Software

Automation Framework for Flight Dynamics Products Generation

XFDS provides an easily adaptable automation platform. To date it has been used to support flight dynamics operations. It coordinates the execution of other applications such as Satellite TookKit, FreeFlyer, MATLAB, and Perl code. It provides a mechanism for passing messages among a collection of XFDS processes, and allows sending and receiving of GMSEC messages. A unified and consistent graphical user interface (GUI) is used for the various tools. Its automation configuration is stored in text files, and can be edited either directly or using the GUI.

XFDS is implemented as a group of cooperating processes. One process coordinates communications, another drives an optional GUI (not needed if running in batch mode), and the rest carry out automation tasks. The software is designed around three concepts: (1) an "action" controls an automation step; (2) a "variable" allows information to be shared among actions; and (3) a "form" corresponds to a GUI widget, which can be reused between action editors.

A significant strength of this approach is to provide a high-level abstraction to the procedures that need to be carried out. Frequently changed parameters are readily available for modification, while the rest are hidden. Additional programs that provide a batch interface can be added to this system.

This work was done by Robert E. Wiegand, Timothy C. Esposito, John S. Watson, Linda Jun, Wendy Shoan, and Carla Matusow of Goddard Space Flight Center and Wayne McCullough of Computer Sciences Corp. For further information, contact the Goddard Innovative Partnerships Office at (301) 286-5810. GSC-15618-1.

Product Operations Status Summary Metrics

The Product Operations Status Summary Metrics (POSSUM) computer program provides a readable view into the state of the Phoenix Operations Product Generation Subsystem (OPGS) data pipeline. POSSUM provides a user interface that can search the data store, collect product metadata, and display the results in an easily-readable layout. It was designed with flexibility in mind for support in future missions. Flexibility over various data store hierarchies is provided through the disk-searching facilities of Marsviewer. This is a proven program that has been in operational use since the first day of the Phoenix mission.

POSSUM presents a graphical representation for tracking and accountability of the existence and version of expected Reduced Data Records (RDRs) for a given Experiment Data Record (EDR) for the Phoenix lander mission. By using POSSUM, operations personnel can easily determine if any RDRs are missing for a given EDR during and after OPGS processing pipeline. A variety of sort options exists, including product file creation time and instrument.

This work was done by Atsuya Takagi and Nicholas Toole of Caltech for NASA's Jet Propulsion Laboratory.

The software used in this innovation is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-46664.

Mars Terrain Generation

A suite of programs for the generation of disparity maps from stereo image pairs via correlation, and conversion of those disparity maps to XYZ maps, has been updated. This suite implements an automated method of deriving terrain from stereo images for use in the ground data system for *in-situ* (lander and rover) cameras. This differs from onboard correlation by concentrating more on accuracy than speed, since near-real-time is not a requirement on the ground. The final result is an XYZ value for every point in the image that passes several quality checks. *A priori* geometric camera calibration information is required for this suite to operate.

The suite is very flexible, enabling its use in many special situations, such as non-linearized images required for applications like the Phoenix arm camera, or long-baseline stereo, where the rover moves between left and right images. The suite consists of:

- marscor3: The primary correlation program used by MER, PHX, and (soon) MSL. Requires a low-resolution disparity map as input, and refines it.
- marsjplstereo: A wrapper around a much faster correlator that assumes the images are epipolar aligned ("linearized"). Creates the input for marscor3.
- marsunlinearize: Takes a linearized correlation result and unprojects it back to raw geometry. Creates the input for marscor3 in some non-linearized situations.
- marsfakedisp: Creates the input for marscor3 in some non-linearized cases by assuming an approximate surface.
- marsdispinvert: "Inverts" a disparity map (e.g., from L→R to R→L) to create an input for marscor3.
- marsxyz: Takes the disparity map (e.g., from marscor3) and generates XYZ coordinates for each pixel.
- marsfilter: Filters the output of marsxyz to mask off undesirable areas (such as the rover itself, horizon, etc.).
- marsrange: Takes the results from marsxyz and computes range from the camera for each pixel.

While the underlying correlation coefficient computation is nearly the same as in the original, the algorithms driving how that correlation is accomplished have been completely redesigned. In addition, significant new capability exists, such as non-linearized stereo, R→L inversion, masking, and range computation. Significant additions to marsxyz's filtering capability to reject bad correlations have also been made.

This work was done by Robert G. Deen of Caltech for NASA's Jet Propulsion Laboratory.

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-46659.

Application-Controlled Parallel Asynchronous Input/Output Utility

A software utility tool has been designed to alleviate file system I/O performance bottlenecks to which many high-end computing (HEC) applications fall prey because of the relatively large volume of data generated for a given amount of computational work. In an effort to reduce computing resource waste, and to improve sustained performance of these HEC applications, a lightweight software utility has been designed to circumvent bandwidth limitations of typical HEC file systems by exploiting the faster inter-processor bandwidth to move output data from compute nodes to designated I/O nodes as quickly as possible, thereby minimizing the I/O wait time. This utility has successfully demonstrated a significant performance improvement within a major NASA weather application.

This work was done by Thomas Clune of Goddard Space Flight Center and Shujia Zhou of Northrop Grumman. For further information, contact the Goddard Innovative Partnerships Office at (301) 286-5810. GSC-15751-1

Planetary Image Geometry Library

The Planetary Image Geometry (PIG) library is a multi-mission library used for projecting images (EDRs, or Experiment Data Records) and managing their geometry for *in-situ* missions. A collection of models describes cameras and their articulation, allowing application programs such as mosaickers, terrain generators, and pointing correction tools to be written in a multi-mission manner, without any knowledge of parameters specific to the supported missions.

Camera model objects allow transformation of image coordinates to and from view vectors in XYZ space. Pointing models, specific to each mission, describe how to orient the camera models based on telemetry or other information. Surface models describe the surface in general terms. Coordinate system objects manage the various coordinate systems involved in most missions. File objects manage access to metadata (labels, including telemetry information) in the input EDRs and RDRs (Reduced Data Records). Label models manage metadata information in output files. Site objects keep track of different locations where the spacecraft might be at a given time. Radiometry models allow correction of radiometry for an image. Mission objects contain basic mission parameters. Pointing adjustment ("nav") files allow pointing to be corrected.

The object-oriented structure (C++) makes it easy to subclass just the pieces of the library that are truly mission-specific. Typically, this involves just the pointing model and coordinate systems, and parts of the file model. Once the library was developed (initially for Mars Polar Lander, MPL), adding new missions ranged from two days to a few months, resulting in significant cost savings as compared to rewriting all the application programs for each mission. Currently supported missions include Mars Pathfinder (MPF), MPL, Mars Exploration Rover (MER), Phoenix, and Mars Science Lab (MSL). Applications based on this library create the majority of operational image RDRs for those missions. A Java wrapper around the library allows parts of it to be used from Java code (via a native JNI interface). Future conversions of all or part of the library to Java are contemplated.

This work was done by Robert G. 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 Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-46658.