6.14 JAVA™ ON THE WORLD WIDE WEB AND BEYOND

Joel Plutchak*, Steven Hall, David Wojtowicz, Robert B. Wilhelmson, and Mohan Ramamurthy
University of Illinois at Urbana-Champaign, Urbana, Illinois

I. Introduction

In 1997, the Atmospheric Sciences Department at the University of Illinois released Weather World 2010 (WW2010), which integrates access to meteorological data with educational material on the World Wide Web (WWW) using a customizable, intuitive user interface (Hall, 1998). One of the components of WW2010 is a set of extended-capability data viewers written in the Java programming language (Sun Microsystems, 1996).

Our previous work with Java (Plutchak, 1997) has been in the form of applets considered to be experimental prototypes, and applications or components meant to be building blocks for future development. The release of WW2010 incorporated the first use of Java technology in a production environment on our web servers.

Many considerations went into development of the Java applets. These include the capability to integrate Java applets smoothly into our web materials, the maturity and stability of the Java language and environment, and the likelihood of easily expanding the role of Java in the future.

II. Integrating Java into Web Documents

The effectiveness of Java applets can be greatly enhanced when combined with the strengths of conventional WWW documents. WW2010 uses Java applets as extensions to the presentation of data and educational material, rather than as the primary or only means of access. This allows those who do not have access to Java to benefit from the web documents, while giving those with Java capability the advantages of greater interactivity with the data.

One obvious advantage to fitting Java applets into conventional web site design is the ability to take advantage of existing educational and informational documents and images. With little additional effort, existing text and graphics can either surround the applet within a web page or be accessed and displayed by the applet itself.

In keeping with the dual-use theme, the server that supplies static images or graphical presentation of data (Wojtowicz, 1998) can also supply images and data to the Java applets. This method is used by WW2010. Each web page is composed on the fly based on user-selected parameters, and contains the HTML required to display the requested data. Generally, this is a static image. When Java has been enabled by the user, the HTML contains a stub Java applet plus a few JavaScript statements that pass parameter file descriptors to the stub applet. These parameter files contain the information needed for the applet to build the appropriate display window and menu items for accessing the data with the more interactive presentation of the Java applet.

III. WW2010 Java Applets

The current applets are deployed as an extension of the WW2010 philosophy of providing integrated, scaleable access to data and educational materials. In that capacity, they serve as the high end method to access data-- a step above simply viewing a static image or graphic.

One applet, the Interactive Weather Report, is a more modular and generic reworking of our first prototype effort. It displays several different sets of graphical data in superimposed layers, reporting data and location information for any point on the display. This basic functionality was implemented to allow the type of data to be parameterized. This allows the same applet to be launched from a variety of web pages with different combinations of display data, geolocation information, data calibration codicil(s), and other ancillary information.

A legend bar contains icons for each displayed layer. In keeping with WW2010 philosophy, these icons are used as buttons that link to educational material for the corresponding type of data.

Another capability considered important by users of our data is animation-- viewing a time sequence of satellite images or other graphical data. A previous animation prototype was totally

* *Corresponding author address:* Joel Plutchak, Dept. of Atmospheric Sciences, Univ. of Illinois at Urbana-Champaign, 105 S. Gregory St., Urbana IL 61801; email <plutchak@uiuc.edu>.

rewritten to provide a cleaner user interface and more maintainable internal organization.

The animation applet allows either a fixed or customizable number of images to be looped through with complete user control of speed, dwell time, direction, etc. As in the Interactive Weather Report applet, geographic location for any point on the display is available, and access to educational material for the data type being displayed is provided.

Further applets under development will allow the display of other types of data interactively. For example, a simple scrolling text applet can display continually updated local forecast or current weather condition text for any desired location. Another may allow charting of upper air soundings in various graphical forms, for example, Stuve or Skew-T diagrams. Regardless of the specific function of these applets, they will be designed to integrate smoothly into the WW2010 WWW site.

IV. Application Development with Java

As an object-oriented programming language, Java allows the developer to design a general-purpose object that contains most of the common functions used for a given high-level component. If a particular program using the component needs to add a small amount of custom functionality to that more general object, it can be extended by adding to or overriding just those parts of the component that need to be different.

Reusability has been a key goal while designing the software. By identifying common concepts, objects, and algorithms early in the development cycle, shared components can be written and used for a variety of end-user applications.

This philosophy of extensibility and reusability has been put into practice by a group of people working loosely under the Horizon Project umbrella at the University of Illinois. Each workgroup, from separate academic departments and groups, had initially made independent use of Java to view scientific data. After comparing notes, a basic set of commonly needed functionality was identified, leading to the design and implementation of an extensible framework for assembling science data browsers (Plante, 1997). The resulting freely available Horizon Image Data Browser (HIDB) package (NCSA, 1997) is currently being used in applications written by each group.

