⊵ Displaying Properties of PDFs

Ames Research Center, Moffett Field, California

PDFVis is a computer program that assists in visualization of uncertainty as represented by a probability density function (PDF) located at each grid cell in a spatial domain. PDFVis performs the following functions:

- Creates maps of first-, second-, and third-order statistics of the PDFs, displaying the maps simultaneously in one image;
- Provides an interactive probe that displays various forms of density estimates (e.g., histogram or kernel density estimator) of the PDF at a grid cell or larger region selected by the user;
- For grid cells in a row or column selected by the user, it displays surfaces of the PDFs as shaded walls and as color-mapped wall according to a scalar variable selectable by the user;
- Displays the modality (basically, the number of peaks) and the locality of the peaks of the PDF in each grid cell;
- Provides a capability for querying the modalities of the PDFs on the basis of ranges and other criteria specified by the user; and
- Facilitates interactivity by providing simple user command keys.

This program was written by David Kao and Jennifer Dungan of Ames Research Center and David Kao of The Regents of the University of California Santa Cruz. Further information is contained in a TSP (see page 1).

Inquiries concerning rights for the commercial use of this invention should be addressed to the Ames Technology Partnerships Division at (650) 604-2954. Refer to ARC-15039-1

⊵ Modular Filter and Source-Management Upgrade of RADAC

Goddard Space Flight Center, Greenbelt, Maryland

In an upgrade of the Range Data Acquisition Computer (RADAC) software, a modular software object library was developed to implement required functionality for filtering of flight-vehicle-tracking data and management of tracking-data sources. (The RADAC software is used to process flight-vehicle metric data for real-time display in the Wallops Flight Facility Range Control Center and Mobile Control Center. This software is part of the Range Control/Range Safety System used to contain impacts of debris from flight-vehicle operations.) The library includes implementations of tracking-data-editing and -filtering techniques that can be readily found in the literature and was constructed to facilitate future incorporation of implementations of other, more-exotic filtering techniques. A pragmatic technique is used to maintain statistics on differences between measurements and estimates of positions in order to compute a figure of merit for each radar, Global-Positioning-System, or inertial-navigation-system source of tracking data.

An automatic source-qualification technique is implemented to enable optional replacement of a pre-existing manual data-qualification function. The library was written in the C++ language and was designed to be hosted within an application program running on virtually any computing platform.

This program was written by R. James Lanzi and Donna C. Smith of Goddard Space Flight Center. Further information is contained in a TSP (see page 1).
GSC-14974-1

⊵ Automatic Command Sequence Generation

NASA's Jet Propulsion Laboratory, Pasadena, California

Automatic Sequence Generator (Autogen) Version 3.0 software automatically generates command sequences for the Mars Reconnaissance Orbiter (MRO) and several other JPL spacecraft operated by the multi-mission support team. Autogen uses standard JPL sequencing tools like APGEN, ASP, SEQGEN, and the DOM database to automate the generation of uplink command products, Spacecraft Command Message Format (SCMF) files, and the corresponding ground command products, DSN Keywords Files (DKF). Autogen supports all the major multi-mission mission phases including the cruise, aerobraking, mapping/science, and relay mission phases.

Autogen is a Perl script, which functions within the 'mission operations' UNIX environment. It consists of two parts: a set of model files and the autogen Perl script. Autogen encodes the behaviors of the system into a model and encodes algorithms for context-sensitive customizations of the modeled behaviors. The model includes knowledge of different mission phases and how the resultant command products must differ for these phases. The executable software portion of Autogen, automates the setup and use of APGEN for constructing a spacecraft activity sequence file (SASF). The setup includes file retrieval through the DOM (Distributed Object Manager), an object database used to store project files. This step retrieves all the needed input files for generating the command products.

Depending on the mission phase, Autogen also uses the ASP (Automated Sequence Processor) and SEQGEN to generate the command product sent to the spacecraft. Autogen also provides the means for customizing sequences through the use of configuration files. By automating the majority of the sequencing generation process, Autogen eliminates many sequence generation errors commonly introduced by manually constructing spacecraft command sequences. Through the layering of commands into the sequence by a series of scheduling algorithms, users are able to rapidly and reliably construct the desired uplink command products.

With the aid of Autogen, sequences may be produced in a matter of hours instead of weeks, with a significant reduc-

tion in the number of people on the sequence team. As a result, the uplink product generation process is significantly streamlined and mission risk is significantly reduced. Autogen is used for operations of MRO, Mars Global Surveyor (MGS), Mars Exploration Rover (MER), Mars Odyssey, and will be used for operations of Phoenix. Autogen Version 3.0 is the operational version of Autogen including the MRO adaptation for the cruise mission phase, and was also used for development of the aerobraking and mapping mission phases for MRO.

Generating Scenarios When Data Are Missing

NASA's Jet Propulsion Laboratory, Pasadena, California

A computer program implements the algorithm described in "Hypothetical Scenario Generator for Fault-Tolerant Diagnosis" (NPO-42516), *NASA Tech Briefs,* Vol. 31, No. 6 (June 2007), page 71. To recapitulate: the Hypothetical Scenario Generator (HSG) is being developed in conjunction with other components of artificial-intelligence systems for automated diagnosis and prognosis of faults in spacecraft, aircraft, and other complex engineering systems. The HSG accepts, as input, possibly incomplete data on the current state of a system (see figure).

The HSG models a potential fault scenario as an ordered disjunctive tree of conjunctive consequences, wherein the ordering is based upon the likelihood that a particular conjunctive path will be taken for the given set of inputs. The computation of likelihood is based partly on a numerical ranking of the degree of completeness of data with respect to satisfaction of the antecedent conditions of prognostic rules. The results from the HSG are then used by a model-based artificial-intelligence subsystem to predict realistic scenarios and states.

End-to-End Prognostic Architecture uses existing diagnostic models to generate predictions.

CASPER Version 2.0

NASA's Jet Propulsion Laboratory, Pasadena, California

The Continuous Activity Scheduling Planning Execution and Replanning (CASPER) computer program has been updated to version 2.0. A prototype version was reported in "Software for Continuous Replanning During Execution" (NPO-20972), *NASA Tech Briefs,* Vol. 26, No. 7 (April 2002), page 67. To recapitulate: CASPER is designed to perform automated planning of interdependent activities within a system subject to requirements, constraints, and limitations on resources.

In contradistinction to the traditional concept of batch planning followed by execution, CASPER implements a concept of continuous planning and replanning in response to unanticipated changes (including failures), integrated with execution. Improvements over other, similar software that have been incorporated into CASPER version 2.0 include an enhanced executable interface to facilitate integration with a wide range of execution software systems and supporting software libraries; features to support execution while reasoning about urgency, importance, and impending deadlines; features that enable accommodation to a wide range of computing environments that include various central processing units and random-access-memory capacities; and improved generic time-server and time-control features.