Software

Processing AIRS Scientific Data Through Level 3

The Atmospheric Infra-Red Sounder (AIRS) Science Processing System (SPS) is a collection of computer programs, known as product generation executives (PGEs). The AIRS SPS PGEs are used for processing measurements received from the AIRS suite of infrared and microwave instruments orbiting the Earth onboard NASA's Aqua spacecraft. Early stages of the AIRS SPS development were described in a prior *NASA Tech Briefs* article: "Initial Processing of Infrared Spectral Data" (NPO-35243), Vol. 28, No. 11 (November 2004), page 39.

In summary: Starting from Level 0 (representing raw AIRS data), the AIRS SPS PGEs and the data products they produce are identified by alphanumeric labels (1A, 1B, 2, and 3) representing successive stages or levels of processing. The previous *NASA Tech Briefs* article described processing through Level 2, the output of which comprises geo-located atmospheric data products such as temperature and humidity profiles among others. The AIRS Level 3 PGE samples selected information from the Level 2 standard products to produce a single global gridded product. One Level 3 product is generated for each day's collection of Level 2 data. In addition, daily Level 3 products are aggregated into two multi-day products: an eight-day (half the orbital repeat cycle) product and monthly (calendar month) product.

This work was done by Stephanie Granger, Robert Oliphant, and Evan Manning 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-42146.

Web-Based Requesting and Scheduling Use of Facilities

Automated User's Training Operations Facility Utilization Request (AutoFUR) is prototype software that administers a Web-based system for requesting and allocating facilities and equipment for astronaut-training classes in conjunction with scheduling the classes. AutoFUR also has potential for similar use in such applications as scheduling flight-simulation equipment and instructors in commercial airplane-pilot training, managing preventive-maintenance facilities, and scheduling operating rooms, doctors, nurses, and medical equipment for surgery.

Whereas requesting and allocation of facilities was previously a manual process that entailed examination of documents (including paper drawings) from different sources, AutoFUR partly automates the process and makes all of the relevant information available via the requester's computer. By use of AutoFUR, an instructor can fill out a facility-utilization request (FUR) form on line, consult the applicable flight manifest(s) to determine what equipment is needed and where it should be placed in the training facility, reserve the corresponding hardware listed in a training-hardware inventory database, search for alternative hardware if necessary, submit the FUR for processing, and cause paper forms to be printed. AutoFUR also maintains a searchable archive of prior FURs.

This program was written by Carolyn M. Yeager of Aptek, Inc., for Johnson Space Center. Further information is contained in a TSP (see page 1). MSC-23247-1

AutoGen Version 5.0

Version 5.0 of the AutoGen software has been released. Previous versions, variously denoted "Autogen" and "autogen," were reported in two articles: "Automated Sequence Generation Process and Software" (NPO-30746), *Software Tech Briefs* (Special Supplement to *NASA Tech Briefs*), September 2007, page 30, and "Autogen Version 2.0" (NPO-41501), *NASA Tech Briefs*, Vol. 31, No. 10 (October 2007), page 58.

To recapitulate: AutoGen (now signifying "automatic sequence generation") automates the generation of sequences of commands in a standard format for uplink to spacecraft. AutoGen requires fewer workers than are needed for older manual sequence-generation processes, and greatly reduces sequence-generation times.

The sequences are embodied in spacecraft activity sequence files (SASFs). AutoGen automates generation of SASFs by use of another previously reported program called "APGEN." AutoGen encodes knowledge of different mission phases and of how the resultant commands must differ among the phases. AutoGen also provides means for customizing sequences through use of configuration files. The approach followed in developing AutoGen has involved encoding the behaviors of a system into a model and encoding algorithms for context-sensitive customizations of the modeled behaviors.

This version of AutoGen addressed the MRO (Mars Reconnaissance Orbiter) primary science phase (PSP) mission phase. On previous Mars missions this phase has more commonly been referred to as mapping phase. This version addressed the unique aspects of sequencing orbital operations and specifically the mission specific adaptation of orbital operations for MRO. This version also includes capabilities for MRO's role in Mars relay support for UHF relay communications with the MER rovers and the Phoenix lander.

This program was written by Roy E. Gladden, Teerapat Khanampornpan, and Forest W. Fisher 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-45984

Time-Tag Generation Script

Time-Tag Generation Script (TTaGS) is an application program, written in the AWK scripting language, for generating commands for aiming one Ku-band antenna and two S-band antennas for communicating with spacecraft. TTaGS saves between 2 and 4 person-hours per every 24 hours by automating the repetitious process of building between 150 and 180 antenna-control commands. TTaGS reads a text database of communication-satellite schedules and a text database of satellite rise and set times and cross-references items in the two databases. It then compares the scheduled start and stop with the geometric rise and set to compute the times to execute antenna control commands. While so doing, TTaGS determines whether to generate commands for guidance, navigation,

and control computers to tell them which satellites to track.

