An EarthCube Technical Solution[†]: the VisAD Data Model

Bill Hibbard[1], Dave Fulker[2], Tom Whittaker[1] and Hank Revercomb[3]
[1]Space Science and Engineering Center
[2]President, OPeNDAP; Unidata Director Emeritus, UCAR
[3]Director, Space Science and Engineering Center

Scientific studies, especially of Earth systems, rely on and generate types of data that vary *immensely* in their structures, formats and semantics, due in part to the wide range of sources (from in situ and remote observations to synthesized values) that pertain. A major consequence is that scientific progress presently is impeded by the practice of developing data-specific software and tools that—as a rule—create *additional* data to which few other tools are directly applicable. Informal surveys have shown that scientists can routinely spend more than half their time just engaged in accessing data.

We assert that EarthCube—to have the intended transformative effect—must address this problem of proliferating data types. We further suggest that the solution requires adopting and propagating, via software tools and infrastructure, an *abstract, unified data model* around which software designs can coalesce community-wide. If these abstractions exhibit the right balance of utility and expressivity, with the latter sufficient to represent most observed and synthesized data now in use, we think that EarthCube can greatly reduce barriers to data access, use and reuse. This reduction will in turn accelerate growth in scientific understanding, especially across disciplines and other differentiating factors within the EarthCube community.

Though widespread adoption of the *relational model* and associated operations has yielded huge advances in data-base management system (DBMS) infrastructure, this model is inadequate in many areas of scientific study. As reflected in rapidly growing use of non-relational models such as the Network Common Data Form (netCDF) and HDF, this inadequacy is most pronounced when studies involve quantities that vary across time and two or three space coordinates. In the remainder of this paper we argue for EarthCube adoption of a grammar-based data model that is informed by experience with the Visualization for Algorithm Development (VisAD) data model, if not based on it. As described below, this model embodies abstract data types that have been effective in the analysis and visualization of data integrated from remarkably diverse data sources.

Process and Community

The Space Science and Engineering Center (SSEC) has long experience with scientific data in the development of its McIDAS and Vis5D systems, both with large user communities. Based on lessons learned from this experience, in 1990 SSEC began development of the VisAD system as an attempt to bring all these diverse data into a single, unified data system. VisAD version 1 was written in C and was used to analyze data from several meteorology and astronomy projects. The system was difficult to use

---

[†] This paper is being submitted jointly by SSEC and OPeNDAP. It is one of three Technical Solution papers from SSEC. These are among five EarthCube papers submitted by a newly formed, collaborative working group at University of Wisconsin-Madison that spans many colleges, centers, departments, and partners.

but did exhibit considerable flexibility for analyzing and visualizing diverse data. In January 1996, when the Java language became available, SSEC began a total redesign of VisAD in Java. This was done in collaboration with the Unidata Program Center, which shared SSEC's enthusiasm for Java and for achieving generality via data abstractions. A series of joint design reviews combined the considerable experience of SSEC and Unidata (each serving its user community), to formulate a robust design for VisAD version 2. Programmers from SSEC and Unidata collaborated on the implementation, which was first demonstrated at the 1998 JavaOne Conference. A short while later the Australian Bureau of Meteorology joined the collaboration and more recently the Indian Space Research Organisation has also contributed important improvements. As an open source system, VisAD received ideas and code contributions from several other groups.

VisAD is a distributed component library that can be combined with other libraries to build application systems. Significant examples (in multiple disciplines) include SSEC's McIDAS-V, Unidata's IDV, UW LOCI's VisBio for microscopy data, GEOpod developed by students at Millersville Univeristy, and Geon IDV developed by Unidata, UNAVCO and GEON for geophysical data. McIDAS-V and the IDV share much code and are closely related. Furthermore, the RAMADDA system for publishing and sharing scientific analyses and visualizations is tightly integrated with the IDV and McIDAS-V. For more information about these systems see:

http://www.ssec.wisc.edu/mcidas/software/v/
http://www.unidata.ucar.edu/software/idv/
http://www.loci.wisc.edu/software/visbio
http://www.unidata.ucar.edu/committees/usercom/2011Apr/geopod.html
http://www.unidata.ucar.edu/projects/index.html#geon
http://www.unidata.ucar.edu/software/ramadda/

The Data Model

In order to combine diverse data, VisAD is designed around a unified and extensible data model, which is a formal mechanism (i.e., a grammar) for expressing data in a manner that is rich with meaning. The atoms of this grammar are numeric and text variables, which have names and optional units. For example, the built-in variable Latitude has unit Degrees and the variable Time has unit Seconds. These, along with user-defined variables, can be grouped in tuples, such as:

