5.4 WEB PORTAL AND SERVICES FOR DELIVERING WEATHER INFORMATION

Earl V. Ravid* and LCDR Ken Malmquist
Fleet Numerical Meteorology and Oceanography Center
Monterey, California

1. INTRODUCTION

Technology continues to race forward at a breakneck pace. Web technology has propelled open standards and Open Source into the mainstream of application development. Small and large enterprises alike compete for 'mind share' in the arena many affectionately call: "The Web."

The Navy's Fleet Numerical Meteorology and Oceanography Center (FNMOC) is aggressively engaged in deploying new and existing applications using e-business, n-tier software and hardware architectures. Our objective is the timely delivery of meteorological and oceanographic data and visualizations using this technology. The present paper details the architecture and approach employed in these efforts.

2. BACKGROUND

FNMOC has been delivering data and visualization capabilities over the World Wide Web since 1994. Operational users all over the world access our web pages and web-based applications each day. The number and sophistication of these applications has continued to increase over this timeframe, and so has the maintenance cost.

In December of 2000, the Vice Chief of Naval Operations established the "Task Force Web" (TFW) initiative. It sets forth the goal of "Transforming Business and Operations with Web Technology." The result was the establishment of a task force to oversee the Navy's web-enabling efforts. To quote an example from the capstone document [TFW-Capstone, 2000]: "A 5,000 person division at Oracle subsists on four human resources professionals through the adoption of self-service." The clear and compelling message is that empowering Naval personnel through more self-service capabilities can yield huge savings.

To say that transforming an enterprise the size of the US Navy is daunting would be a gross understatement. Challenges include decades of "stove pipe" applications and a culture that requires a healthy intolerance to failure. Web development is fast paced and encourages rapid incremental development techniques. Users expect and demand applications that are responsive to their workflow and readily available on any computing platform they encounter.

Much of the risk of web development has been minimized by the myriad commercial entities publishing information and applications. Indeed, application tools are freely available to developers. From complete programming development environments available for computer languages like Java™ to source code control tools like CVS, there is no lack of resources available to web developers. E-business is entering its maturing phase, and there are ample techniques and technologies available to minimize the costs and risks of deploying applications. In this paper, we focus on two key components of the TFW effort: Web Portals and Web Services.

3. APPROACH

The Navy's approach to web enabling applications follows a conventional e-business model. Developers are encouraged to utilize open standards like the extensible markup language (XML), and the Simple Object Access Protocol (SOAP). Using Open Source tools like the Concurrent Versions System and Bugzilla saves time and money because of their large and active user communities.

The driving factors for web enabling applications are that they become: 1) interoperable; 2) scalable; 3) easier to develop / support; 4) are available on any computer connected to a network.

3.1 E-BUSINESS

It is beyond the scope of this paper to provide an in-depth description of e-business. We assume that the reader is familiar with the general topic. An accepted rendering of the standard n-tier, e-business architecture with the Navy's Enterprise Application architecture integrated is shown in Figure 1.

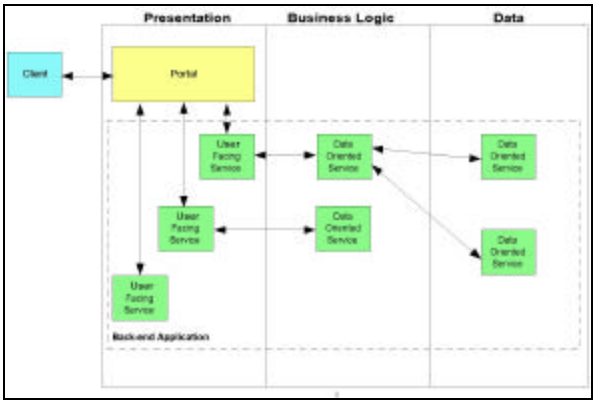

Figure 1

* Corresponding author address: Earl V. Ravid,
Fleet Numerical Meteorology and Oceanography Center,
7 Grace Hopper Ave., Monterey, CA 93943-5501
ravide@fnmoc.navy.mil

The familiar distributed tiers of Presentation, Business Logic, and Data are represented in Figure 1, which is part of the "Navy Enterprise Application Development Guide" [NEADG 2002]. This approach of separating the functions of an application is what permits one to scale the computing resources available at each tier. Moreover, if development is constrained to be operating system and programming language agnostic, risks and costs are reduced.

