Software

Earth Observing System Data Gateway

The Earth Observing System Data Gateway (EDG) software provides a "one-stop-shopping" standard interface for exploring and ordering Earth-science data stored at geographically distributed sites. EDG enables a user to do the following:

• Search for data according to high-level criteria (e.g., geographic location, time, or satellite that acquired the data);

• Browse the results of a search, viewing thumbnail sketches of data that satisfy the user's criteria; and

• Order selected data for delivery to a specified address on a chosen medium (e.g., compact disk or magnetic tape).

EDG consists of (1) a component that implements a high-level client/server protocol, and (2) a collection of C-language libraries that implement the passing of protocol messages between an EDG client and one or more EDG servers. EDG servers are located at sites usually called "Distributed Active Archive Centers" (DAACs). Each DAAC may allow access to many individual data items, called "granules" (e.g., single Landsat images). Related granules are grouped into collections called "data sets." EDG enables a user to send a search query to multiple DAACs simultaneously, inspect the resulting information, select browseable granules, and then order selected data from the different sites in a seamless fashion.

This program was developed by Robin Pfister of **Goddard Space Flight Center** *and Joe McMahon, James Amrhein, Ed Sefert, Lorena Marsans, Mark Solomon, and Mark Nestler of Global Science & Technology, Inc. Further information is contained in a TSP (see page 1).*
GSC-14938-1

Power User Interface

Power User Interface 5.0 (PUI) is a system of middleware that is an alternative to the computer program described in the immediately preceding article. Written for expert users in the Earth-science community, PUI enables expedited ordering of data granules on the basis of specific granule-identifying information that the users already know or can assemble. PUI also enables expert users to perform quick searches for orderable-granule information for use in preparing orders. PUI 5.0 is available in two versions (note: PUI 6.0 has command-line mode only): a Web-based application program and a UNIX command-line-mode client program. Both versions include modules that perform data-granule-ordering functions in conjunction with external systems. The Web-based version works with Earth Observing System Clearing House (ECHO) metadata catalog and order-entry services and with an open-source order-service broker server component, called the Mercury Shopping Cart, that is provided separately by Oak Ridge National Laboratory through the Department of Energy. The command-line version works with the ECHO metadata and order-entry process service. Both versions of PUI ultimately use ECHO to process an order to be sent to a data provider. Ordered data are provided through means outside the PUI software system.

This program was developed by Robin Pfister of **Goddard Space Flight Center** *and Joe McMahon of Global Science & Technology, Inc. Further information is contained in a TSP (see page 1).*
GSC-14939-1

Mercury Shopping Cart Interface

Mercury Shopping Cart Interface (MSCI) is a reusable component of the Power User Interface 5.0 (PUI) program described in the immediately preceding article. MSCI is a means of encapsulating the logic and information needed to describe an orderable item consistent with Mercury Shopping Cart service protocol. Designed to be used with Web-browser software, MSCI generates Hypertext Markup Language (HTML) pages on which ordering information can be entered. MSCI comprises two types of Practical Extraction and Report Language (PERL) modules: template modules and shopping-cart logic modules. Template modules generate HTML pages for entering the required ordering details and enable submission of the order via a Hypertext Transfer Protocol (HTTP) post. Shop-ping-cart modules encapsulate the logic and data needed to describe an individual orderable item to the Mercury Shopping Cart service. These modules evaluate information entered by the user to determine whether it is sufficient for the Shopping Cart service to process the order. Once an order has been passed from MSCI to a deployed Mercury Shopping Cart server, there is no further interaction with the user.

This program was developed by Robin Pfister of **Goddard Space Flight Center** *and Joe McMahon of Global Science & Technology, Inc. Further information is contained in a TSP (see page 1).*
GSC-14940-1

Cassini Archive Tracking System

The Cassini Archive Tracking System (CATS) is a computer program that enables tracking of scientific data transfers from originators to the Planetary Data System (PDS) archives. Without CATS, there is no systematic means of locating products in the archive process or ensuring their completeness. By keeping a database of transfer communications and status, CATS enables the Cassini Project and the PDS to efficiently and accurately report on archive status. More importantly, problem areas are easily identified through customized reports that can be generated on the fly from any Web-enabled computer. A Web-browser interface and clearly defined authorization scheme provide safe distributed access to the system, where users can perform functions such as create customized reports, record a transfer, and respond to a transfer. CATS ensures that Cassini provides complete science archives to the PDS on schedule and that those archives are available to the science community by the PDS. The three-tier architecture is loosely coupled and designed for simple adaptation to multi-mission use. Written in the Java programming language, it is portable and can be run on any Java-enabled Web server.

This work was done by Diane Conner, Elias Sayfi, and Adrian Tinio of Caltech for **NASA's Jet Propulsion Laboratory**. *Further information is contained in a TSP (see page 1).*

Architecture Adaptive Computing Environment

Architecture Adaptive Computing Environment (aCe) is a software system that includes a language, compiler, and run-time library for parallel computing. aCe was developed to enable programmers to write programs, more easily than was previously possible, for a variety of parallel computing architectures. Heretofore, it has been perceived to be difficult to write parallel programs for parallel computers and more difficult to port the programs to different parallel computing architectures. In contrast, aCe is supportable on all high-performance computing architectures. Currently, it is supported on LINUX clusters. aCe uses parallel programming constructs that facilitate writing of parallel programs. Such constructs were used in single-instruction/multiple-data (SIMD) programming languages of the 1980s, including Parallel Pascal, Parallel Forth, C*, *LISP, and MasPar MPL. In aCe, these constructs are extended and implemented for both SIMD and multiple-instruction/multiple-data (MIMD) architectures. Two new constructs incorporated in aCe are those of (1) scalar and virtual variables and (2) pre-computed paths. The scalar-and-virtual-variables construct increases flexibility in optimizing memory utilization in various architectures. The pre-computed-paths construct enables the compiler to pre-compute part of a communication operation once, rather than computing it every time the communication operation is performed.

Computing Fault Displacements From Surface Deformations

Simplex is a computer program that calculates locations and displacements of subterranean faults from data on Earth-surface deformations. The calculation involves inversion of a forward model (given a point source representing a fault, a forward model calculates the surface deformations) for displacements, and strains caused by a fault located in isotropic, elastic half-space. The inversion involves the use of nonlinear, multiparameter estimation techniques. The input surface-deformation data can be in multiple formats, with absolute or differential positioning. The input data can be derived from multiple sources, including interferometric synthetic-aperture radar, the Global Positioning System, and strain meters. Parameters can be constrained or free. Estimates can be calculated for single or multiple faults. Estimates of parameters are accompanied by reports of their covariances and uncertainties. Simplex has been tested extensively against forward models and against other means of inverting geodetic data and seismic observations.