DIS/SE/05

A Scientific Web-based Application for Global Tropical Cyclone Monitoring

FY 2003 Proposal to the NOAA HPCC Program

August 19, 2002

| Title Page | Proposed Project | Budget Page |

Principle Investigator: Dr. Mark D. Powell
Line Organization: OAR
Routing Code: R/E/AO/HRD
Address: NOAA/AOML
Hurricane Research Division
4301 Rickenbacker Cswy
Miami, FL 33149-1097
Phone: (305) 361-4403
Fax: (305) 361-4402
E-mail Address: Mark.Powell@noaa.gov

Co-Investigators:
Nirva Morisseau-Leroy Nirva.Morisseau@noaa.gov
Sonia Otero Sonia.Otero@noaa.gov

Proposal Theme: Crisis / Disaster monitoring or response

Funding Summary: FY 2003 $110,000.

Authorizing Manager:

Mark D. Powell	Frank Marks, Jr.	Kristina Katsaros
Principal Investigator	HRD Director	AOML Director
NOAA-HRD-AOML	NOAA-HRD-AOML	NOAA-HRD-AOML

A Scientific Web-based Application for Global Tropical Cyclone Monitoring

Proposal for FY 2003 HPCC Funding

Prepared by: Mark D. Powell, Nirva Morisseau-Leroy, and Sonia Otero

Executive Summary:

The advent of the Internet has spawned a whole new generation of Web-based applications that allow dynamic interactions with databases. The benefits of a web-based approach include ease of maintenance and development, and centralized distribution of data services. Historically, technologies such as HTML and Common Gateway Interface (CGI) have provided a mechanism for distributing dynamic content over the Web. Web programming techniques have matured rapidly since then to provide enterprise-level functionality, reliability, and scalability in business and scientific applications. Promising and increasingly popular technologies for dynamic content generation include Java Servlets and JavaServer Pages (JSPs). These new technologies are part of Java 2 Enterprise Edition (J2EE), which offers a powerful platform on which to build Web-based component applications. In year 2002, we tested the J2EE suite of products by transforming some of the functionality of the award-winning Distributed Real-time Hurricane Wind Analysis System (H*Wind) system into a web-based application running over the AOML intranet. In year 2003, we will continue development of the web-based application up to a mature level, feasible to serve as a global tool for monitoring tropical cyclones.

H*Wind introduction:

The HRD Real-time Hurricane Wind Analysis System (H*Wind) is a distributed system that ingests real-time atmospheric observations (from all over the world) measured by land-, sea-, space-, and air-borne platforms into an object-relational database, adjusts them to a common framework, and offers graphical capabilities and research tools to display the data relative to the storm so scientists can quality control, objectively analyze, and visualize the information. H*Wind's greatest strength is the ability to quickly evaluate and analyze observations from many diverse observation platforms. The quality control element of H*Wind is a complex graphical user interface written entirely in Java, and communicates with a database via Java Database Connectivity (JDBC) and SQLJ implementations. JDBC provides the capability of embedding dynamic SQL in Java programs whereas SQLJ allows the embodiment of static and dynamic SQL in Java programs. Applying Distributed Objects technology via Java Remote Method Invocation (RMI), the user can invoke an objective analysis (written in FORTRAN) from the Quality Control interface. After each successful analysis execution, color images of wind contours are automatically generated and posted on the Web. A more complete explanation of H*Wind characteristics can be found in Otero (2002). Since 2000, H*Wind has been evaluated by HRD scientists conducting real-time hurricane wind analyses every 3-6 hours in concert with the National Hurricane Center's forecast and warning cycle. Wind analyses are made available to the public on HRD's website at http://www.aoml.noaa.gov/hrd/Storm_pages/wind.html.

