Dave Fulker
President of OPeNDAP, Inc.
Unidata Director Emeritus at UCAR
James Gallagher
Vice President of OPeNDAP, Inc.
EarthCube White Paper
Technical Solutions Category
17 October 2011

BRITTLENESS MITIGATION

Introduction

We offer observations on ways that distributed (and in particular, web-service) data systems break, highlighting the brittleness that makes them vulnerable. We also share thoughts how EarthCube might mitigate such brittleness, touching on matters of permanence as well as motivational factors for changes in behavior among data producers. We seek to complement Russ Rew's white paper—Technology Solutions for Scientific Data Interoperability: Unidata's Perspective—because it is highly consistent with OPeNDAP's own EarthCube-pertinent experience (see the About OPeNDAP section at the end of this paper).

Compounded Effects of Impermanence on Distributed Systems

Distributed (loosely coupled) systems, though they embody simplicity and powerful flexibility as well articulated by Rew[1], are vulnerable to the lack of persistence prevalent in this era of rapid technological change. Because distributed systems typically require *persistence in each of their constituent parts*, the brittleness of these systems—i.e., the probability of breakage—is exponentially related to the number of (distributed) components. We discuss this further in the following subsections, recognizing that it is simply one aspect of the more general and universally agreed need for a permanent scientific record that includes data.

Brittleness from Metadata Volatility

Because clients often need information not available from their servers, client developers or users often supplement server data with locally stored metadata or, alternatively, metadata from a third party. However, if the data being served change in any way (even to introduce corrections), the supplemental information may become so stale that the client-server relationship breaks.

This frustratingly simple case can diminish the utility of distributed systems, and such risks arise in almost any context where metadata are kept separately from the data they describe. Though careful application of the "wrapper" approach discussed in Rew's white paper may reduce the

[1] Rew, e.g., describes one compelling scenario as follows: *"Making the model outputs interoperable required only the installation of a specific server at each coastal center and use of small text files to adapt local model outputs to a common data model. As a result, each application client could view or run analyses on each model output as if they all conformed to a common standard, without changing the models or their outputs."*

volatility problems and enhance backward compatibility, there are human-behavior dimensions that are less easily solved.

Brittleness from Locator Volatility

Particularly in the absence of comprehensive data-location services (more on this later), client systems often employ locally acquired information about data-set locations and this too can become stale (when, e.g., a server's or a data set's URL changes).

Brittleness from both locator and metadata volatility might be characterized as consequences of "data mobility." Large historical data sets, such as the satellite observations used by physical oceanographers, are often perceived as changing little and infrequently. Our experience suggests otherwise, because even a change in a data set's URL can have significant programmatic impacts. In fact, while metadata volatility may seem more profound (and it is, from a semantic perspective), changes in URLs break client-server relations more often.

Exacerbations Related to Human Motivation

Data creators play critical roles in all EarthCube-like systems. They must be motivated to make data available, even though the payback for doing so is often low. This problem is exacerbated by demands on data creators to provide complex, comprehensive metadata. Indeed, some metadata requirements are considered quite impractical from the data creators' perspectives, such as when ontologies are developed to maximize expressivity without consideration of practicality.

A similar problem arises when data systems are designed with data discovery/location as the primary objective and access/use as a *secondary* aim (if it is an aim at all). Unfortunately, the data creators typically have no need to locate their data sets, but considerable need to *access and use* them. Hence such systems start out with requirements that are mismatched to a set of key players.

These motivational issues contribute to the brittleness problem, because they encourage the separation of data and metadata. In other words, data users, data integrators and data synthesizers often have no choice but to create supplemental metadata to support data discovery/location and/or forms of data access/use not envisaged by the original creators. This of course is precisely the advantage that Rew so compellingly articulates, but we think EarthCube strategies will be required to mitigate the associated brittleness.

Mitigation of Distributed-System Brittleness

The three subsections below sketch our ideas for how EarthCube might reduce brittleness without diminishing its commitment to a loosely-coupled, distributed architecture, as suggested in papers by Russ Rew, Bill Hibbard and others.

Use of Search Systems to Reduce Locator Volatility

Brittleness from locator volatility can be reduced by designing clients that employ search systems in place of locally-specified or user-supplied URLs. Assuming suitable generality, the search system eliminates any need for knowing exactly where or under what name a specific resource is

stored. In our view this approach is more realistic than the URI (Universal Resource Identifier) solution that is often advocated, but if truly effective URIs become practical, they would not diminish the value of an effective search system.

