Inquiries concerning rights for the commercial use of this invention should be addressed to NASA Glenn Research Center, Commercial Technology Office, Attn: Steve Fedor, Mail Stop 4–8, 21000 Brookpark Road, Cleveland, Ohio 44135. Refer to LEW-17465-1.

Self-Organizing-Map Program for Analyzing Multivariate Data

SOM_VIS is a computer program for analysis and display of multidimensional sets of Earth-image data typified by the data acquired by the Multi-angle Imaging Spectro-Radiometer [MISR (a spaceborne instrument)]. In SOM_VIS, an enhanced self-organizing-map (SOM) algorithm is first used to project a multidimensional set of data into a nonuniform three-dimensional lattice structure. The lattice structure is mapped to a color space to obtain a color map for an image. The Voronoi cell-refinement algorithm is used to map the SOM lattice structure to various levels of color resolution. The final result is a false-color image in which similar colors represent similar characteristics across all its data dimensions. SOM_VIS provides a control panel for selection of a subset of suitably preprocessed MISR radiance data, and a control panel for choosing parameters to run SOM training. SOM_VIS also includes a component for displaying the false-color SOM image, a color map for the trained SOM lattice, a plot showing an original input vector in 36 dimensions of a selected pixel from the SOM image, the SOM vector that represents the input vector, and the Euclidean distance between the two vectors.

This program was written by P. Peggy Li, Joseph C. Jacob, Gary L. Block, and Amy J. Braverman 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 Karina Edmonds of the California Institute of Technology at (818) 393-2827. Refer to NPO-40666.

Tool for Sizing Analysis of the Advanced Life Support System

Advanced Life Support Sizing Analysis Tool (ALSSAT) is a computer model for sizing and analyzing designs of environmental-control and life support systems (ECLSS) for spacecraft and surface habitats involved in the exploration of Mars and Moon. It performs conceptual designs of advanced life support (ALS) subsystems that utilize physicochemical and biological processes to recycle air and water, and process wastes in order to reduce the need of resource resupply. By assuming steady-state operations, ALSSAT is a means of investigating combinations of such subsystems' technologies and thereby assisting in determining the most cost-effective technology combination available. In fact, ALSSAT can perform sizing analysis of the ALS subsystems that are operated dynamically or steady in nature. Using the Microsoft Excel® spreadsheet software with Visual Basic programming language, ALSSAT has been developed to perform multiple-case trade studies based on the calculated ECLSS mass, volume, power, and Equivalent System Mass, as well as parametric studies by varying the input parameters. ALSSAT's modular format is specifically designed for the ease of future maintenance and upgrades.

This program was developed by Hue-Hsia (Jannivine) Yeh, Cheryl B. Brown, and Frank J. Jeng of Lockheed Martin Corp. for **Johnson Space Center**. *For further information, contact the Johnson Technology Transfer Office at (281) 483-3809.*
MSC-23506

Control Software for a High-Performance Telerobot

A computer program for controlling a high-performance, force-reflecting telerobot has been developed. The goal in designing a telerobot-control system is to make the velocity of the slave match the master velocity, and the environmental force on the master match the force on the slave. Instability can arise from even small delays in propagation of signals between master and slave units. The present software, based on an impedance-shaping algorithm, ensures stability even in the presence of long delays. It implements a real-time algorithm that processes position and force measurements from the master and slave and represents the master/slave communication link as a transmission line. The algorithm also uses the history of the control force and the slave motion to estimate the impedance of the environment. The estimate of the impedance of the environment is used to shape the controlled slave impedance to match the transmission-line impedance. The estimate of the environmental impedance is used to match the master and transmission-line impedances and to estimate the slave/environment force in order to present that force immediately to the operator via the master unit.

This Robert J. Kline-Schoder and William Finger of Creare, Inc., for **Johnson Space Center**. *For further information, contact the Johnson Technology Transfer Office at (281) 483-3809.*
MSC-23412

Java Radar Analysis Tool

Java Radar Analysis Tool (JRAT) is a computer program for analyzing two-dimensional (2D) scatter plots derived from radar returns showing pieces of the disintegrating Space Shuttle *Columbia*. JRAT can also be applied to similar plots representing radar returns showing aviation accidents, and to scatter plots in general. The 2D scatter plots include overhead map views and side altitude views. The superposition of points in these views makes searching difficult. JRAT enables three-dimensional (3D) viewing: by use of a mouse and keyboard, the user can rotate to any desired viewing angle. The 3D view can include overlaid trajectories and search footprints to enhance situational awareness in searching for pieces. JRAT also enables playback: time-tagged radar-return data can be displayed in time order and an animated 3D model can be moved through the scene to show the locations of the *Columbia* (or other vehicle) at the times of the corresponding radar events. The combination of overlays and playback enables the user to correlate a radar return with a position of the vehicle to determine whether the return is valid. JRAT can optionally filter single radar returns, enabling the user to selectively hide or highlight a desired radar return.

