Building the New Environmental Software World in Java using VisAD

Bill Hibbard[1], Don Murray[2], James Kelly[3], Ugo Taddei[4], Doug Lindholm[5] and Tom Whittaker[1]

[1] University of Wisconsin [2] Unidata Program Center [3] Australian Bureau of Meteorology
[4] University of Jena [5] National Center for Atmospheric Research

Physical computing and communications equipment is constantly improving. Computing speeds and memory capacities increase exponentially according to Moore's Law2. Computers are virtually all connected together in a global Internet and communications speeds are increasing exponentially.

Periodically software makes abrupt changes in order to translate these quantitative physical improvements into qualitative changes for users. For example, increasing speed of supercomputers led not only to increased weather model resolution, but also to the introduction of ensemble forecasting. And increasing workstation performance led to graphical user interfaces, color, 3-D and interactivity in visualization.

The ubiquity and performance of computer networking have set the stage for another major change in software, and Java is the basis for that change. Java is a programming language and set of APIs specifically designed for the network. In order to exchange information over the network, the global community is developing de jure and de facto standards such as ASCII and Unicode for text, HTML and XML for hypertext, GIF, JPEG, TIFF and MPEG for images, WAV and AVI for sound, and VRML for 3-D shapes. However, programs are the most fundamental type of information in computers, so a format for exchanging programs answers the most fundamental need. Java is such a format. It must solve a variety of very difficult technical problems, such as building program compilation into the program loading process, and defining standard APIs for every basic service.

In addition to exchanging programs between computers, the network also requires a way for programs to run on multiple computers rather than being restricted to one computer. Distributed object technologies such as CORBA and DCOM enable this. But the Java RMI (Remote Method Invocation) technology is so much easier to use, because it enables the syntax and semantics of remote objects to be almost identical with local objects. In particular, the portability of Java code means that the arguments and return values of methods of remote objects can be sent across the network even if they are full-fledged objects. In CORBA and DCOM only pure data arguments and return values can be sent across the network to methods of remote objects.

By making programs portable between operating systems, Java threatens Microsoft's operating system monopoly and hence faces fierce opposition from the most powerful corporation in the world. Despite these technical and political challenges, we believe that Java must eventually remake the entire computing world just because a format for exchanging programs on the network is so fundamental. Java is now taught as the primary programming language at most world universities, and is the primary programming language for phones, credit cards and other small devices that will define the next major physical increase for the Internet.

Java for Meteorological Workstations

Recognizing the importance of the Java revolution, there are a number of projects to rewrite meteorological workstation software in Java. VisAD is such a project, or perhaps it would be more accurate to say that it is the infrastructure shared by a number of such projects.

Just as the new realities of the networked world drove new design features of Java, the new realities of the networked meteorological world drive new design features of meteorological workstation software. The primary new reality is the increased sharing of many different types of information among scientists from different institutions and even different disciplines.

In order to support increased information sharing, VisAD is centered around a universal numerical data model. This data model can be used to express any numerical or textual data, based on the flexibility of its data schemas and the integration of metadata. For example, a satellite image of Earth may be described by the schema:

((line, element) -> radiance)

which defines a functional dependence of radiance on pixel line and element coordinates. This function may also include metadata defining its earth navigation as an invertible coordinate transform:

(line, element) <—> (latitude, longitude)

An image function is typically represented by a finite sampling at discrete pixels (in VisAD the sampling metadata of a function may be regular or irregular). But the VisAD interface for functions can have other representations such as finite sets of spectral components, or procedures for computing function values.

Any real values in VisAD data objects may include units. For example latitude and longitude values may have units of degrees or radians. Also function range values such as radiance may be marked as missing or include error estimates.

A time sequence of images may have the schema:

(time -> ((line, element) -> radiance))

This function will define some finite sampling of time values, and may define units for time such as seconds since 1 January 1970.

A set of map boundaries may be described using the schema:

set(latitude, longitude)

VisAD defines many different classes for sets in real vector spaces. For example, map boundaries would be defined as a UnionSet of Gridded2Dsets with manifold dimension = 1 (i.e., a union of polylines in two dimensions).

