arraying/VLBI (very long baseline interferometry)/delta DOR (differential one-way ranging); continuous tracks; segmented tracks; gap-to-track ratio; and override or block-out of requirements. The scheduling models now include conflict identification for SOA (start of activity), BOT (beginning of track), RFI (radio frequency interference), and equipment constraints. This software will search through all possible allocations while providing a best-effort solution at any time.

The engine reschedules to accommodate individual emergency tracks in 0.2 second, and emergency antenna downtime in 0.2 second. The software handles doubling of one mission's track requests over one week (to 42 total) in 2.7 seconds. Further tests will be performed in the context of actual schedules.

This program was written by Bradley Clement, Mark Johnston, Allan Wax, and Caroline Chouinard 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-44346.

Replacement Sequence of Events Generator

The soeWINDOW program automates the generation of an ITAR (International Traffic in Arms Regulations)-compliant sub-RSOE (Replacement Sequence of Events) by extracting a specified temporal window from an RSOE while maintaining page header information. RSOEs contain a significant amount of information that is not ITAR-compliant, yet that foreign partners need to see for command details to their instrument, as well as the surrounding commands that provide context for validation. soeWINDOW can serve as an example of how command support products can be made ITAR-compliant for future missions.

This software is a Perl script intended for use in the mission operations UNIX environment. It is designed for use to support the MRO (Mars Reconnaissance Orbiter) instrument team. The tool also provides automated DOM (Distributed Object Manager) storage into the special ITAR-okay DOM collection, and can be used for creating focused RSOEs for product review by any of the MRO teams.

This program was written by Forest Fisher, Daniel Wenkert Roy Gladden, and Teerapat Khanampornpan 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-44392.

Force-Control Algorithm for Surface Sampling

A G-FCON algorithm is designed for small-body surface sampling. It has a linearization component and a feedback component to enhance performance. The algorithm regulates the contact force between the tip of a robotic arm attached to a spacecraft and a surface during sampling. The control algorithm is insensitive to the surface properties, enabling it to maintain the right contact force for a wide range of surface compliance properties.

The objective of the algorithm is to bring the sampler in contact with the small body surface, and maintain a desired contact force for a prescribed duration of time for sampling. Once the sampling period is over, the control algorithm guides the spacecraft safely away from the surface.

This work was done by Behçet Açikmeşe, Marco B. Quadrelli, and Linh Phan of Caltech for NASA's Jet Propulsion Laboratory.

The software used in this innovation is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-44377.

Tool for Merging Proposals Into DSN Schedules

A Practical Extraction and Reporting Language (Perl) script called "merge7da" has been developed to facilitate determination, by a project scheduler in NASA's Deep Space Network, of whether a proposal for use of the DSN could create a conflict with the current DSN schedule. Prior to the development of merge7da, there was no way to quickly identify potential schedule conflicts: it was necessary to submit a proposal and wait a day or two for a response from a DSN scheduling facility. By using merge7da to detect and eliminate potential schedule conflicts before submitting a proposal, a project scheduler saves time and gains assurance that the proposal will probably be accepted. merge7da accepts two input files, one of which contains the current DSN schedule and is in a DSN-standard format called "7da."

The other input file contains the proposal and is in another DSN-standard format called "C1/C2." merge7da processes the two input files to produce a merged 7da-format output file that represents the DSN schedule as it would be if the proposal were to be adopted. This 7da output file can be loaded into various DSN scheduling software tools now in use.

This program was written by Teerapat Khanampornpan, John Kwok, and Jared Call 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-44582.