Software

Supplier Management System

Supplier Management System (SMS) allows for a consistent, agency-wide performance rating system for suppliers used by NASA. This version (2.0) combines separate databases into one central database that allows for the sharing of supplier data. Information extracted from the NBS/Oracle database can be used to generate ratings. Also, supplier ratings can now be generated in the areas of cost, product quality, delivery, and audit data. Supplier data can be charted based on real-time user input. Based on these individual ratings, an overall rating can be generated.

Data that normally would be stored in multiple databases, each requiring its own log-in, is now readily available and easily accessible with only one log-in required. Additionally, the database can accommodate the storage and display of quality-related data that can be analyzed and used in the supplier procurement decision-making process. Moreover, the software allows for a Closed-Loop System (supplier feedback), as well as the capability to communicate with other federal agencies.

While Version 1.0 only had an Approved Supplier list, Version 2.0 includes the Approved Supplier List, Supplier Rating System, and the Supplier Outreach and Process Control Assurance System. The Supplier Rating System is a new database that was developed specifically for this software. The system allows for segregation of data to ensure sensitive data is protected.

This program was written by Eric Ramirez, Sandy Gutheinz, James Brison, Anita Ho, James Allen, Olga Ceritelli, Claudia Tobar, Thuykien Nguyen, and Harrel Crenshaw of Caltech and Roxann Santos of SkillStorm Inc. 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-44084.

Improved CLARAty Functional-Layer/Decision-Layer Interface

Improved interface software for communication between the CLARAty Decision and Functional layers has been developed. [The Coupled Layer Architecture for Robotics Autonomy (CLARAty) was described in "Coupled-Layer Robotics Architecture for Autonomy" (NPO-21218), *NASA Tech Briefs*, Vol. 26, No. 12 (December 2002), page 48. To recapitulate: the CLARAty architecture was developed to improve the modularity of robotic software while tightening coupling between planning/execution and basic control subsystems. Whereas prior robotic software architectures typically contained three layers, the CLARAty contains two layers: a decision layer (DL) and a functional layer (FL).] Types of communication supported by the present software include sending commands from DL modules to FL modules and sending data updates from FL modules to DL modules.

The present software supplants prior interface software that had little error-checking capability, supported data parameters in string form only, supported commanding at only one level of the FL, and supported only limited updates of the state of the robot. The present software offers strong error checking, and supports complex data structures and commanding at multiple levels of the FL, and relative to the prior software, offers a much wider spectrum of state-update capabilities.

This program was written by Tara Estlin, Gregg Rabideau, Daniel Gaines, Mark Johnston, Caroline Chouinard, Issa Nessnas, and I-Hsiang Shu 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 (626) 395-2322. Refer to NPO-44571.

JAVA Stereo Display Toolkit

This toolkit provides a common interface for displaying graphical user interface (GUI) components in stereo using either specialized stereo display hardware (e.g., liquid crystal shutter or polarized glasses) or anaglyph display (red/blue glasses) on standard workstation displays. An application using this toolkit will work without modification in either environment, allowing stereo software to reach a wider audience without sacrificing high-quality display on dedicated hardware.

The toolkit is written in Java for use with the Swing GUI Toolkit and has cross-platform compatibility. It hooks into the graphics system, allowing any standard Swing component to be displayed in stereo. It uses the OpenGL graphics library to control the stereo hardware and to perform the rendering. It also supports anaglyph and special stereo hardware using the same API (application-program interface), and has the ability to "simulate" color stereo in anaglyph mode by combining the red band of the left image with the green/blue bands of the right image.

This is a low-level toolkit that accomplishes simply the display of components (including the JadeDisplay image display component). It does not include higher-level functions such as disparity adjustment, 3D cursor, or overlays — all of which can be built using this toolkit.

This program was written by Robert Deen and Oleg Pariser 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-43786.

Remote-Sensing Time Series Analysis, a Vegetation Monitoring Tool

The Time Series Product Tool (TSPT) is software, developed in MATLAB®, which creates and displays high signal-to-noise Vegetation Indices imagery and other higher-level products derived from remotely sensed data. This tool enables automated, rapid, large-scale regional surveillance of crops, forests, and other vegetation. TSPT temporally processes high-revisit-rate satellite imagery produced by the Moderate Resolution Imaging Spectroradiometer (MODIS) and by other remote-sensing systems. Although MODIS imagery is acquired daily, cloudiness and other sources of noise can greatly reduce the effective temporal resolution. To improve cloud statistics, the TSPT combines MODIS data from multiple satellites (Aqua and Terra). The TSPT produces MODIS products as single time-frame and multitemporal change images, as time-series plots at a selected location, or as temporally processed image videos. Using the TSPT program, MODIS metadata is used to remove and/or correct bad and suspect data. Bad pixel removal, multiple satellite data fusion, and temporal processing techniques create high-quality plots and animated image video sequences that depict changes in vegetation greenness. This tool provides several temporal processing options not found in other comparable imaging software tools. Because the framework to generate and use other algorithms is established, small modifications to this tool will enable the use of a large range of remotely sensed data types.

