

Portal Development Tools

Konstantinos Robotis

University of the Aegean, Greece

Theodoros Tzouramanis

University of the Aegean, Greece

INTRODUCTION

A number of customer and company needs can be identified, to which Web portal services appear to provide the best answer. These needs include information sharing and accessibility, content management and abstraction with the use of interfaces to multiple data formats, such as physical data, documents, files and databases. The main purpose of a Web portal is to enable users to access and modify information.

This short survey will provide an overview on how portal technology has evolved and how it has acquired a share of the market. The survey will examine the interoperability and integration of portal application products with other technologies and systems; it will then attempt to predict the future of Web portals development; in a next stage it will try to deal with the questions that vendors encounter in the portal implementation process and it will take note of some of the customer requirements. An essential question in dealing with the above concerns whether the next generation Web portal products will be designed as autonomous software independent systems, or as parts of other bundled systems.

The survey will also refer to the dominant players in the field of Web portal development tools. It will address the similarities and differences of the approaches of the various corporations. Each of these products will be briefly described and both its distinctive features and the requirements that it fulfils will be presented. The survey will conclude with a reference to the most significant trends in Web portal technology and to the needs that this technology will satisfy in the near future.

BACKGROUND

Reading section 2.1 of the Portlet Specification (Sun, 2003), a portal is defined as:

a Web based application that—commonly—provides personalization, single sign on, content aggregation from different sources and hosts the presentation layer of Information Systems. Aggregation is the action of integrating content from different sources within a Web page. A portal may have

Table 1. Leading Web portal vendors and corresponding products

Portal Vendors	Product
IBM	IBM WebSphere Portal
Microsoft	Microsoft SharePoint
Oracle	Oracle Application Server Portal
SAP	SAP Enterprise Portal
Sun	Sun Java System Portal Server
BEA Systems	BEA WebLogic Portal

sophisticated personalization features to provide customized content to users. Portal pages may have different set of portlets creating content for different users.

A number of products to address the issues described in the above definition are available. The vendors selected for evaluation are listed in Table 1. The corresponding products possess the lion's share in the portal market (Phifer, Valdes, Gootzit, Underwood & Wurste, 2005).

Along with the commercial portal products, a multitude of open source portal frameworks compliant with JSR and WSRP have been developed. The latest JetSpeed 2.0 (Apache, 2006) from Apache is an implementation of an Enterprise Information Portal written in Java and XML. GridSphere portal framework (GridSphere, 2006) is an open source framework fully compatible with IBM's WebSphere.

TECHNOLOGICAL EVOLUTION OF WEB PORTALS

When the first Web portal products came out, they had entirely proprietary APIs, and a different set of features. Some had personalization features, while others excelled at enabling workflow or content management (Richardson, Avondolio, Vitale, Len, & Smith, 2004). Along the way, Java 2 Platform

Enterprise Edition (J2EE) began its meteoric rise, providing strong enterprise application development and integration features. It also had a substantial impact on Web development activities with the servlet and JSP specifications.

Quickly, though, the portal capabilities caused technical discriminations, and provided nonstandard extensions to J2EE. These extensions ranged from very close approximations of the standard components, to full-blown rewrites of presentation logic code (Richardson et al., 2004). The problem was that the portal implementations were starting to fracture the J2EE application base, which had a negative effect on the portability of enterprise applications.

To deal with this problem, two standards have been adopted by many of the prominent portal vendors. Rather than compete with one another, these standards compliment one another. In 2003, Sun in cooperation with dominant portal vendors, proposed the JSR 168 specification (Sun, 2003). JSR 168 defines a Java Portlet API for Web application components (portlets) that interact with and can be aggregated in applications such as portals.

Additionally, Web services for remote portals (WSRP) ratified an OASIS standard that views the portal and Web service interaction from a completely different angle, defines visual, user-facing Web services that plug and play with portals or other applications (OASIS, 2003).

