This software is available for commercial licensing. Please contact Dan Broderick at Daniel.F.Broderick@jpl.nasa.gov. Refer to NPO-47781.

ISTP CDF Skeleton Editor

Basic Common Data Format (CDF) tools (e.g., cdfedit) provide no specific support for creating International Solar-Terrestrial Physics/Space Physics Data Facility (ISTP/SPDF) standard files. While it is possible for someone who is familiar with the ISTP/SPDF metadata guidelines to create compliant files using just the basic tools, the process is error-prone and unreasonable for someone without ISTP/SPDF expertise. The key problem is the lack of a tool with specific support for creating files that comply with the ISTP/SPDF guidelines. There are basic CDF tools such as cdfedit and skeletoncdf for creating CDF files, but these have no specific support for creating ISTP/ SPDF compliant files.

The SPDF ISTP CDF skeleton editor is a cross-platform, Java-based GUI editor program that allows someone with only a basic understanding of the ISTP/SPDF guidelines to easily create compliant files. The editor is a simple graphical user interface (GUI) application for creating and editing ISTP/SPDF guideline-compliant skeleton CDF files. The SPDF ISTP CDF skeleton editor consists of the following components: A swing-based Java GUI program, JavaHelp-based manual/tutorial, Image/Icon files, and HTML Web page for distribution. The editor is available as a traditional Java desktop application as well as a Java Network Launching Protocol (JNLP) application. Once started, it functions like a typical Java GUI file editor application for creating/editing application-unique files.

The editor provides ease of use and support for ISTP/SPDF and project-specific standards. The editor provides support for creating/editing CDF files that comply with the ISTP/SPDF guidelines.

This work was done by Reine Chimiak and Bernard Harris of Goddard Space Flight Center, and Phillip Williams of QSS Group. Further information is contained in a TSP (see page 1). GSC-16256-1

Uplink Summary Generator (ULSGEN) Version 1.0

The Uplink Summary Generator (ULSGEN) provides a convenient means of gathering together a set of uplink related files, parsing and analyzing these files, and producing a summary of their contents, which may then be electronically signed by one or more reviewers to verify the commands. Spacecraft operations personnel view this summary as a final sanity check before actual radiation of the uplink data.

Unique features of the software are a browser-based application that can be used both inside and outside the flight operations firewall, file retrieval from the project file server or from the project DOM (Distributed Object Manager), and the ability to parse and analyze spacecraft command files (SCMF). The software also features DSN keyword file (DKF) parsing for uplink windows, and enables hosting of one or more projects in a single server. Each project can define its own uplink summary template.

Each uplink summary is generated based on the analysis results from the parsers and the selected project template. The uplink summary review and signature collection cycle supports both parallel and sequential workflows. RadList file generation enables linkage to the command system.

This work was done by Yeou-Fang Wang, Mitchell Schrock, Timothy J. Reeve, Kristine T. Fong, and Benjamin D. Smith 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 Dan Broderick at Daniel.F.Broderick@jpl.nasa.gov. Refer to NPO-48423.

Robotics On-Board Trainer (ROBoT)

ROBoT is an on-orbit version of the ground-based Dynamics Skills Trainer (DST) that astronauts use for training on a frequent basis. This software consists of two primary software groups. The first series of components is responsible for displaying the graphical scenes. The remaining components are responsible for simulating the Mobile Servicing System (MSS), the Japanese Experiment Module Remote Manipulator System (JEMRMS), and the H-II Transfer Vehicle (HTV) Free Flyer Robotics Operations. The MSS simulation software includes: Robotic Workstation (RWS) simulation, a simulation of the Space Station Remote Manipulator System (SSRMS), a simulation of the ISS Command and Control System (CCS), and a portion of the Portable Computer System (PCS) software necessary for MSS operations.

These components all run under the CentOS4.5 Linux operating system. The JEMRMS simulation software includes real-time, HIL, dynamics, manipulator multi-body dynamics, and a moving object contact model with Tricks discrete time scheduling. The JEMRMS DST will be used as a functional proficiency and skills trainer for flight crews. The HTV Free Flyer Robotics Operations simulation software adds a functional simulation of HTV vehicle controllers, sensors, and data to the MSS simulation software. These components are intended to support HTV ISS visiting vehicle analysis and training. The scene generation software will use DOUG (Dynamic On-orbit Ubiquitous Graphics) to render the graphical scenes. DOUG runs on a laptop running the CentOS4.5 Linux operating system. DOUG is an Open GL-based 3D computer graphics rendering package. It uses pre-built three-dimensional models of on-orbit ISS and space shuttle systems elements, and provides realtime views of various station and shuttle configurations.

This work was done by Genevieve Johnson of Johnson Space Center and Greg Alexander of Harmony Lane Studios, Inc. Further information is contained in a TSP (see page 1). MSC-25005-1

Software Engineering Tools for Scientific Models

Software tools were constructed to address issues the NASA Fortran development community faces, and they were tested on real models currently in use at NASA. These proof-of-concept tools address the High-End Computing Program and the Modeling, Analysis, and Prediction Program. Two examples are the NASA Goddard Earth Observing System Model, Version 5 (GEOS-5) atmospheric model in Cell Fortran on the Cell Broadband Engine, and the Goddard Institute for Space Studies (GISS) coupled atmosphere-ocean model called ModelE, written in fixed format Fortran.

