Software

The PICWidget

The Plug-in Image Component Widget (PICWidget) is a software component for building digital imaging applications. The component is part of a methodology described in "GIS Methodology for Planning Planetary-Rover Operations" (NPO-41812), which appears elsewhere in this issue of *NASA Tech Briefs*. Planetary rover missions return a large number and wide variety of image data products that vary in complexity in many ways. Supported by a powerful, flexible image-data-processing pipeline, the PICWidget can process and render many types of imagery, including (but not limited to) thumbnail, subframed, downsampled, stereoscopic, and mosaic images; images coregistred with orbital data; and synthetic red/green/blue images. The PICWidget is capable of efficiently rendering images from data representing many more pixels than are available at a computer workstation where the images are to be displayed. The PICWidget is implemented as an Eclipse plug-in using the Standard Widget Toolkit, which provides a straightforward interface for re-use of the PICWidget in any number of application programs built upon the Eclipse application framework. Because the PICWidget is tile-based and performs aggressive tile caching, it has flexibility to perform faster or slower, depending whether more or less memory is available.

This work was done by Jeffrey Norris, Jason Fox, Kenneth Rabe, I-Hsiang Shu, and Mark Powell 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-41813.*

Fusing Symbolic and Numerical Diagnostic Computations

"X-2000 Anomaly Detection Language" denotes a developmental computing language, and the software that establishes and utilizes the language, for fusing two diagnostic computer programs, one implementing a numerical analysis method, the other implementing a symbolic analysis method into a unified event-based decision analysis software system for real-time detection of events (e.g., failures) in a spacecraft, aircraft, or other complex engineering system. The numerical analysis method is performed by beacon-based exception analysis for multi-missions (BEAMs), which has been discussed in several previous *NASA Tech Briefs* articles. The symbolic analysis method is, more specifically, an artificial-intelligence method of the knowledge-based, inference engine type, and its implementation is exemplified by the Spacecraft Health Inference Engine (SHINE) software. The goal in developing the capability to fuse numerical and symbolic diagnostic components is to increase the depth of analysis beyond that previously attainable, thereby increasing the degree of confidence in the computed results. In practical terms, the sought improvement is to enable detection of all or most events, with no or few false alarms.

This program was written by Mark James 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-42512.

Probabilistic Reasoning for Robustness in Automated Planning

A general-purpose computer program for planning the actions of a spacecraft or other complex system has been augmented by incorporating a subprogram that reasons about uncertainties in such continuous variables as times taken to perform tasks and amounts of resources to be consumed. This subprogram computes parametric probability distributions for time and resource variables on the basis of user-supplied models of actions and resources that they consume. The current system accepts bounded Gaussian distributions over action duration and resource use. The distributions are then combined during planning to determine the net probability distribution of each resource at any time point. In addition to a full combinatoric approach, several approximations for arriving at these combined distributions are available, including maximum-likelihood and pessimistic algorithms. Each such probability distribution can then be integrated to obtain a probability that execution of the plan under consideration would violate any constraints on the resource. The key idea is to use these probabilities of conflict to score potential plans and drive a search toward planning low-risk actions. An output plan provides a balance between the user's specified averseness to risk and other measures of optimality.

This program was written by Steven Schaffer, Bradley Clement, and Steve Chien 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-42152.

Short-Term Forecasting of Radiation Belt and Ring Current

A computer program implements a mathematical model of the radiation-belt and ring-current plasmas resulting from interactions between the solar wind and the Earth's magnetic field, for the purpose of predicting fluxes of energetic electrons (10 keV to 5 MeV) and protons (10 keV to 1 MeV), which are hazardous to humans and spacecraft. Given solar-wind and interplanetary-magnetic-field data as inputs, the program solves the convection-diffusion equations of plasma distribution functions in the range of 2 to 10 Earth radii. Phenomena represented in the model include particle drifts resulting from the gradient and curvature of the magnetic field; electric fields associated with the rotation of the Earth, convection, and temporal variation of the magnetic field; and losses along particle-drift paths. The model can readily accommodate new magnetic- and electric-field submodels and new information regarding physical processes that drive the radiation-belt and ring-current plasmas. Despite the complexity of the model, the program can be run in real time on ordinary computers. At present, the program can calculate present electron and proton fluxes; after further development, it should be able to predict the fluxes 24 hours in advance.

This program was written by George V. Khazanov of **Marshall Space Flight Center** *and Mei-Ching Fok of Goddard Space Flight Center. Further information is contained in a TSP (see page 1).MFS-32128-1*

JMS Proxy and C/C++ Client SDK

JMS Proxy and C/C++ Client SDK ("JMS" signifies "Java messaging service" and "SDK" signifies "software development kit") is a software package for developing interfaces that enable legacy programs (here denoted "clients") written in the C and C++ languages to communicate with each other via a JMS broker. This package consists of two main components: the JMS proxy server component and the client C library SDK component. The JMS proxy server component implements a native Java process that receives and responds to requests from clients. This component can run on any computer that supports Java and a JMS client. The client C library SDK component is used to develop a JMS client program running in each affected C or C++ environment, without need for running a Java virtual machine in the affected computer. A C client program developed by use of this SDK has most of the quality-of-service characteristics of standard Java-based client programs, including the following:

- Durable subscriptions;
- Asynchronous message receipt;
- Such standard JMS message qualities as "TimeToLive," "Message Properties," and "DeliveryMode" (as the quoted terms are defined in previously published JMS documentation); and
- Automatic reconnection of a JMS proxy to a restarted JMS broker.

This program was written by Paul Wolgast and Paul Pechkam 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-42527.

XML Flight/Ground Data Dictionary Management

A computer program generates Extensible Markup Language (XML) files that effect coupling between the command- and telemetry-handling software running aboard a spacecraft and the corresponding software running in ground support systems. The XML files are produced by use of information from the flight software and from flight-system engineering. The XML files are converted to legacy ground-system data formats for command and telemetry, transformed into Web-based and printed documentation, and used in developing new ground-system data-handling software. Previously, the information about telemetry and command was scattered in various paper documents that were not synchronized. The process of searching and reading the documents was time-consuming and introduced errors. In contrast, the XML files contain all of the information in one place. XML structures can evolve in such a manner as to enable the addition, to the XML files, of the metadata necessary to track the changes and the associated documentation. The use of this software has reduced the extent of manual operations in developing a ground data system, thereby saving considerable time and removing errors that previously arose in the translation and transcription of software information from the flight to the ground system.

This program was written by Jesse Wright and Colette Wiklow 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-42291.

Cross-Compiler for Modeling Space-Flight Systems

Ripples is a computer program that makes it possible to specify arbitrarily complex space-flight systems in an easy-to-learn, high-level programming language and to have the specification automatically translated into LibSim, which is a text-based computing language in which such simulations are implemented. LibSim is a very powerful simulation language, but learning it takes considerable time, and it requires that models of systems and their components be described at a very low level of abstraction. To construct a model in LibSim, it is necessary to go through a time-consuming process that includes modeling each subsystem, including defining its fault-injection states, input and output conditions, and the topology of its connections to other subsystems. Ripples makes it possible to describe the same models at a much higher level of abstraction, thereby enabling the user to build models faster and with fewer errors. Ripples can be executed in a variety of computers and operating systems, and can be supplied in either source code or binary form. It must be run in conjunction with a Lisp compiler.

This program was written by Mark James 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-42532.