The Navy Enterprise Application Development Guide further defines a framework for the implementation of applications. Referring again to Figure 1, a typical application will implement two types of components: a User-Facing Service (UFS) and a Data Oriented Service (DOS). These topics will be elaborated upon in subsequent sections.

At the Presentation tier, a web portal provides the interactive environment for browser clients. For the benefit of readers unfamiliar with portals, a portal is a technique of presenting information in a browser such that there is the appearance of multiple mini-applications (a.k.a. portlets). These portlets can be spatially placed and sized by users. Adding or removing portlets is controlled by users through a preferences portlet interaction. An example of a simplified portal is My Yahoo!, where signed-in users have some limited control over their browser's content.

One of the most significant advances in web technology is the invention of the web service. In short, a web service is any software service that exposes its interface via the web over a web-supported protocol. For example, one could expose a web service to report the current value of the Dow Jones Industrial Average. Such a service could be called by an application that uses the returned data to perform a host of statistical output and store those results in a database for later viewing. In this example, two programs without any presentation client are communicating (interoperating). This paradigm is rapidly being adopted for business-to-business transactions (e.g., banks, point of sale devices). In the context of the METOC business, a service that would provide a derived field based on a mathematical expression would be of value to applications and clients alike.

3.2 USER FACING SERVICES

The interface between a UFS and the Portal follows open standards. A UFS produces an XML file containing the content for a portlet, and an Extensible Stylesheet Language (XSL) file detailing how to present the content. The Navy Enterprise Portal assumes the responsibility of applying the transformation of the content (XML) according to the presentation style (XSL) and renders the result in a portlet.

User Facing Services can be implemented in any programming language provided they can be invoked via a web server (by a Common Gateway Interface (CGI) or a Java™ servlet). For example, if a portlet's content is a form that allows one to search for information and the user presses the 'submit' button, a Perl program could be executed via CGI. The result of such an execution would be an XML file and an XSL file that is returned to the portal engine, which then renders the content for presentation in the portlet.

Applications that exceed the features of simple forms content will likely choose a more robust programming paradigm than a scripting language. However, buffer overruns and other kinds of exploits make traditional programming languages like C or C++ too risky for web development. Choosing programming languages that support robust application construction is necessary for both the initial development as well as long-term maintenance. In fact, a rapid spiral development methodology requires a software architecture that supports such incremental improvement.

For UFS development, we chose Java™ for its web-centricity, object orientation, rich object libraries, strong security model, and the availability of talented programmers. Within this programming language domain, there are many choices of applications and middleware that can aid in rapid development. For UFS development, we chose STRUTS [STRUTS, 2002], an Open Source framework for building Web applications developed under the Apache project. Following a modified Model-View-Controller software pattern, STRUTS implements an extensible development environment for applications, based on published standards and proven design patterns.

Choosing Java™ allowed us to leverage the Open Source STRUTS application framework for building robust, object oriented enterprise web applications. Java™ is committed to open standards as evidenced by the many Java™ libraries supporting XML, SOAP, security, and other web-oriented development.

3.3 DATA ORIENTED SERVICES

The business logic and data functions of applications are implemented as Data Oriented Services. These services interact via SOAP messages [SOAP, 2002] transmitted over https protocol. This stateless protocol transmitted over secure connections is the heart of web services.

Many of the legacy applications that make up the mainstay of meteorology and oceanography applications were constructed before the web became ubiquitous. There are many useful and significant applications that are "hidden" in enterprises. Migrating these into the web-centric world might be akin to mining jewels. This requires new thinking about tried

and true applications. We must be willing to 'refactor' these legacy capabilities to make them shine in the light of new technology.

Data Oriented Services are programmed to respond to invocation from a web services engine supporting SOAP. The invocation passes input information to the DOS in the form of XML content (e.g., input parameters). The DOS executes according to its input and produces output in the form of XML content (i.e., results) which are returned to the web services engine.

