Software

Translator for Optimizing Fluid-Handling Components

A software interface has been devised to facilitate optimization of the shapes of valves, elbows, fittings, and other components used to handle fluids under extreme conditions. This software interface translates data files generated by PLOT3D (a NASA grid-based plotting-and-data-display program) and by computational fluid dynamics (CFD) software into a format in which the files can be read by Sculptor, which is a shape-deformation-and-optimization program. Sculptor enables the user to interactively, smoothly, and arbitrarily deform the surfaces and volumes in two- and three-dimensional CFD models. Sculptor also includes design-optimization algorithms that can be used in conjunction with the arbitrary-shape-deformation components to perform automatic shape optimization. In the optimization process, the output of the CFD software is used as feedback while the optimizer strives to satisfy design criteria that could include, for example, improved values of pressure loss, velocity, flow quality, mass flow, etc.

This program was written by Mark Landon and Ernest Perry of Optimal Solutions Software, LLC for Stennis Space Center.

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:

Optimal Solutions Software, LLC
Attn. Mark Landon, President
2825 West 1700 North
Idaho Falls, ID 83402
Phone No.: (208) 521-4660
E-mail: mlandon@optimalsolutions.us
Refer to SSC-00190, volume and number of this NASA Tech Briefs *issue, and the page number.*

AIRSAR Web-Based Data Processing

The AIRSAR automated, Web-based data processing and distribution system is an integrated, end-to-end synthetic aperture radar (SAR) processing system. Designed to function under limited resources and rigorous demands, AIRSAR eliminates operational errors and provides for paperless archiving. Also, it provides a yearly tune-up of the processor on flight missions, as well as quality assurance with new radar modes and anomalous data compensation.

The software fully integrates a Web-based SAR data-user request subsystem, a data processing system to automatically generate co-registered multi-frequency images from both polarimetric and interferometric data collection modes in 80/40/20 MHz bandwidth, an automated verification quality assurance subsystem, and an automatic data distribution system for use in the remote-sensor community. Features include Survey Automation Processing in which the software can automatically generate a quick-look image from an entire 90-GB SAR raw data 32-MB/s tape overnight without operator intervention. Also, the software allows product ordering and distribution via a Web-based user request system.

To make AIRSAR more user friendly, it has been designed to let users search by entering the desired mission flight line (Missions Searching), or to search for any mission flight line by entering the desired latitude and longitude (Map Searching). For precision image automation processing, the software generates the products according to each data processing request stored in the database via a Queue management system. Users are able to have automatic generation of co-registered multi-frequency images as the software generates polarimetric and/or interferometric SAR data processing in ground and/or slant projection according to user processing requests for one of the 12 radar modes.

This program was written by Anhua Chu, Jakob Van Zyl, Yunjin Kim, Scott Hensley, Yunling Lou, Soren Madsen, Bruce Chapman, David Imel, and Stephen Durden of Caltech, and Wayne Tung of Columbus Technologies and Services, Inc. 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-40998.

Pattern Matcher for Trees Constructed From Lists

A software library has been developed that takes a high-level description of a pattern to be satisfied and applies it to a target. If the two match, it returns success; otherwise, it indicates a failure. The target is semantically a tree that is constructed from elements of terminal and non-terminal nodes represented through lists and symbols. Additionally, functionality is provided for finding the element in a set that satisfies a given pattern and doing a tree search, finding all occurrences of leaf nodes that match a given pattern. This process is valuable because it is a new algorithmic approach that significantly improves the productivity of the programmers and has the potential of making their resulting code more efficient by the introduction of a novel semantic representation language. This software has been used in many applications delivered to NASA and private industry, and the cost savings that have resulted from it are significant.

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-42732.

Reducing a Knowledge-Base Search Space When Data Are Missing

This software addresses the problem of how to efficiently execute a knowledge base in the presence of missing data. Computationally, this is an exponentially expensive operation that without heuristics generates a search space of $1 + 2^n$ possible scenarios, where n is the number of rules in the knowledge base. Even for a knowledge base of the most modest size, say 16 rules, it would produce 65,537 possible scenarios. The purpose of this software is to reduce the complexity of this operation to a more manageable size. The problem that this system solves is to develop an automated approach that can reason in the presence of missing data. This is a meta-reasoning capability that repeatedly calls a diagnostic engine/model to provide prognoses and prognosis tracking. In the big picture, the scenario generator takes as its input the current state of a system, including probabilistic information from Data Forecasting. Using model-based reasoning techniques, it returns an ordered list of fault scenarios that could be