Software

Software Framework for Peer Data-Management Services

Object Oriented Data Technology (OODT) is a software framework for creating a Web-based system for exchange of scientific data that are stored in diverse formats on computers at different sites under the management of scientific peers. OODT software consists of a set of cooperating, distributed peer components that provide distributed peer-to-peer (P2P) services that enable one peer to search and retrieve data managed by another peer. In effect, computers running OODT software at different locations become parts of an integrated data-management system.

OODT now incorporates a client/server communication substrate, but in other respects, its design resembles that of a P2P network, and it is planned to make a transition to a P2P communication substrate in the near future. OODT uses standard Transmission Control Protocol/ Internet Protocol (TCP/IP) connections. The architecture of OODT is that of a plug-in system. The OODT framework includes a set of classes and interfaces that can be customized and then registered with an application programmer's interface. The classes and interfaces tell the programmer at each site exactly what is needed for customization.

This program was written by John Hughes, Sean Hardman, Daniel Crichton, and Jason Hyon of Caltech and Sean Kelly and Thuy Tran of Northrop Grumman 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-40370.

Autogen Version 2.0

Version 2.0 of the autogen software has been released. "Autogen" (automated sequence generation) signifies both a process and software used to implement the process of automated 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 reduces sequence-generation times from weeks to minutes. The autogen software comprises the autogen script plus the Activity Plan Generator (APGEN) program. APGEN can be used for planning missions and command sequences. APGEN generates a graphical user interface that facilitates scheduling of activities on a time line and affords a capability to automatically expand, decompose, and schedule activities. The earlier version of the autogen software was developed for the Mars 2001 *Odyssey* spacecraft. Version 2.0 offers enhanced capabilities to serve, simultaneously, multiple spacecraft (including the Mars Global Surveyor, the Mars Exploration Rovers, and the future Mars Reconnaissance Orbiter) that may be at different phases of their missions (including cruise, aerobraking, mapping, and relay operations).

This work was done by Roy Gladden 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-41501.

Tracking-Data-Conversion Tool

A computer program denoted Tracking Data Delivery Software Orbit Data File (TDDSODF) converts deep-space-radio-communication spacecraft-tracking data from a currently used file format known in the art as "TRK-2-34" to a legacy format denoted "TRK-2-18." TDDSODF reads standard formatted data units (SFDUs) of several TRK-2-34 types and processes them into an orbit data file (ODF) containing data of one or more of several different TRK-2-18 types. TDDSODF offers the user the following options:

- To set processing parameters (including default values for use when TRK-2-34 values are not available) through configuration files or from a command line,
- To specify a compression interval used in the generation of carrier-frequency observables from TRK-2-34 downlink-carrier-phase and Doppler-count SFDUs,
- To fix the formulation of observables to a time-tag that is an integer multiple of a user-specified interval,
- To verify whether the configuration files contain valid keywords or out-of-range keyword values, and
- To specify both required and optional command-line parameters in a single file.

This program was written by Dana Flora-Adams, Jeanne Makihara, Zabel Benenyan, Jeff Berner, and Andrew Kwok 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-41741.

NASA Enterprise Visual Analysis

NASA Enterprise Visual Analysis (NEVA) is a computer program undergoing development as a successor to Launch Services Analysis Tool (LSAT), formerly known as Payload Carrier Analysis Tool (PCAT). NEVA facilitates analyses of proposed configurations of payloads and packing fixtures (e.g. pallets) in a space-shuttle payload bay for transport to the International Space Station. NEVA reduces the need to use physical models, mock-ups, and full-scale ground support equipment in performing such analyses. Using NEVA, one can take account of such diverse considerations as those of weight distribution, geometry, collision avoidance, power requirements, thermal loads, and mechanical loads.

NEVA accepts mass-property data from computational models of payloads, carriers, and interfaces, and uses these data to perform weight and center-of-gravity analyses. NEVA accepts results from structural-, thermal-, and fluid-analysis programs and translates them for incorporation into visual displays along with the results of the weight-distribution analyses. After contemplated further development, NEVA will also be able to accept, translate, and display results of communication- and electromagnetic-compatibility-analysis programs. Thus, NEVA is expected to continue to evolve into an increasingly capable tool for supporting technical and management decisions regarding ever more complex payload configurations.