Of course, the diverse group of portal vendors presently in the market will offer differing sets of components to leverage within their portal, even addressing points where the portlet specification in its current state falls short, such as interportlet communication, portlet filters, extending the CSS support, and integration of existing Web application frameworks to be leveraged in portlet development, such as JSF/My Faces, Struts, and Spring MVC (Viet & Russo, 2005).

BASIC WEB PORTAL DESIGN CONCEPTS

This section describes the design aspects that define the technological side of a portal platform. The first four criteria stated are presented by Homan and Klima (2001). It is important to emphasize that the majority of portal solutions today comes as a part of an application server platform providing these functionalities by the application server and not by the portal product. However in case of stand-alone portal solutions, these services must be implemented directly by them.

- **Fault Tolerance and Clustering:** Each portal server instance should also be able to pick up the load when other instances crash either from a hardware or software failure. Load balancing is usually provided between multiple instances of the portal server using algorithms

that take a round-robin-based approach (Tanenbaum, 2001).

- **Caching:** Local data storage so that acceptable response times are preserved since response times from the distributed systems may otherwise be very long as a result of the load of aggregated data some of which may originate from geographically distributed sources.
- **Repository Structure:** Web portal servers use repositories to store security information and metadata among others. Choices for storage range from flat files or serialized objects to relational databases or lightweight directory access protocol (LDAP) directory servers.
- **Platform Support:** Because Web portals are designed to work in a cross-platform environment, support for multiple operating systems application platforms (such as those from BEA, Oracle or IBM) is highly desirable.
- **Standards Support:** Standards-based portals are a must. If the product complies with portal standards, such as WSRP and JSR-168, and Web services standards (SOAP, WSDL), users are assured that third-party products will work with it and the applications they develop will work with the existing infrastructure (MacVittie, 2004).
- **Security Services:** Portal security encompasses a range of technologies that address the issues of authentication, integrity and confidentiality so as to support mechanisms to ensure these features (Richardson et al., 2004).

DOMINANT VENDORS IN THE FIELD AND THEIR APPROACHES

The top Web portal solutions will run on common J2EE application servers (such as IBM WebSphere or BEA WebLogic) or .NET, or both. According to Heck (2004), a difference between two otherwise closely matched products is: whether a portal runs best on a vendor's own platform and how well it truly integrates with existing enterprise systems.

IBM WebSphere Portal

WebSphere Portal (IBM Corporation, 2006) is typically built on top of the J2EE-compliant WebSphere application server. The portal server provides development and runtime infrastructure for the portal. WebSphere Portal itself installs as an Enterprise application in WebSphere Application Server. The portal infrastructure allows load balancing, fault tolerance, caching and external security management (IBM Corporation, 2006).

5 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/portal-development-tools/17953

Related Content

Semantic Web Portals

Shouhong Wang and Hai Wang (2007). *Encyclopedia of Portal Technologies and Applications* (pp. 901-904). www.irma-international.org/chapter/semantic-web-portals/17983

Dataspaces Enhancing Decision Support Systems in Clouds

Michele Argiolas, Maurizio Atzori, Nicoletta Dessi and Barbara Pes (2012). *International Journal of Web Portals* (pp. 35-55).

www.irma-international.org/article/dataspaces-enhancing-decision-support-systems/73914

Wireless Local Communities in Mobile Commerce

Jun Sun (2007). *Encyclopedia of Portal Technologies and Applications* (pp. 1204-1209).

www.irma-international.org/chapter/wireless-local-communities-mobile-commerce/18031

A Formal Approach for the Validation of Web Service Orchestrations

Wael Sellami, Hatem Hadj Kacem and Ahmed Hadj Kacem (2013). *International Journal of Web Portals* (pp. 41-54).

www.irma-international.org/article/formal-approach-validation-web-service/78352

How Thick Is Your Client?

Ed Young and Michael Jessopp (2012). *Enhancing Enterprise and Service-Oriented Architectures with Advanced Web Portal Technologies* (pp. 131-141).

www.irma-international.org/chapter/thick-your-client/63950