(Latitude, Longitude, Altitude)	for earth locations
(X, Y, Z)	for earth locations
(U, V, W)	for wind vectors

Any tuple of numeric variables may describe a coordinate system and include a transform to reference coordinates, for example from (X, Y, Z) to (Latitude, Longitude, Altitude). Such coordinate transforms may be user-defined and may even be time dependent, as is common in situations where some variable, such as pressure or density, serves as a proxy for elevation. More complex tuples can be constructed such as:

(Temperature, Pressure, Humidity, (U, V, W))

Data types can be combined in functional relations, such as the following, which represents a spectrum:

(WaveLength → Radiance)

Such functions are usually implemented by finite samplings. A more complex example of functional dependency is:

(Time → ((X, Y, Z) → (Temperature, Pressure, Humidity, (U, V, W))))

This represents the time varying state of the atmosphere, such as output by a weather or climate model. It typically includes metadata for a complex sampling topology and geometry in its 3 spatial dimensions, which may vary between time samples.

In addition to units, coordinate transforms, and sampling topology and geometry, metadata may include missing-data indicators (any data object or sub-object may be missing) and error estimates for numeric values. The system implements complex and transparent processes for converting units, transforming coordinates, resampling, and propagating missing data and error estimates as necessary when combining data in computations and visualizations. Currently supported sampling topologies include both rectangular and triangular grids. Other sampling topologies could be added, which optionally include user-defined interpolation algorithms. Samplings may be constrained to manifolds with lower dimension than the domain, for example to represent temperature on the 2-D topographical surface of the Earth embedded in 3-D space. The data model supports representation of functions by means other than samplings, such as by linear combinations of basis functions, although these are not currently implemented. Additional metadata can be included as fields in data structures, such as:

(CreatorName, CreationDate, Latitude, Longitude, (Time → Temperature))

Simply adopting a data model—as an "EarthCube Standard" for example—will not have the desired impact, due to the twin challenges of developer motivation and compliance checking. However, the authors' experience with netCDF, OPeNDAP and VisAD (among others) indicates clearly that properly designed *software libraries* can motivate developers and guarantee compliance to a degree that yields significant enhancement to interoperability. Hence we propose that something akin to the VisAD data model *and* something akin to its realization as the *VisAD software library* become integral parts of the EarthCube infrastructure. This of course implies a non-trivial need for software engineering as well as training and support services, but the practicality and benefits of this approach have been well demonstrated at SSEC, Unidata and elsewhere.

We note that Russ Rew's EarthCube paper, Technology Solutions for Scientific Data Interoperability: Unidata's Perspective, presents strong arguments for an approach similar to ours. Rew's solution brings diverse data into a common file application programming interface (API), for netCDF, whereas our solution brings diverse data into a common data object API, for VisAD. Since netCDF is one of the file APIs that has been

adapted to its data model, VisAD can take advantage of any data format accessible via netCDF. Also note that the IDV is mentioned as part of the solution in Rew's paper as well as in Beth Plale's EarthCube paper, Atmospheric Sciences and Informatics EarthCube Driver Whitepaper: Technical Infrastructure. We believe that EarthCube would benefit from netCDF and the VisAD-based IDV/McIDAS-V.

Rew also mentions Unidata's *Common Data Model*, which we believe aligns closely or exactly with major parts of the VisAD model. Two differences we perceive are that VisAD uses more mathematics-oriented (i.e., less discipline-specific) terminology, and it is designed for greater extensibility in function representations and sampling topologies. VisAD extensibility has already been employed for non-rectangular sampling topologies and it can be used to embrace new kinds of sampling manifolds (such as may arise with remote-sensing innovation) and/or additional representations (including those that are function based, including Fourier series and wavelets). We think a key challenge for EarthCube will be to address the tradeoffs between simplicity and generality in the data models adopted to gain interoperability.

At the top level, VisAD is a library for building networks of data, computation, display and user interface components that span multiple machines across the network. The system supports a Python scripting capability to ease customization by end users. The system also includes code for importing hundreds of different data sources, as file formats and as server protocols, into its data model. This enables data from diverse sources to be combined in computations and visualizations. We are experimenting with an interface that allows users to interactively examine files and specify their structures to the system. This approach can work for "well-behaved" files but for the general case there is no alternative to custom programming for access to novel file formats. For more information about VisAD see: *http://www.ssec.wisc.edu/~billh/cacm2005.html*

Future Plans and Lessons Learned

The ORIGAMI project at SSEC demonstrated analysis of large data sets across processor clusters integrated with the VisAD data model and the McIDAS-V user interface. An important future direction for VisAD/IDV/McIDAS-V is their integration with a production capability for high-throughput computing. However, because they support large user communities, IDV and McIDAS-V programmers are also kept busy serving the more immediate needs of users. This connection with users is an essential source of understanding and—assuming adequate resources—a great example for EarthCube.