The output of a weather model may be described using the schema:

(time -> ((row, column, level) ->
(temperature, pressure, moisture, wind_u, wind_v, wind_w)))

with earth locations defined by an associated invertible coordinate transform:

(row, column, level) <—> (latitude, longitude, altitude)

The goal of integrating metadata is actually to create systems that enable end users to ignore metadata (but also to manipulate metadata if they wish to). For example, a user might read satellite images from several different sources and several different file formats, each sampled at different map projections, and at different time steps. The file format adapters will read each file into a VisAD data object that includes the image data and metadata objects containing the image's spatial and temporal sampling information. Arithmetical operations will co-locate the data. For example, if one image is subtracted from another image, the radiances from the second image will be resampled to the spatial and temporal locations of the first image before they are subtracted. If the two images use different radiance units, these will be converted before values are subtracted, and before they are displayed together. As the Internet enables greater data sharing among scientists, it increases the problems associated with metadata and file format differences among scientists. Metadata integration in a common data model is an important tool for addressing these problems, both for those users who want to ignore metadata and those who want to control metadata.

The VisAD data model includes interfaces to a large and growing set of file formats. These include netCDF, FITS, HDF-5 (needs native C library from NCSA), McIDAS, McIDAS ADDE, DODS, GIF, JPEG, TIFF, Quicktime, Vis5D, HDF-EOS (needs native C libraries from NCSA and NASA), ASCII, Biorad, F2000 (neutrino events), Shape (distributed by Unidata with their MetApps system), VisAD (serialized VisAD data objects), ArcGrid ASCII (distributed as part of University of Jena's DEMViewer). In addition, others in the VisAD community have announced, through the VisAD mailing list, their work on the development of interfaces for importing data stored in other file formats such as GRIB (using a native C library) and MIF. We expect that the list of file formats supported by VisAD will continue to grow as the community grows.

In addition to its universal numerical data model, VisAD defines abstract interfaces for displays and computations. Its display interface currently has implementations in Java3D and Java2D, and is designed to enable implementations in other graphics APIs.

Any data object can be linked to any display object. This is a critical capability to allow sharing of data among different applications built using the VisAD infrastructure. For example, a radar application built using VisAD can export its radar data objects to a model application, where they can be overlaid in space and time with displays of model output data objects.

Furthermore, based on Java RMI, VisAD data objects can be linked to VisAD displays on multiple workstations enabling groups of scientists to share visualizations of and interactions with the same data. In fact, VisAD defines a cookbook procedure by which any application can easily be converted into a collaborative application for use by groups of scientists.

VisAD's classes are specifically designed to enable applications to modify their behavior by defining extension classes. This includes the ability to define new coordinate systems and new interpolation algorithms embedded in data objects, as well as whole new ways to represent functions (e.g., as procedures) and other data objects (e.g., representing real numbers using intervals with rational end points). There are numerous ways to change the behavior of displays by extending classes, including the way that data objects are transformed into visual depictions, the graphics API used for rendering, the way axis scales are rendered, and the way user gestures are interpreted as interactions that modify data values.

With the earlier Vis5D system it was impossible to prevent the development of multiple incompatible versions, as different institutions modified it to meet their particular needs. This has not occurred with VisAD and we believe it will not, based on its support for customization via its object-oriented design. Thus different institutions can customize it to meet their needs, and will still be able to exchange data and even merge their applications1,3,4.

Meteorological Applications of VisAD

The University of Wisconsin is developing a variety of VisAD applications for viewing data from models (JMET) and new instruments (MODIS, AERI, GIFTS and IceCube). The AERI application is notable because it fuses data from GOES, AERI (an up-looking vertical sounder that senses temperature and moisture) and wind profilers, aligning them in time and space. The GIFTS application is notable because of the large data volumes anticipated from the GIFTS instrument: global images with thousands of spectral channels per pixel. The JMET application provides a generic ability to visualize model data from netCDF files. The JMET and AERI applications are distributed with the VisAD system. VisAD is freely available for download, including source code, documentation and example applications, from:

http://www.ssec.wisc.edu/~billh/visad.html

The Unidata Program Center is using VisAD in their Metapps project, developing the next generation of meteorological applications for university education and research. Metapps is freely available for download from:

http://www.unidata.ucar.edu/projects/metapps/SoftwareDownload.html

Metapps currently includes an Image Viewer, a Gridded Data Viewer and an Interactive Sounding Viewer. The Image Data Viewer is notable for its flexible ability to download animation sequences from any McIDAS ADDE server for any common satellite, and to display them with map overlays in a variety of map projections and on a 3-D globe. The Gridded Data Viewer is notable for its replication of Vis5D's 3-D rendering techniques. The Interactive Sounding Viewer is notable for enabling users to modify sounding data by graphical gestures. Work is in progress on an integrated data viewer to combine satellite and radar image data, model grids, and point data from observing stations.