3.4 A ROUND TRIP THROUGH THE PORTAL

With a basic understanding of a Portal, UFS, and DOS components, we can examine their interaction by the flow of interaction from client request (e.g., pressing a button in a portlet application) through the fulfillment of that request. This process is diagramed in Figure 2.

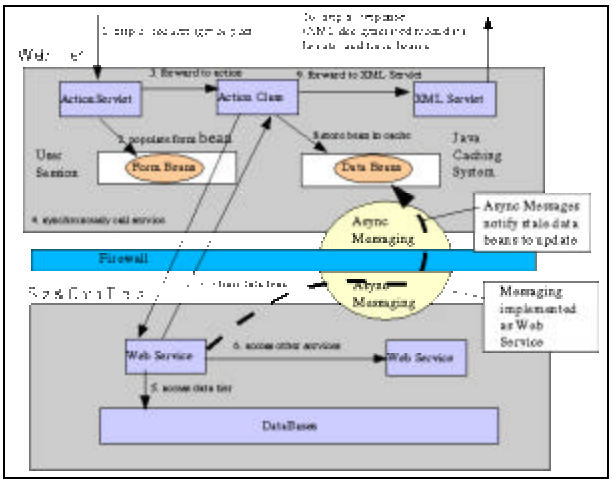

Figure 2

The steps are numbered sequentially. In this figure, "Web Tier" is synonymous with Presentation tier. Starting from the incoming http(s) request, continuing through an action class to the Data Oriented Services of the business tier, back to the UFS, and out through an XML Servlet (or through a Java Server Page (JSP)) to the requestor. Note the use of a "Form Bean" to conveniently encapsulate user input. This is a standard feature provided by the STRUTS framework. Also note the use of a "Data Bean" to package data returned from the business layer. As the figure indicates, an asynchronous messaging system provides invalidation notification to Data Beans that might cache their data.

4. CONCLUSIONS

FNMOC has embraced the Web Enabled Navy goals. We are engaged in a transformation of how we conduct our business; focusing on supporting our customers through a Navy-wide application portal.

We presented how we are approaching web enabling our applications in a broad-brush manner. However, we hope that some of the details will be of benefit to our colleagues in education, industry, and government as they face the challenges of a new way of application deployment.

Many meteorology and oceanography applications are highly interactive. We have implemented applications that have reasonable interactivity [Ravid 1999, 2000], but current web browser technology is not up to the challenge of replacing highly optimized interactive programs like Vis5D. However, we expect that support for interactive graphics is on the horizon.

The following is our list of the benefits of applying e-business technology to the meteorology and oceanography domain:

1. Scalability
2. Interoperability of applications
3. Leverage Open Source and standards
4. Reduced development risks and costs
5. N-tier architectures can be deployed in an 'accordion' fashion – distributed across multiple hardware systems or all on one.

The TFW initiative adds to these benefits. Users are empowered to tailor workspaces in a portal to their workflow. Also, developers register their applications with the portal's directory services so that users are not limited to a particular domain of services. Application developers benefit from a directory of published capabilities and can construct new applications that call upon existing web services. Central repositories for application registration encourages 'best of breed' choices.

ACKNOLWLEDGEMENTS

The authors would like to thank the members of the FNMOC Web Portal / Services development team for their professional efforts and dedication.

5. REFERENCES

NEADG, 2002: Navy Enterprise Application Development Guide, Version 0.07, https://tfw-opensource.spawar.navy.mil

Ravid, Earl, et. al., 1999: High Volume METOC Chart Production Using Beowulf Technology. In *preprints, of 16th Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology*. January 9-14, 2000 Long Beach, CA.

Ravid, Earl, et. al., 2000: CWxMAP – Customer-Driven Interactive METOC Charts Via The WWW. In *preprints of the 17th Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology*., January 14–19, 2001, Albuquerque, New Mexico.

STRUTS Web Pages: http://jakarta.apache.org/struts/index.html

TFW-Capstone document, 2000: http://www.tfw.navy.mil; *Archives; History - Task Force Whiskey;* TFW Web Enabling the Navy CAPSTONE Document