5.13 AN EXAMPLE OF UNIDATA'S FUTURE IN NEW SOFTWARE: THE VisAD COMPONENT ARCHITECTURE FOR COLLABORATIVE DATA ANALYSIS AND VISUALIZATION

William Hibbard*

Space Science and Engineering Center
University of Wisconsin - Madison

There is currently great interest in reusable software components as a way to decrease software development costs. For this strategy to work it is not enough to simply wrap software in some component architecture such as JavaBeans or ActiveX. In order to achieve reuse and to decrease costs, components must be general and must provide significant services.

VisAD is a Java class library for interactive and collaborative visualization and analysis of numerical data. This library defines four basic types of components:

1. Data.
2. Display.
3. Computation.
4. User interface.

Visualization is the core service provided by VisAD. The generality of this service is defined by the variety of implementations of the Data and Display component interfaces. The data model includes essentially any numerical data and integrates metadata for data organization, units, coordinate systems, sampling geometries and topologies (regular and irregular), missing data and error estimates. This means that any numerical data can be visualized and that unit conversions, coordinate transforms and resamplings are done implicitly as needed when data from diverse sources are overlaid in one display. The data model has been implemented for the netCDF, FITS, HDF-EOS, McIDAS, Vis5D, GIF and JPEG file formats, with an HDF-5 implementation under development.

The display model includes Java3D and Java2D implementations and has been used in web browsers, ordinary applications on Unix and Windows, and in the immersive virtual reality ImmersaDesk. The display model achieves generality via its scalar mapping technique for parameterizing the different ways that data can be displayed (e.g., map time to the X axis and temperature to the Y axis to define a time series plot), via a hierarchy of control classes for user interaction with displays (e.g., pan, zoom, rotate, animate, change iso-levels and color maps), and via the extensibility of its classes. Note that different interactive controls can be linked together to define behavior such as fonts that maintain constant screen size during zoom, and progressive disclosure of detail during zoom.

The generality of the core visualization service allows VisAD components to be reused in a wide variety of applications. This is especially valuable when there is a requirement to merge the functionality of multiple applications: by supporting a wide range of applications VisAD makes it possible to merge them based on their shared data and display models.

In order to reduce development costs, VisAD components integrate a number of other services, in addition to the core visualization service. They include:

1. Distributed computing: developers can easily define networks of data, display, computation and user interface components distributed arbitrarily around the Internet.
2. Thread-safeness: synchronization of operations in highly parallel networks is handled internally by components so developers can generally ignore this issue.
3. Collaboration: state consistency between linked components is handled internally, so developers can make collaborative versions of their applications with minimal extra effort.
4. Visual manipulation of data: data components can linked with display components in a way that allows users to modify data values by redrawing data depictions.
5. Integration with analysis: computation components can be linked with data, display and user interface components to support interactive data analysis and computational steering. Furthermore, data components support a variety of mathematical operations including "array operations" on complex data objects such as images. Implicit unit conversions, coordinate transforms, resampling and propagation of missing data and error estimates are built into these operations.

The VisAD system is freely available from http://www.ssec.wisc.edu/~billh/visad.html, including source code, documentation and example applications.

* Corresponding author address: William Hibbard, SSEC, 1225 W. Dayton St., Madison, WI 53706; email address: hibbard@facstaff.wisc.edu