This program (copyright © The Boeing Company 2005, all rights reserved) was written by Maria Lopez-Tellado of Kennedy Space Center and Brenda DiSanto, Robert Humeniuk, Richard Bard Jr., Mia Little, Robert Edwards, Tien-Chi Ma, Kenneth

Hollifield, and Chuck White of The Boeing Co. GSA-23F-0183K, Order No. NNK04MB84D.

Inquiries concerning licenses for its commercial development should be addressed to:

Terrance Mason, Boeing Patent Licensing Professional
Boeing Management Company
Mail Stop 1650-7002 Laguna Canyon Road
Irvine, CA 92618
Phone No.: (949) 790-1331
E-mail: terrance.mason@boeing.com
Refer to Boeing Docket No 03-1523
Web site: http://www.boeing.com
KSC-12712

Advanced Reference Counting Pointers for Better Performance

A computer program implements reference counting pointers (RCPs) that are lock-free, thread-safe, async-safe, and operational on a multiprocessor computer. RCPs are powerful and convenient means of managing heap memory in C++ software. Most prior RCP programs use locks to ensure thread safety and manage concurrency. The present program was developed in a continuing effort to explore ways of using the C++ programming language to develop safety-critical and mission-critical software.

This effort includes exploration of lock-free algorithms because they offer potential to avoid some costly and difficult verification problems. Unlike previously published RCP software, the present program does not use locks (meaning that no thread can block progress on another thread): Instead, this program implements algorithms that exploit capabilities of central-processing-unit hardware so as to avoid locks. Once locks are eliminated, it becomes possible to realize the other attributes mentioned in the first sentence. In addition to the abovementioned attributes, this program offers several advantages over other RCP programs that use locks: It is smaller (and, hence, is faster and uses less memory), it is immune to priority inversion, and there is no way for it to cause a C++ exception.

This program was written by William Reinholtz 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
(818) 354-2240
E-mail: iaoffice@jpl.nasa.gov

Refer to NPO-41196, volume and number of this NASA Tech Briefs *issue, and the page number.*

C Namelist Facility

C Namelist Facility (CNL) is a package of software that supports the development of data-driven programs that utilize relatively free-form input files (e.g., text files) to control complex operations. The only comparable prior namelist facility is built into Fortran and does not support arrays or records. Newer computing languages, including C and Pascal, do not include built-in namelist facilities. A namelist facility enables a program to utilize relatively free-form input files that contain assignment statements that give values to variables. Variables to which values are not assigned in input files remain unchanged; therefore, it becomes possible to have default values set by static or dynamic initialization of values prior to namelist input and updating of values is optional. Because it is not required to include values of variables in namelist input files, new parameters can be added to evolving programs without rendering old namelist input files obsolete — provided that the new parameters have useful default values. It should be possible to execute CNL in any operating system that supports the ANSI C programming language. It has been executed in several variants of Unix and in VxWorks.

This program was written by Bruce Bon 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-40087.

Efficient Mosaicking of Spitzer Space Telescope Images

A parallel version of the MOPEX software, which generates mosaics of infrared astronomical images acquired by the Spitzer Space Telescope, extends the capabilities of the prior serial version. In the parallel version, both the input image space and the output mosaic space are divided among the available parallel processors. This is the only software that performs the point-source detection and the rejection of spurious imaging effects of cosmic rays required by Spitzer scientists. This software includes components that implement outlier-detection algorithms that can be fine-tuned for a particular set of image data by use of a number of adjustable parameters.

This software has been used to construct a mosaic of the Spitzer Infrared Array Camera Shallow Survey, which comprises more than 17,000 exposures in four wavelength bands from 3.6 to 8 μm and spans a solid angle of about 9 square degrees. When this software was executed on 32 nodes of the 1,024-processor Cosmos cluster computer at NASA's Jet Propulsion Laboratory, a speedup of 8.3 was achieved over the serial version of MOPEX. The performance is expected to improve dramatically once a true parallel file system is installed on Cosmos.

This program was written by Joseph Jacob, David Makovoz, and Peter Eisenhardt 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-42860.