DEPLOYING OBJECT ORIENTED DATA TECHNOLOGY TO THE PLANETARY DATA SYSTEM.

S. Kelly[1], D. Crichton[2], and J.S. Hughes[2], [1]Independent Consultant (3600 Alma Rd Unit 3524, Richardson TX 75080-1121), [2]NASA Jet Propulsion Laboratory (4800 Oak Grove Drive, Pasadena CA 99109-8001).

Introduction: How do you provide more than 350 scientists and researchers access to data from every instrument in Odyssey when the data is curated across half a dozen institutions and in different formats and is too big to mail on a CD-ROM anymore? The Planetary Data System (PDS) faced this exact question [1]. The solution was to use a metadata-based middleware framework developed by the Object Oriented Data Technology [2] task at NASA's Jet Propulsion Laboratory. Using OODT, PDS provided—for the first time ever—data from *all* mission instruments through a *single* system *immediately* upon data delivery [3].

Middleware and Metadata: With newer instruments and future missions promising to yield greater and greater amounts and kinds of scientific data, PDS could no longer support a distribution model that relies on images, tables, plots, and calibration parameters fitting on a CD-ROM or a DVD-ROM. Online access would be an alternative, but a web interface that understands the storage formats, organization, and idiosyncrasies of each curating node would be unscalable and impossible to maintain.

The OODT framework solves that problem by providing a software component that plays the role of middleware. Moreover, this middleware system is based on metadata.

Metadata for Location Independence. Metadata is literally data about data. While one curating node of the PDS may embrace concepts such as emissivity and latitude, other nodes may have entirely different concepts like temperature and altitude. What's necessary is an automatic software system that can understand and correlate *all* these concepts. OODT's metadata system uses ISO/IEC [4] and other metadata standards to precisely describe and relate data elements like latitude and altitude along with units, conversions, synonyms, and so forth. By developing a metadata description of datasets available within the PDS, we make it possible to query the entire PDS as if it were a single giant repository independent of the location of and differences between the nodes that comprise it. We call such metadata descriptions *profiles*, and such queries *profile queries*, served by the OODT profile service. The results of a profile query are the *locations* of resources that can provide products. Recursive profile queries (that is profiles that describe other profile servers) are handled transparently by the system.

In the end, researchers don't need to know how SPICE kernels are arranged in a repository or what keywords are necessary to retrieve accelerometer datasets, or even *where* such datasets are physically located.

Middleware for Product Retrieval. Once we have the location of a desired resource, we need to determine how to retrieve products stored at that location. Every product is different, and every curating node is free to store products in whatever way is appropriate, convenient, or *de rigeur* at its location. OODT's middleware comes into play. Middleware is the glue that connects software components together. By providing a communications method between the nodes and a plug-in architecture for retrieving products from node-specific formats and storage media and converting them to Internet-standard formats based on MIME types, scientists can retrieve any kind of dataset using the single interface provided by the middleware layer.

For example, if future altimeter datasets go into a proprietary relational database, a plug-in in the OODT product service can retrieve such data, convert it into text with sentinel separated values, XML, or other standard format, and return it to the researcher. Again, scientists, engineers, and other analysts need no foreknowledge of how or where the data is stored.

Remote Maintenance. A plug-in architecture is useful insofar as it's possible to actually plug a software component in at a remote PDS node. The OODT framework includes a facility to remotely debug software, diagnose problems, upgrade components, and start and stop individual framework services. These features enable upgrading of the entire PDS network from any location as well as the installation of new plug-ins when new datasets are brought online.

Accessing the Data: PDS developers worked with the OODT group to create a web application that provides a single point of entry to the entire set of participating nodes. The application automates access to profile servers and product servers, guiding users to correct resources, and retrieving products with simple web navigation. The URL for this application is http://starbrite.jpl.nasa.gov/pds/.

For those who need to develop analysis tools, access to the OODT framework is possible with IDL (Interactive Data Language), FORTRAN, C, C++, and Java, as well as with any system capable of communicating over HTTP (Hypertext Transport Protocol).

Going Beyond the Planets: OODT's metadata-based middleware is not specific to PDS at all. It's a *general purpose* framework especially suited to the interchange and correlation of any kind of data (especially scientific, medical, and engineering). Based on XML for an interchange language, ISO/IEC 11179 and Dublin Core [5] for metadata descriptions, and a number of standard communication protocols, the system has generic application.

Cancer Research. We deployed the identical OODT framework software to a cancer biomarkers research project under the aegis of the National Institutes of Health and the National Cancer Institute [6]. The *same software* makes specimen databases stored in different databases with different schemata and different policies appear as a *single* giant tissue bank. Cancer researchers are no longer limited to specimens at their local hospital or institution.

As with the PDS, individual hospitals and institutions did not need to change their local databases or the process by which new specimens are stored and cataloged, thereby avoiding impacting everyday operations taken for granted.

Peer-to-Peer Metadata. We are currently researching new ways to provide redundancy and fault tolerance within the framework so that the unavailability of PDS's web application can be failed-over to another system with minimal (on the order of minutes) downtime. Other research includes using a peer-to-peer approach. In peer-to-peer software applications, every node is both a client and a server. There is no longer a single point of entry for a user to make a query (queries come in at *any* node), and therefore no longer a single point of failure.

Further, each node can monitor the queries and mirror (where local disk space allows) popular products, providing automatic, unattended redundancy. Should an entire PDS node go down, be disconnected, or destroyed, it's entirely possible that its most popular products will still be available through other PDS nodes.

Conclusions: The Object Oriented Data Technology framework is the foundation of the next-generation data distribution system for the Planetary Data System. It is currently making different datasets, different cataloging and categorizing systems, and different product formats appear as a single, unified whole. It provides location- and format-independence. It is remotely managed. As a generic system, it's not limited to PDS and has a wide variety of potential applications in a number of domains inside and outside of the field of planetary science. And we continue to develop and improve it.

References: [1] Slavney, S., Arvidson, E., Guinness, E.A. (2002) LPSC XXXIII 1303. [2] Crichton, D. et al., (2000) CODATA Science Search and Retrieval using XML and http://oodt.jpl.nasa.gov/ [3] LaVoie, S., (2002) PDS-D01 Quarterly Status Report. [4] ISO/IEC (1999) 11179-1, 6. [5] DCMI (1997) Dublin Core Metadata Element Set. [6] Crichton, D. et al. (2001) An Interoperable Data Architecture for Data Exchange in a Biomedical Research Network.