Our experience with VisAD/IDV/McIDAS-V can contribute other lessons learned to the EarthCube effort. The VisAD data model has been sufficiently expressive to represent literally every type of data our users have brought to it (from geoscience research to microbiology to financial analysis), yielding effective fulfillment of their visualization needs, a good argument in favor of the grammar approach to data models. Figure 1 in our CACM article (URL given above) shows our attempt at a graphical user interface (GUI) for designing data visualizations without the need for programming. This was not flexible enough to meet users' needs, similar to the experience with other researchers' proposals for designing visualizations without programming. While the VisAD API provides powerful functions, it does so at the price of a significant learning

curve. We are continuing our efforts to simplify the API by use of our integrated Python scripting language. But note that over the past 30 years none of the attempts to enable users to create novel visualizations with zero or minimal programming effort have been successful. Therefore a realistic goal for a project like EarthCube must be to increase the productivity of programmers and to enable them to do things they couldn't previously do rather than eliminate the need for programming.

The grammar approach and the expressivity of the VisAD data model have successfully enabled IDV and McIDAS-V developers to bring highly varied atmospheric data into a common framework for analysis and visualization. There are fewer success stories about integrating data across the atmospheric, oceanic, hydrologic, geophysical and biological sciences, although VisBio, GEON IDV, IDV access to ocean data, and a few experiments with astronomy data have fully demonstrated the soundness of the technical approach. Hence the problem may be that few scientists perceive the value of merging data between disciplines, but a successful EarthCube may change this.

For many years we have strived to extend our approach to a broader scientific community, and we are excited to work with EarthCube on that goal. We recognize that new people with new ideas often are needed to bring a new level of success. One idea already suggested by other EarthCube collaborators is a *language-independent* formulation of a grammar-based data model (currently the VisAD data model is expressed in Java and Python). Another suggestion is to reduce the expressiveness of the data model in order to simplify its use, but we know of use cases where *greater* expressiveness is required, encompassing, e.g., meshes comprising more than triangles and rectangles, or fields represented as linear combinations of basis functions. Perhaps the EarthCube project can find alternative ways to realize the benefits of a highly expressive, grammar-based data model but with a less steep programmer learning curve. Many such experiments could exploit the existing software to reduce their required time and effort. In any case, we hope that EarthCube can benefit from lessons learned during the VisAD/IDV/McIDAS-V developments.

About SSEC

SSEC is a research and development center with primary focus on geophysical research and technology to enhance understanding of the atmosphere of Earth, the other planets in our Solar System, and the cosmos. SSEC researchers sometimes explore the universe from space and terrestrial-based telescopes, and probe other planets in our solar system, but more often they examine the Earth to gain information and insight into weather, climate, and other aspects of Earth's global environment. They develop new observing tools for spacecraft, aircraft, and ground-based platforms, and model atmospheric phenomena. They receive, manage and distribute huge amounts of geophysical data and develop software to visualize and manipulate these data for use by researchers and operational meteorologists all over the world.

Three related but conceptually distinct aspects of the above activities, refined during almost four decades serving the Earth Sciences community, form the basis for SSEC’s Technology Solution papers:

1) Data abstraction, exemplified by the VisAD data model, as critical underpinning for interoperable processing, visualization and data exchange (this paper);
2) Broad community support through the Open Geospatial Consortium protocols; and
3) Possible semantic implications for new EarthCube approaches.

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 early1990'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 subsetting operations, thus reducing bandwidth requirements and enhancing performance.

OPeNDAP's core data-access protocol (DAP version 2) is widely used, with realizations deployed by many organizations (including large and small data providers, as well as consumers) in a remarkable array of servers and clients, the latter including MatLab, IDL, GrADS and RAMADDA, as well as software mentioned in this paper: IDV, McIDAS-V and the netCDF Library.

This success is due in major part to having built the DAP on a well-defined abstract data model (closely aligned to netCDF). Furthermore, OPeNDAP plans now include adding richness to that model, deeply informed by the VisAD approach/experience discussed here, but potentially influenced by future EarthCube decisions on data-model adoption.