H*Wind was first demonstrated at the NOAA Tech 2000 Conference in October of 1999 where it won the "Best JAVA Implementation" award. In October of 2001, H*Wind was demonstrated at the NOAA Tech 2002 Conference and won the "Best Technology Transfer to Operations" award. In spring of 2001, H*Wind was rated by the National Hurricane Center as the highest priority research tool for transition to operations. As such, HRD was successful in obtaining support from the U. S. Weather Research Project's Joint Hurricane Testbed (JHT) for a two-year effort to transfer H*Wind to operations. During the current hurricane season, our goal is to train NHC forecasters to conduct H*Wind analyses as

part of a concurrent project with JHT. In addition, FEMA and NESDIS are providing support to provide satellite sensed hurricane wind measurements to H*Wind and ultimately provide hurricane wind field information to FEMA's HAZUS model for use in estimating damage and losses from hurricane landfalls.

The Problem: Limitations of the current H*Wind architecture

H*Wind is a multi-tier application consisting of a FAT quality control front-client and two back-end engines (analysis and database) distributed onto two Unix servers. In a fat client approach, the client contains logic for user interaction via a complex graphical interface, for performing requested algorithms, and for access to remote objects and database, where persistent data resides. Whereas the distributed nature of H*Wind and use of Java have many advantages in terms of code reuse, platform independence, and redundancy, the fat client approach has maintenance, distribution, and performance limitations. In particular, the tight coupling or integration of the Java objects providing the client functionality may require recoding when scientific logic and/or database processing change. Also, even though the code is written in Java, distribution of the H*Wind application would require assisting remote users in getting the software to run properly, namely the proper versions of Java virtual machine and Java Web Start. Another disadvantage of the fat client approach is the fact that a sizeable download must be performed to the client workstation and any interaction with the back-end engines is heavily dependant on the network traffic or available bandwidth.

Proposed solution:

Information technology resources are extremely limited in government research laboratories, hence economical approaches to performance, data delivery, software maintenance, and distribution are valuable. We have designed H*Wind as a global tropical cyclone wind field monitoring system, but users should be able to interact with it without the throughput overhead of physically delivering the observations. To overcome fat client limitations and make H*Wind more widely available, we propose to follow the J2EE model to develop a Web-based version of H*Wind.

J2EE is a set of standards that simplifies development and deployment of enterprise applications by basing them on standardized, modular components, by providing a complete set of services to those components, and by handling many details of application behavior automatically, without complex programming. The J2EE platform manages the infrastructure and supports the Web services to enable development of secure, robust and interoperable business applications

The J2EE programming model (Morisseau-Leroy,et al, 2002, Monson-Haefel, Alur, et al, and Roman, 2001) provides the ability to develop and deploy applications that take advantage of a wide range of new and evolving technologies. Due to its support for distributed applications, J2EE is now being used by many private (Oracle, IBM, BEA, Rational, and etc.) and government institutions. J2EE technology is now being implemented at the following government agencies:

- eia.doe.gov - The Energy Information Administration of the Department of Energy is using the J2EE technology to allow their customers to drop data files into their online system.
- www.nih.gov – The National Institutes of Health is moving from Oracle Forms to J2EE.
- www.cancer.gov – The National Cancer Institute is using J2EE for any new application design.
- www.ojp.gov – The Office of Justice Programs is also using J2EE for any new application design.

Note that all the above government institutions are using external consulting firms to implement the J2EE technology whereas, at HRD, we developed in-house expertise to do so.

The J2EE platform solves many of the problems associated with fat clients by distributing data and software applications from a centralized data center over a wide-area network. The J2EE platform allows loose coupling of components where the fitting components can live separately in different tiers (multi-tier architecture), by enforcing the separation of presentation, logic and persistence tiers. Thus, it reduces the need of changing client's code when scientific logic and database processing need to be modified.

The Java component modeling used in H*Wind is described in Morisseau-Leroy et al, 1999, 2000, 2001, and 2002. Portions of the proposed H*Wind Web-based application (using the J2EE platform) are discussed in Morisseau-Leroy 2001 and 2002. The J2EE platform is designed to support multi-tier architecture. Based on the J2EE architecture, the Web-based version of H*Wind will consist of several components and containers. A container is a service that provides the necessary infrastructure and support for a component to exist and for the component to provide its own services to clients. H*Wind's components and containers will be distributed in the following tiers:

1) Client tier: The client tier will interact with the end users and display information from the system to the end users. In the J2EE platform, HTML and Java applets in a client container implement this tier.

