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 Pfiester of Goddard Space Flight Center and Joe McMahon of Global Science & Technology, Inc. Further information is contained in a TSP (see page 1).
GSC-14938-1

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. Shopping-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 Pfiester 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 Sryfi, and Adrian Timio of Caltech for NASA’s Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).