Software

Public Risk Assessment Program

The Public Entry Risk Assessment (PERA) program addresses risk to the public from shuttle or other spacecraft re-entry trajectories. Managing public risk to acceptable levels is a major component of safe spacecraft operation. PERA is given scenario inputs of vehicle trajectory, probability of failure along that trajectory, the resulting debris characteristics, and field size and distribution, and returns risk metrics that quantify the individual and collective risk posed by that scenario. Due to the large volume of data required to perform such a risk analysis, PERA was designed to streamline the analysis process by using innovative mathematical analysis of the risk assessment equations. Real-time analysis in the event of a shuttle contingency operation, such as damage to the Orbiter, is possible because PERA allows for a change to the probability of failure models, therefore providing a much quicker estimation of public risk.

PERA also provides the ability to generate movie files showing how the entry risk changes as the entry develops. PERA was designed to streamline the computation of the enormous amounts of data needed for this type of risk assessment by using an average distribution of debris on the ground, rather than pinpointing the impact point of every piece of debris. This has reduced the amount of computational time significantly without reducing the accuracy of the results. PERA was written in MATLAB; a compiled version can run from a DOS or UNIX prompt.

This program was written by Gavin Mendeck of Johnson Space Center. Further information is contained in a TSP (see page 1). MSC-24166-1

Particle Swarm Optimization Toolbox

The Particle Swarm Optimization Toolbox is a library of evolutionary optimization tools developed in the MATLAB environment. The algorithms contained in the library include a genetic algorithm (GA), a single-objective particle swarm optimizer (SOPSO), and a multi-objective particle swarm optimizer (MOPSO). Development focused on both the SOPSO and MOPSO. A GA was included mainly for comparison purposes, and the particle swarm optimizers appeared to perform better for a wide variety of optimization problems. All algorithms are capable of performing unconstrained and constrained optimization. The particle swarm optimizers are capable of performing single and multi-objective optimization. The SOPSO and MOPSO algorithms are based on swarming theory and bird-flocking patterns to search the trade space for the optimal solution or optimal trade in competing objectives. The MOPSO generates Pareto fronts for objectives that are in competition.

A GA, based on Darwin evolutionary theory, is also included in the library. The GA consists of individuals that form a population in the design space. The population mates to form offspring at new locations in the design space. These offspring contain traits from both of the parents. The algorithm is based on this combination of traits from parents to hopefully provide an improved solution than either of the original parents. As the algorithm progresses, individuals that hold these optimal traits will emerge as the optimal solutions.

Due to the generic design of all optimization algorithms, each algorithm interfaces with a user-supplied objective function. This function serves as a "black-box" to the optimizers in which the only purpose of this function is to evaluate solutions provided by the optimizers. Hence, the user-supplied function can be numerical simulations, analytical functions, etc., since the specific detail of this function is of no concern to the optimizer. These algorithms were originally developed to support entry trajectory and guidance design for the Mars Science Laboratory mission but may be applied to any optimization problem.

The MSL simulations reside on a computational network of development computers and two clusters at NASA Langley. The MSL can take advantage of the parallel nature of these population-based algorithms with the optimization algorithms running with the Mars entry simulations on the Langley clusters via the user-supplied interface. Other problems for which this software might be used do not necessarily require use of the Langley clusters. The group in which this innovation was developed uses the algorithms for MSL, but due to its generic nature, other uses can include Crew Exploration Vehicle ascent, entry, mission design, or any other project that can use this type of toolset.

This program was written by Michael J. Grant for Johnson Space Center. Further information is contained in a TSP (see page 1). MSC-24261-1.

Telescience Support Center Data System Software

The Telescience Support Center (TSC) team has developed a database-driven, increment-specific Data Requirement Document (DRD) generation tool that automates much of the work required for generating and formatting the DRD. It creates a database to load the required changes to configure the TSC data system, thus eliminating a substantial amount of labor in database entry and formatting.

The TSC database contains the TSC systems configuration, along with the experimental data, in which human physiological data must be de-commutated in real time. The data for each experiment also must be cataloged and archived for future retrieval. TSC software provides tools and resources for ground operation and data distribution to remote users consisting of PIs (principal investigators), bio-medical engineers, scientists, engineers, payload specialists, and computer scientists. Operations support is provided for computer systems access, detailed networking, and mathematical and computational problems of the International Space Station telemetry data.

User training is provided for on-site staff and biomedical researchers and other remote personnel in the usage of the space-bound services via the Internet, which enables significant resource savings for the physical facility along with the time savings versus traveling to NASA sites. The software used in support of the TSC could easily be adapted to other Control Center applications. This would include not only other NASA payload monitoring facilities, but also other types

of control activities, such as monitoring and control of the electric grid, chemical, or nuclear plant processes, air traffic control, and the like.