The HIDB defines generic components, based loosely on the Model-View-Controller paradigm, which represent scientific data sets as well as displayable views into that data. A generalized coordinate transform package, adapted from the FITS astronomical image processing package, is available in order to translate between image coordinates and a variety of real-world coordinates. In addition to the abstract framework definition, the HIDB contains several concrete instances of the components, which can be used as is for a variety of data types, for example, FITS images, HDF data sets, GIF and JPEG images, etc.

While the object-oriented nature of Java, along with the ease of distributing and maintaining web-distributed Java applets, are definite advantages to using Java, there are some disadvantages. Foremost among them is the slow maturation of the language and environment. While the past year has seen progress in the underpinnings of the Java applications programming interface (API), there are still many useful higher-level packages that have been slow to appear. Some elements still missing in final form include 2D and 3D graphical libraries, a richer set of image manipulation packages, and native support for the Virtual Reality Modeling Language (VRML). These are quite important for developing full-featured and robust scientific applications without necessitating a large amount of duplicated development effort.

V. Collaborative Applications

The ability to collaboratively view and annotate scientific data with remote colleagues remains an important goal. A brief survey of existing programming frameworks for developing collaborative applications reveals several efforts underway, all of them using the Java programming language and environment. These include Tango from Syracuse University, Java Collaborative Environment from Old Dominion University, Shaking Hands from IBM, and Habanero from the National Center for Supercomputing Applications at the University of Illinois.

NCSA's Habanero (NCSA, 1998) collaborative framework was chosen for our work due to the maturity of the software and accessibility of the source code. The Interactive Weather report applet has been cast as a collaborative application, and work is progressing on the use of Habanero with additional applets.

However, here too the current state of Java deployment is slowing down development. Fully collaborative applications demand a high level of network connectivity and access to local disk resources. Currently, this is only possible using standalone Java applications, or by limiting deployment of an applet and the necessary collaborative message server to a single web site. Existing security restrictions imposed by web browser suppliers puts up major roadblocks to fully collaborative applets running in the much more easily accessed and ubiquitous WWW environment.

VI. Conclusion

In spite of the rather slow maturation of Java during the past year, applets can still be used to present data effectively, and the future of Java development for data visualization and simulation tools seems secure. The coming year should see the release and adoption of useful APIs and packages, as well as a loosening of some of the security restrictions currently imposed on Java-enabled web browsers. These advances will make it easier and faster to develop useful scientific applets and applications, and make it much more feasible to incorporate collaborative functionality into them.

In addition to waiting for Sun and other corporations to bring Java technology up to speed, it is our hope that the earth science communities continue to jointly and separately develop useful and reusable Java components and packages. This would allow faster and better use of Java to access and analyze scientific data, to the benefit of us all.

Acknowledgments

The authors wish to thank the Project Horizon Java subgroup, especially Ray Plante, Bob McGrath, and Nancy Yeager, for helpful discussions.

Portions of the work were developed under Collaborative Visualization project, NSF Grant #RED-9454729, and Project Horizon, NASA Grant #CAN-OA-94-1.

References

Hall, S. E., Sridhar, M., Wojtowicz, D., Ramamurthy M., Wilhelmson, R. B., and Plutchak, J., 1998: The status and future of Hybrid Multimedia in education: The WW2010 Experience, Preprints, *Seventh Symposium on Education*, Phoenix AZ, Amer. Meteor. Soc.

NCSA, 1997: Horizon Image Data Browser Package v1.0. [Available on-line from <http://imagelib.ncsa.uiuc.edu:90/imagelib/Horizon/>.]

NCSA, cited 1998: NCSA Habanero version 1.0. [Available on-line from <http://www.ncsa.uiuc.edu/SDG/Homepage/habanero.html>.]

Plante, R.L., Goscha, D. Crutcher, R.M., Plutchak, J., McGrath, R.E., Lu, X., and Folk, M., 1997: Java, Image Browsing, and the NCSA Astronomy Digital Image Library, Preprints, *Astronomical Data Analysis Software and Systems VI*, San Francisco CA, Astronomical Society of the Pacific.

Plutchak, J., Wilhelmson, R. B, Ramamurthy, M. K., Hall, S. E., and Tokarskiy, V., 1997: The Weather Visualizer, Java, Habanero, and the Future, Preprints, *13th International Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology*, Long Beach CA, Amer. Meteor. Soc.

Sun Microsystems, 1997: Java Version 1.0.2. [Available on-line from <http://java.sun.com/>.]

Wojtowicz, D., Hall, S. E., Plutchak, J., Bramer, D., Ramamurthy, M. K., and Wilhelmson, R. B, 1998: The Technology Behind the Weather World 2010 Project WWW Content Management System, Preprints, *14th International Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology*, Phoenix AZ, Amer. Meteor. Soc.