To test the tool set, the innovators first extended an annotation and conversation mechanism, known as Activities, allowing developers to provide insights into code without modifying it to include the qualification of Activities with metadata for filtering. Next, the designers created a visualization to present the relationships, or connectivity, between model variables by tracing various constructs through different components and levels of a model.

Also, a type replacement facility was tested for updating primitive types such as such as integers that were not qualified with a size, and thus may change as the underlying architecture is upgraded. Finally, the designers leveraged a tool, called eclim, to bridge the gap between text editors and Integrated Development Environments (IDE), by running both of them simultaneously, and set up to communicate with each other. Through this mechanism, modern IDE features were made available through text editors, minimizing the learning curve for scientists already experienced with their conventions.

This work was done by Marc Abrams, Palabi Saboo, and Mike Sonsini of Harmonia Holdings Group, LLC for Goddard Space Flight Center. Further information is contained in a TSP (see page 1). GSC-16475-1

Automatic Data Filter Customization Using a Genetic Algorithm

This work predicts whether a retrieval algorithm will usefully determine CO_2 concentration from an input spectrum of GOSAT (Greenhouse Gases Observing Satellite). This was done to eliminate needless runtime on atmospheric soundings that would never yield useful results. A space of 50 dimensions was examined for predictive power on the final CO_2 results.

Retrieval algorithms are frequently expensive to run, and wasted effort defeats requirements and expends needless resources. This algorithm could be used to help predict and filter unneeded runs in any computationally expensive regime.

Traditional methods such as the Fischer discriminant analysis and decision trees can attempt to predict whether a sounding will be properly processed. However, this work sought to detect a subsection of the dimensional space that can be simply filtered out to eliminate unwanted runs. LDAs (linear discriminant analyses) and other systems examine the entire data and judge a "best fit," giving equal weight to complex and problematic regions as well as simple, clear-cut regions. In this implementation, a genetic space of "left" and "right" thresholds outside of which all data are rejected was defined. These left/right pairs are created for each of the 50 input dimensions. A genetic algorithm then runs through countless potential filter settings using a JPL computer cluster, optimizing the tossed-out data's yield (proper vs. improper run removal) and number of points tossed.

This solution is robust to an arbitrary decision boundary within the data and avoids the global optimization problem of whole-dataset fitting using LDA or decision trees. It filters out runs that would not have produced useful CO_2 values to save needless computation. This would be an algorithmic preprocessing improvement to any computationally expensive system.

This work was done by Lukas Mandrake 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 Dan Broderick at Daniel.F.Broderick@jpl.nasa.gov. Refer to NPO-47788.

Tracker Toolkit

This software can track multiple moving objects within a video stream simultaneously, use visual features to aid in the tracking, and initiate tracks based on object detection in a subregion. A simple programmatic interface allows plugging into larger image chain modeling suites. It extracts unique visual features for aid in tracking and later analysis, and includes sub-functionality for extracting visual features about an object identified within an image frame.

Tracker Toolkit utilizes a feature extraction algorithm to tag each object with metadata features about its size, shape, color, and movement. Its functionality is independent of the scale of objects within a scene. The only assumption made on the tracked objects is that they move. There are no constraints on size within the scene, shape, or type of movement. The Tracker Toolkit is also capable of following an arbitrary number of objects in the same scene, identifying and propagating the track of each object from frame to frame. Target objects may be specified for tracking beforehand, or may be dynamically discovered within a tripwire region. Initialization of the Tracker Toolkit algorithm includes two steps: Initializing the data structures for tracked target objects, including targets preselected for tracking; and initializing the tripwire region. If no tripwire region is desired, this step is skipped. The tripwire region is an area within the frames that is always checked for new objects, and all new objects discovered within the region will be tracked until lost (by leaving the frame, stopping, or blending in to the background).

This work was done by Steven J. Lewis and David M. Palacios 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 Dan Broderick at Daniel.F.Broderick@jpl.nasa.gov. Refer to NPO-48253.

Towards Efficient Scientific Data Management Using Cloud Storage

A software prototype allows users to backup and restore data to/from both public and private cloud storage such as Amazon's S3 and NASA's Nebula. Unlike other off-the-shelf tools, this software ensures user data security in the cloud (through encryption), and minimizes users' operating costs by using space- and bandwidth-efficient compression and incremental backup. Parallel data processing utilities have also been developed by using massively scalable cloud computing in conjunction with cloud storage.

One of the innovations in this software is using modified open source components to work with a private cloud like NASA Nebula. Another innovation is porting the complex backup-to-cloud software to embedded Linux, running on the home networking devices, in order to benefit more users.

This work was done by Qiming He of Open Research for Goddard Space Flight Center. For further information, contact the Goddard Innovative Partnerships Office at (301) 286-5810. GSC-16415-1