2) Web tier: The Web tier will generate the presentation logic and will accept user responses from the presentation clients (HTML or Java applet clients).
3) Business or scientific tier: Enterprise JavaBeans (EJB) components in an EJB container, and JavaBeans objects will handle the core business and scientific logic of the application.

Analysis:

Our objective is to create a thin-client, multi-tiered application version of H*Wind using the Java 2 Enterprise Edition (J2EE) product from Sun Microsystems. In a thin-client application, the logic to access remote objects and databases is removed from the client and moved to an application server or middle-tier server freeing the client from scientific logic and database processing that can be done more efficiently in the server. Application servers provide the basic structure required for developing and deploying multi-tiered enterprise applications. Another advantage of removing database processing from the client is the reduction of network traffic. The reduction of network traffic results in faster application downloads and less client RAM. With the release of J2EE, the combination of Enterprise JavaBeans (Roman et al, 2001, Monson-Haefel, 2001, and Morisseau-Leroy, 2002), Servlets, and JSP offers a powerful platform on which to build web-based component applications (Tait 2000). The EJB effectively encapsulates business and scientific logic while the JSP and Servlet Web-tier provides the HTTP-based data entry and presentation to the "front-end" tier for human interaction.

The Web-based version of the H*Wind system will rely on several cutting edge technologies including XML, applets, Java Servlets, Enterprise JavaBeans, and Java Server Pages (JSP). The static nature of HTML and JSP alone does not provide the required graphical infrastructure as available in the current application; therefore, we propose to use applets to achieve a similar interface quality level. XML provides information about data in a standardized format, and so it could play a vital role on the optimization of communication between the applet and Servlets. In multi-tier architecture, Java Servlets provide web developers with a simple, consistent mechanism for extending the functionality of a web server and for accessing existing systems. Servlets provide a component-based, platform-independent method for building web-based applications, without the performance limitations of CGI programs. We

propose to remove from the H*Wind client all remote object invocation logic and move it to Java Servlets. This should result in network traffic reduction, and better application and system performance. Enterprise JavaBeans component architecture extends "Write Once, Run Anywhere" capability to reusable component development. It is used for developing or assembling network-aware solutions for heterogeneous hardware and operating system environments -- within the enterprise or across the Internet. At all tiers, all the fundamental Java code pertinent to the current H*Wind stand-alone application, heavily tested for the past three years of operations, can be safely reused and adapted to comply with the J2EE architecture.

This research is consistent with the goal of NOAA's HPCC program to disseminate real-time and historical information to users more completely, in usable forms, and in a timely manner via the Internet. The analysis system will help to fulfill NOAA's strategic plan objective as research to advance short-term warning and forecast services. These uses of storm information are consistent with the findings of the World Disasters Report 1999 issued by the International Federation of Red Cross and Red Crescent Societies (EOS 1999), the National Science and Technology Council symposium "Real-time Monitoring and Warning for Natural Hazards" (EOS 1998), and recent comments by United Nations Secretary General Kofi Annan calling for a "culture of prevention" citing wars and natural disasters as "the major threats to security of individuals and human communities worldwide" EOS (1999). Our research goals are also consistent with the National Academy of Sciences (1996) report, "Computing and Communications in the Extreme" which identified challenges confronting crisis managers, including: 1) "need for cooperation among many different actors", 2) "need to rapidly identify, collect, and integrate crucial information about the developing situation", and 3) "capability to make projections and initiate actions in the face of an inevitable degree of uncertainty and incompleteness of information".

Performance Measures:
Milestones: Implementation of a Web-based Application requires two parts:
Part 1 (Months 0-6) Implementation of some of the presentation and middle tier:
1 . Implement layered image drawing, crucial for optimizing speed of graphical display.
2 . Convert current application graphical user interface to a friendlier thin-client applet.
3 . Handling of user interaction events in applet by Servlet controller.
Part 2 (Months 7-12)
1 . Develop EJB components to handle database queries
2 . Database access through Servlet and EJB components.
3 . Transform Analysis component from a Java RMI servant to an EJB.