This program was written by Mariusz P. Zaczek of **Johnson Space Center**. *For further information, contact the Johnson Technology Transfer Office at (281) 483-3809.*
MSC-23742

Architecture for Verifiable Software

Verifiable MDS Architecture (VMA) is a software architecture that facilitates the construction of highly verifiable flight software for NASA's Mission Data System (MDS), especially for smaller missions subject to cost constraints. More specifically, the purpose served by VMA is to facilitate aggressive verification and validation of flight software while imposing a minimum of constraints on overall functionality. VMA exploits the state-based architecture of the MDS and partitions verification issues

into elements susceptible to independent verification and validation, in such a manner that scaling issues are minimized, so that relatively large software systems can be aggressively verified in a cost-effective manner.

This work was done by William Reinholtz and Daniel Dvorak 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 Karina Edmonds of the California Institute of Technology at (818) 393-2827. Refer to NPO-40842.

Tool for Ranking Research Options

Tool for Research Enhancement Decision Support (TREDS) is a computer program developed to assist managers in ranking options for research aboard the International Space Station (ISS). It could likely also be adapted to perform similar decision-support functions in industrial and academic settings. TREDS provides a ranking of the options, based on a quantifiable assessment of all the relevant programmatic decision factors of benefit, cost, and risk. The computation of the benefit for each option is based on a figure of merit (FOM) for ISS research capacity that incorporates both quantitative and qualitative inputs. Qualitative inputs are gathered and partly quantified by use of the time-tested analytical hierarchical process and used to set weighting factors in the FOM corresponding to priorities determined by the cognizant decision maker(s). Then by use of algorithms developed specifically for this application, TREDS adjusts the projected benefit for each option on the basis of levels of technical implementation, cost, and schedule risk. Based partly on Excel spreadsheets, TREDS provides screens for entering cost, benefit, and risk information. Drop-down boxes are provided for entry of qualitative information. TREDS produces graphical output in multiple formats that can be tailored by users.

This program was written by James N. Ortiz of **Johnson Space Center**, *Kelly Scott of Booz Allen Hamilton Inc., and Harold Smith of Raytheon Co. For further information, contact the Johnson Technology Transfer Office at (281) 483-3809.*
MSC-23744

Enhanced, Partially Redundant Emergency Notification System

The Johnson Space Center Emergency Notification System (JENS) software utilizes pre-existing computation and communication infrastructure to augment a prior variable-tone, siren-based, outdoor alarm system, in order to enhance the ability to give notice of emergencies to employees working in multiple buildings. The JENS software includes a component that implements an administrative Web site. Administrators can grant and deny access to the administrative site and to an originator Web site that enables authorized individuals to quickly compose and issue alarms. The originator site also facilitates maintenance and review of alarms already issued. A custom client/server application program enables an originator to notify every user who is logged in on a Microsoft Windows-based desktop computer by means of a pop-up message that interrupts, but does not disrupt, the user's work. Alternatively or in addition, the originator can send an alarm message to recipients on an e-mail distribution list and/or can post the notice on an internal Web site. An alarm message can consist of (1) text describing the emergency and suggesting a course of action and (2) a replica of the corresponding audible outdoor alarm.

This program was written by Clark D. Pounds of Science Applications International Corp. for **Johnson Space Center**. *For further information, contact the Johnson Technology Transfer Office at (281) 483-3809.*
MSC-23773

Close-Call Action Log Form

"Close Call Action Log Form" ("CCALF") is the name of both a computer program and a Web-based service provided by the program for creating an enhanced database of close calls (in the colloquial sense of mishaps that were avoided by small margins) assigned to the Center Operations Directorate (COD) at Johnson Space Center. CCALF provides a single facility for on-line collaborative review of close calls. Through CCALF, managers can delegate responses to employees. CCALF utilizes a pre-existing e-mail system to notify managers that there are close calls to review, but eliminates the need for the prior practices of passing multiple e-mail messages around the COD, then collecting and consolidating them into final responses: CCALF now collects comments from all responders for incorporation into reports that it generates. Also, whereas it was previously necessary to manually calculate metrics (e.g., numbers of maintenance-work orders necessitated by close calls) for inclusion in the reports, CCALF now computes the metrics, summarizes them, and displays them in graphical form. The reports and all pertinent information used to generate the reports are logged, tracked, and retained by CCALF for historical purposes.

This work was done by Linda M. Spuler and Patricia K. Ford of **Johnson Space Center** *and Darren C. Skeete, Scot Hershman, Pushpa Raviprakash, John W. Arnold, Victor Tran, and Mary Alice Haenze of Science Applications International Corp. For further information, contact the Johnson Technology Transfer Office at (281) 483-3809.*
MSC-23808

Task Description Language

Task Description Language (TDL) is an extension of the C++ programming language that enables programmers to quickly and easily write complex, concurrent computer programs for controlling real-time autonomous systems, including robots and spacecraft. TDL is based on earlier work (circa 1984 through 1989) on the Task Control Architecture (TCA). TDL provides syntactic support for hierarchical task-level control functions, including task decomposition, synchronization, execution monitoring, and exception handling. A Java-language-based compiler transforms TDL programs into pure C++ code that includes calls to a platform-independent task-control-management (TCM) library. TDL has been used to control and coordinate multiple heterogeneous robots in projects sponsored by NASA and the Defense Advanced Research Projects Agency (DARPA). It has also been used in Brazil to control an autonomous airship and in Canada to control a robotic manipulator.

This program was written by Reid Simmons and David Apfelbaum of Carnegie Mellon University for **Johnson Space Center**. *For further information, contact the Johnson Technology Transfer Office at (281) 483-3809.*
MSC-23460