Software

Software for Managing Parametric Studies

The Information Power Grid Virtual Laboratory (ILab) is a Practical Extraction and Reporting Language (PERL) graphical-user-interface computer program that generates shell scripts to facilitate parametric studies performed on the Grid. ("The Grid" denotes a worldwide network of supercomputers used for scientific and engineering computations involving data sets too large to fit on desktop computers.) Heretofore, parametric studies on the Grid have been impeded by the need to create control language scripts and edit input data files — painstaking tasks that are necessary for managing multiple jobs on multiple computers. ILab reflects an object-oriented approach to automation of these tasks: All data and operations are organized into packages in order to accelerate development and debugging. A "container" or "document" object in ILab, called an "experiment," contains all the information (data and file paths) necessary to define a complex series of repeated, sequenced, and/or branching processes. For convenience and to enable reuse, this object is serialized to and from disk storage. At run time, the current ILab experiment is used to generate required input files and shell scripts, create directories, copy data files, and then both initiate and monitor the execution of all computational processes.

This program was written by Maurice Yarrow, Karen M. McCann, and Adrian DeVivo of **Ames Research Center**. *For further information, access http://www.nas.nasa.gov/ILab/.*

Inquiries concerning rights for the commercial use of this invention should be addressed to the Patent Counsel, Ames Research Center, (650) 604-5104. Refer to ARC-14649-1.

Software Aids Visualization of Computed Unsteady Flow

Unsteady Flow Analysis Toolkit (UFAT) is a computer program that synthesizes motions of time-dependent flows represented by very large sets of data generated in computational fluid dynamics simulations. Prior to the development of UFAT, it was necessary to rely on static, single-snapshot depictions of time-dependent flows generated by flow-visualization software designed for steady flows. Whereas it typically takes weeks to analyze the results of a large-scale unsteady-flow simulation by use of steady-flow visualization software, the analysis time is reduced to hours when UFAT is used. UFAT can be used to generate graphical objects of flow visualization results using multi-block curvilinear grids in the format of a previously developed NASA data-visualization program, PLOT3D. These graphical objects can be rendered using FAST, another popular flow visualization software developed at NASA. Flow-visualization techniques that can be exploited by use of UFAT include time-dependent tracking of particles, detection of vortex cores, extractions of stream ribbons and surfaces, and tetrahedral decomposition for optimal particle tracking. Unique computational features of UFAT include capabilities for automatic (batch) processing, restart, memory mapping, and parallel processing. These capabilities significantly reduce analysis time and storage requirements, relative to those of prior flow-visualization software. UFAT can be executed on a variety of supercomputers.

This program was written by David Kao of **Ames Research Center** *and David Kenwright formerly of Computer Sciences Corp. 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 Patent Counsel, Ames Research Center, (650) 604-5104. Refer to ARC-14800.

Software for Testing Electroactive Structural Components

A computer program generates a graphical user interface that, in combination with its other features, facilitates the acquisition and preprocessing of experimental data on the strain response, hysteresis, and power consumption of a multilayer composite-material structural component containing one or more built-in sensor(s) and/or actuator(s) based on piezoelectric materials. This program runs in conjunction with LabVIEW software in a computer-controlled instrumentation system. For a test, a specimen is instrumented with applied-voltage and current sensors and with strain gauges. Once the computational connection to the test setup has been made via the LabVIEW software, this program causes the test instrumentation to step through specified configurations. If the user is satisfied with the test results as displayed by the software, the user activates an icon on a front-panel display, causing the raw current, voltage, and strain data to be digitized and saved. The data are also put into a spreadsheet and can be plotted on a graph. Graphical displays are saved in an image file for future reference. The program also computes and displays the power and the phase angle between voltage and current.