Deliverables:
1. Servlet able to interact with data back-end, to query storm track positions and wind observations.
2. Servlet for dealing with presentation logic needed by applet.
3. Thin applet able to display image prepared by Servlet, and provide tools to inspect, zoom, flag or “unflag” observations, analyze the final observation set, and deliver the graphical analysis product to the client browser.

2002 HPCC Project Progress Report
Since receiving HPCC funds in April 2002, work focused on further testing of the prototype application to test the feasibility of our approach, which was demonstrated at the NOAATech 2002 Conference. The following have been accomplished (listed with the most recent first):

1. Updated web-interface to take advantage of new J2EE standards and to work with Oracle 9i Applications Server (9iAS): Oracle Containers for Java (OC4J). Communication with database EJB's optimized through use of new development models and architectures. 4/01//2002-present.
2. HURDAT file-based database loaded into Oracle object-relational database for Atlantic and East Pacific Basins with the use of an in-house Java application. 1/2002-.3/31/2002
3. The investigation of performance gain or loss in using JDBC and Oracle SQLJ implementations resulted in the selection of SQLJ for data ingestion and JDBC for query mechanisms. 1/2002-3/31/2002. The storage of a typical QcSet (~15,000 pointers) was unacceptable (over fourteen hours) when using JDBC. By using SQLJ, we were able to dramatically reduce storage time to thirty seconds.
4. Developed prototype front-end web-interface using Java Servlets and Java Server Pages (JSPs) to display storm tracks graphically and textually. Dynamically created images are sent to an HTML/JSP interface, allowing users to interact with the track. Changes and other display request (such as zooming) are sent to the Servlet where a new image is generated. 11/2001-3/31/2002.
5. Tested web-interface on Oracle8i Applications Server and Apache JServ. Demonstrated at NOAATECH 2002. Fall 2001.

Note: The end result of the H*Wind system is a variety of graphical analysis products derived from observations (and not the observations themselves); therefore, FGDC metadata are not required.

Project related References

Alur, D., Crupi, J., Malks, D., 2001: Core J2EE Patterns: Best Practices and Designs. Sun Microsystems Press; ISBN 0130648841.

Cade, M., et al, Sun Certified Enterprise Architect for J2EE Technology Study Guide, Prentice Hall PTR; ISBN: 0130449164, 3/2002

EOS, 1998: Real-time monitoring and warning for natural hazards can provide real-time benefits. AGU Transactions, p.329.

EOS, 1999: Report warns of more natural disasters. AGU Transactions, 317-319.

EOS, 1999: UN head calls for "culture of prevention" in addressing natural disasters. AGU Transactions, 438.

Monson-Haefel, Richard., Enterprise JavaBeans, O'Reilly & Associates; ISBN: 0596002262, 10/2001

Morisseau-Leroy, N., et al, October 2002: Oracle9iAS: Building J2EE Applications, Osborne McGraw-Hill; ISBN: 0-07-222614-5

Morisseau-Leroy, N., Solomon, M., and Momplaisir, G., June 2001: Oracle 9i SQLJ Programming, Osborne McGraw-Hill; ISBN: 0072190930

Morisseau-Leroy, N., Basu, J., and Solomon, M., 2000: Oracle 8I Java Components, Osborne McGraw-Hill; ISBN 0072127376

Morisseau-Leroy, N., Solomon, M., and Momplaisir, G., 1999: Oracle 8I SQLJ Programming. Osborne McGraw-Hill; ISBN 0072121602.

National Research Council, 1996: Computing and Communications in the Extreme.

Otero, Sonia, 2002: A Real-time Distributed Analysis Automation for Hurricane Surface Wind Observations, Master Thesis, School of Computer Science, Florida International University, Miami, FL.

Perrone, P., Changanti, V., 2000: Building Java Enterprise Systems with J2EE. Sams Press; IBSN 0672317958

Roman Ed, et al, Mastering Enterprise JavaBeans, John Wiley & Sons; ISBN: 0471417114, 12/2001

Tait, B., July 2000: Java Servlets and JSP: Separating Presentation from Business Logic. Java Developer Journal vol 5 issue 7 page 18