Unfortunately, the tendency to date has been to base search systems on *completely different protocols* from those data access and use. This requires client developers to master two forms of data description—not to mention that it risks all the same problems that arise when metadata are kept separately from the data they describe—and it substantially complicates client software. In fact, with the possible exception of the Kepler workflow client, we do not know of any clients that are highly effective in their integration of search and access functions.

We recommend that EarthCube should plan to include a search system that actually *leverages data access technologies, such as OPeNDAP and THREDDS*. We think that designing search around the metadata types required for data usage (which often are quite distinct from those that appear in search-oriented vocabularies) will offer users great power, while being relatively easy to engineer and even easier to deploy (as much or most of the needed metadata is already in place).

Use of Wrappers to Reduce Metadata Volatility

An important approach to reducing the issue of metadata-volatility is through use of wrappers as mentioned in the Rew paper[2]. A wrapper creates a new view of the underlying data set (i.e., a virtual data set) that is accessible via our data access protocol (DAP). We recommend that EarthCube should articulate and encourage a "best practice" something like this:

> *When correcting, augmenting or redefining metadata, the original information should remain untouched and accessible. The changes should be realized solely via wrappers, such as NCML, that create a new* ***virtual*** *data set, which internally and transparently points to the original.*

In many cases this practice would ensure backward compatibility, and in all cases (perhaps with a little legwork) it should ensure replicability of prior results and calculations. Virtual data sets are especially valuable when the context of use is far removed (geographically or disciplinarily) from that of the data creator. Such distances often lead to vocabulary disparities that, in turn increase the risk of metadata volatility, but even these can be reduced through effective use of wrappers.

Another benefit of virtual data sets is that they can be employed effectively for in-house data systems—where the motivations for creating metadata are inherently highest—without diminishing their potential (which may grow incrementally, especially with community involvement) for utility among geographically or disciplinarily distant users.

Use of Multiple-Copy Strategies to Enhance Permanence

Our concluding idea about reducing brittleness is for EarthCube to become directly engaged in the broader problem of permanence (a.k.a. persistence). OPeNDAP does not have direct experience with this matter, but data that have been made Web-accessible using our DAP

---

[2] Rew lists, among Useful Current Technologies, "virtual dataset wrappers providing subsets, aggregations, and additional metadata for data collections."

(optionally employing wrappers, as above) are ideally positioned for automatic replication in institutional repositories or other facilities.

Having in existence multiple copies of a digital artifact hugely increases the likelihood of its being permanently accessible, especially if some of the copies are retained by institutions, such as libraries, with long-standing reputations in archival matters. The best thinking we have seen on this matter, as it pertains to scientific data, may be found among the DataOne materials[3]

Closely related to the idea of replication are the twin ideas of using unique identifiers to track data granules (also a component in DataOne's design) and the use of checksums to verify data integrity (including subsets of data granules). Both of these tools can be applied without undue cost or loss of generality to the kinds of distributed systems under consideration. Each can greatly enhance the robustness of the resulting systems as well as increase the confidence of users in the authenticity and integrity of the data they access.

About OPeNDAP

The Open-source Project for a Network Data Access Protocol (doing business as OPeNDAP) is a not-for-profit corporation that develops and supports systems and software for Internet-based "publication" of complex scientific data on large scales. The root OPeNDAP concept, conceived in the early 1990's (as the Distributed Oceanographic Data System, or DODS), was to build a data-access protocol over http, one that conveys structural semantics as well as data and metadata and that also supports powerful (server-side) subsetting operations to reduce bandwidth requirements and enhance performance.

OPeNDAP's core data-access protocol (DAP version 2) is widely used, with realizations deployed by many organizations (including large and small data creators, as well as consumers) in a remarkable array of servers and clients. The latter include IDL, IDV, GrADS, MatLab, McIDAS-V and RAMADDA, as well as any tool that incorporates the netCDF Library (because that library can read from both *netCDF files* and *DAP servers*).

This success is due in major part to having built the DAP on a well-defined abstract data model that is both language-independent and closely related to netCDF (see the white paper by Bill Hibbard, et al). Furthermore, OPeNDAP is presently collaborating with Unidata (supported by NOAA) to achieve closer alignment between the DAP-compatible servers that each organization now offers—Hyrax from OPeNDAP and TDS (THREDDS Data Server) from Unidata. One longer-term objective of this collaboration is to migrate the systems toward a common framework.

[3] http://mule1.dataone.org/ArchitectureDocs-current/design/PreservationStrategy.html