The Australian Bureau of Meteorology is using VisAD to develop an Automated Marine Forecast System, and a Tropical Cyclone Module.

The Automated Marine Forecast System is designed to streamline marine forecast operations by allowing meteorologists to interactively manipulate forecast wind and sea fields (from Numerical Weather Prediction schemes), and then to automatically produce the graphical and textual products required. In this way, the forecaster is able to concentrate on the science of getting the forecast right, rather than spend time crafting the words. In addition, better verification procedures are able to operate on the digital forecast fields stored in the Forecast Database. The Tropical Cyclone Module is designed to assist operational meteorologists in their task of tracking Tropical Cyclone position and movement, as well as assist in the forecast role. This work involves the integration of a wide range of existing data types (including observational, radar and satellite data), as well as the necessary cyclone track data. A key component is to allow forecasters to interactively manipulate cyclone forecast positions on screen, and to be able to simultaneously view/modify the result in a tabular type display if desired.

The National Center for Atmospheric Research (NCAR) is developing the Visual Meteorology Tool (VMET) within their Research Applications Program. This system provides a framework and a growing set of extendable components for creating a variety of displays integrating a variety of data, including observations, images and model output grids. Current development includes a real-time display integrating meteorological data to support US Army test and evaluation. Examples of its displays are available from:

http://www.4dwx.org/vmet/

The NCAR Scientific Computing Division (SCD) is using VisAD as part of the Visualization Enviroment of the Visual Geophysical Exploration Environment (VGEE):

http://www.dlese.org/vgee/vistool.htm

SCD is also investigating the use of VisAD in developing applications to replace the functionality now provided by Vis5D.

The University of Jena has developed DEMViewer for visualizing digital elevation maps and hydrology catchments. It is available from:

http://www.geogr.uni-jena.de/~p6taug/demviewer/demv.html

The University of Jena has developed other applications for visualizing time series, as well as the VisAD Tutorial.

In addition to these efforts, there are about 360 developers subscribed to the VisAD mailing list. They are developing a wide variety of environmental, biological, astronomical and financial applications of VisAD.

Conclusion

The effort to rewrite VisAD in Java began in January 1996, with the initial release of Java out of beta testing. However, given the scale of the effort to rebuild the computing world in Java, the VisAD effort is just getting started. The generality of VisAD's data model and distributed component architecture will keep it technologically relevant for decades. And the increase in computing speed, memory capacity, network bandwidth, and network ubiquity over the next decade or two will increase the need for VisAD's new technologies.

VisAD's status as an open source system and a collaboration among global institutions will enable the system to play an increasing infrastructure role in meteorological computing far into the future. We demonstrated the effectiveness of this approach with Vis5D, whose primary development has migrated to a number of institutions other than the University of Wisconsin.

Acknowledgements

VisAD is a collaboration among a growing number of programmers at different institutions. In addition to those listed as authors of this paper they include Curtis Rueden, Tom Rink, Dave Glowacki and Tommy Jasmin at the University of Wisconsin, Steve Emmerson and Stuart Weir at the Unidata Program Center, Andrew Donaldson, Ken Stephenson, KoKo Aye and Jim Koutsovasilis at the Australian Bureau of Meteorology, David Leberknight, Troy Sandblom and Julien Chastang at NCAR, and Peter Cao at NCSA.

References

1. **Hibbard, W.** (1998): VisAD. Connecting People to Computations and People to People. - ACM Siggraph *Computer Graphics* 32, **No. 3**: 10-12.

2. **http://www.intel.com/intel/museum/25anniv/hof/moore.htm**

3. **Murray, D., T. Whittaker, J. Kelly**, Accessing remote data servers through Java, Preprints, 17th Int. Conf. on IIPS for Meteorology, Oceanography and Hydrology, pp. 178-181, 2001

4. **Taddei, U.; David, O. & Michl, C.** (2000): Application of VisAD in Hydrological Modeling and Simulation. - ACM Siggraph *Computer Graphics* 34, **No. 1**: 48-51.