To help prevent Ku-band irradiation of the Earth, TTaGS accepts input from the user about horizon tolerance and accordingly restricts activation and effects deactivation of the transmitter. TTaGS can be modified easily to enable tracking of additional satellites and for such other tasks as reading Sun-rise/set tables to generate commands to point the solar photovoltaic arrays of the International Space Station at the Sun.

This program was written by Dan E. Jackson of Barrios Technology for Johnson Space Center. For further information, contact the Johnson Commercial Technology Office at (281) 483-3809. MSC-23588-1

PPM Receiver Implemented in Software

A computer program has been written as a tool for developing optical pulse-position-modulation (PPM) receivers in which photodetector outputs are fed to analog-to-digital converters (ADCs) and all subsequent signal processing is performed digitally. The program can be used, for example, to simulate an all-digital version of the PPM receiver described in "Parallel Processing of Broad-Band PPM Signals" (NPO-40711), which appears elsewhere in this issue of *NASA Tech Briefs*. The program can also be translated into a design for digital PPM-receiver hardware.

The most notable innovation embodied in the software and the underlying PPM-reception concept is a digital processing subsystem that performs synchronization of PPM time slots, even though the digital processing is, itself, asynchronous in the sense that no attempt is made to synchronize it with the incoming optical signal *a priori* and there is no feedback to analog signal-processing subsystems or ADCs. Functions performed by the software receiver include time-slot synchronization, symbol synchronization, coding preprocessing, and diagnostic functions. The program is written in the MATLAB® and Simulink® software system. The software receiver is highly parameterized and, hence, programmable: for example, slot- and symbol-synchronization filters have programmable bandwidths.

This program was written by Andrew Gray, Edward Kang, Norman Lay, Victor Vilnrotter, Meera Srinivasan, and Clement Lee of Caltech for NASA's Jet Propulsion Laboratory.

In accordance with Public Law 96-517, the contractor has elected to retain title to this invention. Inquiries concerning rights for its commercial use should be addressed to:

Innovative Technology Assets Management
JPL
Mail Stop 202-233
4800 Oak Grove Drive
Pasadena, CA 91109-8099
E-mail: iaoffice@jpl.nasa.gov

Refer to NPO-40712, volume and number of this NASA Tech Briefs *issue, and the page number.*

Tropospheric Emission Spectrometer Product File Readers

TES Product File Reader software extracts data from publicly available Tropospheric Emission Spectrometer (TES) HDF (Hierarchical Data Format) product data files using publicly available format specifications for scientific analysis in IDL (interactive data language). In this innovation, the software returns data fields as simple arrays for a given file. A file name is provided, and the contents are returned as simple IDL variables.

This work was done by Brendan M. Fisher of Caltech for NASA's Jet Propulsion Laboratory. For more information, see http://tes.jpl.nasa.gov/.

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-47000.

Reporting Differences Between Spacecraft Sequence Files

A suite of computer programs, called "seq diff suite," reports differences between the products of other computer programs involved in the generation of sequences of commands for spacecraft. These products consist of files of several types: replacement sequence of events (RSOE), DSN keyword file [DKF (wherein "DSN" signifies "Deep Space Network)], spacecraft activities sequence file (SASF), spacecraft sequence file (SSF), and station allocation file (SAF). These products can include line numbers, request identifications, and other pieces of information that are not relevant when generating command sequence products, though these fields can result in the appearance of many changes to the files, particularly when using the UNIX diff command to inspect file differences. The outputs of prior software tools for reporting differences between such products include differences in these non-relevant pieces of information.

In contrast, seq diff suite removes the fields containing the irrelevant pieces of information before processing to extract differences, so that only relevant differences are reported. Thus, seq diff suite is especially useful for reporting changes between successive versions of the various products and in particular flagging difference in fields relevant to the sequence command generation and review process.

This program was written by Teerapat Khanampornpan, Roy E. Gladden, and Forest W. Fisher 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-45438.

Coordinating "Execute" Data for ISS and Space Shuttle

The Joint Execute Package Development and Integration tool is a Web utility program that provides an integrated capability to generate and manage messages and "execute" package data for members of a space shuttle and the International Space Station (ISS). (An "execute" package consists of flight plans, short-term plans, procedure updates, data needed to operate the space-shuttle and ISS systems, in-flight maintenance procedures, inventory-stowage data, software upgrades, flight notes, scripts for publicized events, and other instructions.) This program is a third-generation "execute"-package Web tool, built on experience gained from two programs used previously to support realtime operations.

This program provides integration and synchronization between the space-shuttle and ISS teams during joint operations. Hundreds of messages per week must be uplinked as "joint" messages; that is, messages for crewmembers of both spacecraft. The program includes configuration-management components that ensure that the same message goes to both crews and spacecraft, effectively eliminating the potential for error in manual direction of messages. The program also controls the format and layout of the crews' Web pages, ensuring consistency between uplinks. If the crews' Web pages were edited man-