This program was written by Robert W. Moses, Robert L. Fox, Archie D. Dimery, Robert G. Bryant, and Qamar Shams of **Langley Research Center**, *and William C. White of Wyle Laboratories. Further information is contained in a TSP (see page 1).*
LAR-16546

Advanced Software for Analysis of High-Speed Rolling-Element Bearings

COBRA-AHS is a package of advanced software for analysis of rigid or flexible shaft systems supported by rolling-element bearings operating at high speeds under complex mechanical and thermal loads. These loads can include centrifugal and thermal loads generated by motions of bearing components. COBRA-AHS offers several improvements over prior commercial bearing-analysis programs: It includes innovative probabilistic fatigue-life-estimating software that provides for computation of three-dimensional stress fields and incorporates stress-based (in contradistinction to prior load-based) mathematical models of fatigue life. It interacts automatically with the ANSYS finite-element code to generate finite-element models for estimating distributions of temperature and temperature-induced changes in dimensions in iterative thermal/dimensional analyses: thus, for example, it can be used to predict changes in clearances and thermal lockup. COBRA-AHS provides an improved graphical user interface that facilitates the iterative cycle of analysis and design by providing analysis results quickly in graphical form, enabling the user to control interactive runs without leaving the program environment, and facilitating transfer of plots and printed results for inclusion in design reports. Additional features include roller-edge stress prediction and influence of shaft and housing distortion on bearing performance.

This program was written by J. V. Poplawski, J. H. Rumbarger, S. M. Peters, H. Galatis, and R. Flower of J. V. Poplawski & Associates for **Glenn Research Center**. *For further information, access www.bearingspecialists.com.*

Inquiries concerning rights for the commercial use of this invention should be addressed to NASA Glenn Research Center, Commercial Technology Office, Attn: Steve Fedor, Mail Stop 4-8, 21000 Brookpark Road, Cleveland Ohio 44135. Refer to LEW-17390.

Web Program for Development of GUIs for Cluster Computers

WIGLAF (a Web Interface Generator and Legacy Application Façade) is a computer program that provides a Web-based, distributed, graphical-user-interface (GUI) framework that can be adapted to any of a broad range of application programs, written in any programming language, that are executed remotely on any cluster computer system. WIGLAF enables the rapid development of a GUI for controlling and monitoring a specific application program running on the cluster and for transferring data to and from the application program. The only prerequisite for the execution of WIGLAF is a Web-browser program on a user's personal computer connected with the cluster via the Internet. WIGLAF has a client/server architecture: The server component is executed on the cluster system, where it controls the application program and serves data to the client component. The client component is an applet that runs in the Web browser. WIGLAF utilizes the Extensible Markup Language to hold all data associated with the application software, Java to enable platform-independent execution on the cluster system and the display of a GUI generator through the browser, and the Java Remote Method Invocation software package to provide simple, effective client/server networking.

This program was written by Akos Czikmantory, Thomas Cwik, Gerhard Klimeck, Hook Hua, Fabiano Oyafuso, and Edward Vinyard 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 Don Hart of the California Institute of Technology at (818) 393-3425. Refer to NPO-30842.

XML-Based Generator of C++ Code for Integration With GUIs

An open source computer program has been developed to satisfy a need for simplified organization of structured input data for scientific simulation programs. Typically, such input data are parsed in from a flat American Standard Code for Information Interchange (ASCII) text file into computational data structures. Also typically, when a graphical user interface (GUI) is used, there is a need to completely duplicate the input information while providing it to a user in a more structured form. Heretofore, the duplication of the input information has entailed duplication of software efforts and increases in susceptibility to software errors because of the concomitant need to maintain two independent input-handling mechanisms. The present program implements a method in which the input data for a simulation program are completely specified in an Extensible Markup Language (XML)-based text file. The key benefit for XML is storing input data in a structured manner. More importantly, XML allows not just storing of data but also describing what each of the data items are. That XML file contains information useful for rendering the data by other applications. It also then generates data structures in the C++ language that are to be used in the simulation program. In this method, all input data are specified in one place only, and it is easy to integrate the data structures into both the simulation program and the GUI. XML-to-C is useful in two ways:

1. As an executable, it generates the corresponding C++ classes and
2. As a library, it automatically fills the objects with the input data values.

This program was written by Hook Hua, Fabiano Oyafuso, and Gerhard Klimeck 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 Don Hart of the California Institute of Technology at (818) 393-3425. Refer to NPO-30844.