This program was written by Hasan Rahman of Lockheed Martin for Johnson Space Center. Further information is contained in a TSP (see page 1). MSC-24337-1

Update on PISCES

An updated version of the Platform Independent Software Components for the Exploration of Space (PISCES) software library is available. A previous version was reported in "Library for Developing Spacecraft-Mission-Planning Software" (MSC-22983), *NASA Tech Briefs*, Vol. 25, No. 7 (July 2001), page 52. To recapitulate: This software provides for Web-based, collaborative development of computer programs for planning trajectories and trajectory-related aspects of spacecraft-mission design. The library was built using state-of-the-art object-oriented concepts and software-development methodologies. The components of PISCES include Java-language application programs arranged in a hierarchy of classes that facilitates the reuse of the components.

As its full name suggests, the PISCES library affords platform-independence: The Java language makes it possible to use the classes and application programs with a Java virtual machine, which is available in most Web-browser programs. Another advantage is expandability: Object orientation facilitates expansion of the library through creation of a new class. Improvements in the library since the previous version include development of orbital-maneuver-planning and rendezvous-launch-window application programs, enhancement of capabilities for propagation of orbits, and development of a "desktop" user interface.

This program was written by Don Pearson, Dustin Hamm, Brian Kubena, and Jonathan K. Weaver of Johnson Space Center. For further information, contact the Johnson Commercial Technology Office at (281) 483-3809. MSC-23633-1

Ground and Space Radar Volume Matching and Comparison Software

This software enables easy comparison of ground- and space-based radar observations. The software was initially designed to compare ground radar reflec-

tivity from operational, ground based S- and C-band meteorological radars with comparable measurements from the Tropical Rainfall Measuring Mission (TRMM) satellite's Precipitation Radar (PR) instrument. The software is also applicable to other ground-based and space-based radars. The ground and space radar volume matching and comparison software was developed in response to requirements defined by the Ground Validation System (GVS) of Goddard's Global Precipitation Mission (GPM) project.

This software innovation is specifically concerned with simplifying the comparison of ground- and space-based radar measurements for the purpose of GPM algorithm and data product validation. This software is unique in that it provides an operational environment to routinely create comparison products, and uses a direct geometric approach to derive common volumes of space- and ground-based radar data. In this approach, spatially coincident volumes are defined by the intersection of individual space-based Precipitation Radar rays with the each of the conical elevation sweeps of the ground radar. Thus, the resampled volume elements of the space and ground radar reflectivity can be directly compared to one another.

This work was done by Kenneth Morris and Mathew Schwaller of Goddard Space Flight Center. Further information is contained in a TSP (see page 1). Additional information can also be found on the GPM GVS web site: http://gpm.gsfc.nasa.gov/groundvalidation.html. GSC-15738-1

Web-Based Interface for Command and Control of Network Sensors

This software allows for the visualization and control of a network of sensors through a Web browser interface. It is currently being deployed for a network of sensors monitoring Mt. Saint Helen's volcano; however, this innovation is generic enough that it can be deployed for any type of sensor Web. From this interface, the user is able to fully control and monitor the sensor Web. This includes, but is not limited to, sending "test" commands to individual sensors in the network, monitoring for real-world events, and reacting to those events.

This work was done by Michael N. Wallick, Joshua R. Doubleday, and Khawaja S. Shams of Caltech for NASA's Jet Propulsion

Laboratory. For more information, contact iaoffice@jpl.nasa.gov.

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47110.

Orbit Determination Toolbox

The Orbit Determination Toolbox is an orbit determination (OD) analysis tool based on MATLAB and Java that provides a flexible way to do early mission analysis. The toolbox is primarily intended for advanced mission analysis such as might be performed in concept exploration, proposal, early design phase, or rapid design center environments. The emphasis is on flexibility, but it has enough fidelity to produce credible results. Insight into all flight dynamics source code is provided.

MATLAB is the primary user interface and is used for piecing together measurement and dynamic models. The Java Astrodynamics Toolbox is used as an engine for things that might be slow or inefficient in MATLAB, such as high-fidelity trajectory propagation, lunar and planetary ephemeris look-ups, precession, nutation, polar motion calculations, ephemeris file parsing, and the like. The primary analysis functions are sequential filter/smoother and batch least-squares commands that incorporate Monte-Carlo data simulation, linear covariance analysis, measurement processing, and plotting capabilities at the generic level.

These functions have a user interface that is based on that of the MATLAB ODE suite. To perform a specific analysis, users write MATLAB functions that implement truth and design system models. The user provides his or her models as inputs to the filter commands. The software provides a capability to publish and subscribe to a software bus that is compliant with the NASA Goddard Mission Services Evolution Center (GMSEC) standards, to exchange data with other flight dynamics tools to simplify the flight dynamics design cycle. Using the publish and subscribe approach allows for analysts in a rapid design center environment to seamlessly incorporate changes in spacecraft and mission design into navigation analysis and vice versa.

This work was done by James R. Carpenter and Kevin Berry of Goddard Space Flight Center and Kate Gregory, Keith Speckman, Sun Hur-Diaz, Derek Surka, and Dave Gaylor of Emergent Space Technologies, Inc. For