An effective remote-sensing crop-monitoring system must be able to detect subtle changes in plant health in the earliest stages, before the effects of a disease outbreak or other adverse environmental conditions can become widespread and devastating. The integration of the time series analysis tool with ground-based information, soil types, crop types, meteorological data, and crop growth models in a Geographic Information System, could provide the foundation for a large-area crop-surveillance system that could identify a variety of plant phenomena and improve monitoring capabilities.

This program was written by Rodney McKellip of Stennis Space Center; Donald Prados of Computer Sciences Corporation; Robert Ryan, Kenton Ross, and Joseph Spruce of Science Systems and Applications; and Gerald Gasser and Randall Greer of Lockheed Martin Space Operations, Information Systems Directorate.

Inquiries concerning rights for the commercial use of this invention should be addressed to the Intellectual Property Manager at Stennis Space Center (228) 688-1929. Refer to SSC-00261, volume, and number of this NASA Tech Briefs *issue and the page number.*

PyPele Rewritten To Use MPI

A computer program known as “PyPele,” originally written as a Python-language extension module of a C++ language program, has been rewritten in pure Python language. The original version of PyPele dispatches and coordinates parallel-processing tasks on cluster computers and provides a conceptual framework for spacecraft-mission-design and -analysis software tools to run in an embarrassingly parallel mode. The original version of PyPele uses SSH (Secure Shell — a set of standards and an associated network protocol for establishing a secure channel between a local and a remote computer) to coordinate parallel processing. Instead of SSH, the present Python version of PyPele uses Message Passing Interface (MPI) [an unofficial *de-facto* standard language-independent application programming interface for message-passing on a parallel computer] while keeping the same user interface.

The use of MPI instead of SSH and the preservation of the original PyPele user interface make it possible for parallel application programs written previously for the original version of PyPele to run on MPI-based cluster computers. As a result, engineers using the previously written application programs can take advantage of embarrassing parallelism without need to rewrite those programs.

This program was written by George Hockney and Seungwon Lee 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-44729.

Data Assimilation Cycling for Weather Analysis

This software package runs the atmospheric model MM5 in data assimilation cycling mode to produce an optimized weather analysis, including the ability to insert or adjust a hurricane vortex. The program runs MM5 through a cycle of short forecasts every three hours where the vortex is adjusted to match the observed hurricane location and storm intensity. This technique adjusts the surrounding environment so that the proper steering current and environmental shear are achieved. MM5cycle uses a Cressman analysis to blend observation into model fields to get a more accurate weather analysis. Quality control of observations is also done in every cycle to remove bad data that may contaminate the analysis. This technique can assimilate and propagate data in time from intermittent and infrequent observations while maintaining the atmospheric field in a dynamically balanced state.

The software consists of a C-shell script (MM5cycle.driver) and three FORTRAN programs (splitMM5files.F, comRegrid.F, and insert_vortex.F), and are contained in the pre-processor component of MM5 called "Regridder." The model is first initialized with data from a global model such as the Global Forecast System (GFS), which also provides lateral boundary conditions. These data are separated into single-time files using splitMM5.F. The hurricane vortex is then bogussed in the correct location and with the correct wind field using insert_vortex.F. The modified initial and boundary conditions are then recombined into the model fields using comRegrid.F. The model then makes a three-hour forecast. The three-hour forecast data from MM5 now become the analysis for the next short forecast run, where the vortex will again be adjusted. The process repeats itself until the desired time of analysis is achieved. This code can also assimilate observations if desired.

This program was written by Nam Tran, Yongzuo Li, and Patrick Fitzpatrick of the GeoResources Institute at Mississippi State University for Stennis Space Center.

Inquiries concerning rights for its commercial use should be addressed to:

GeoResources Institute
Mississippi State University
Building 1103, Room 233
Stennis Space Center, MS 39529
Phone No.: (228) 688-4218
Web site: www.gri.msstate.edu

Refer to SSC-00276/7, the volume and number of this NASA Tech Briefs *issue, and the page number.*