

Chapter 2.2

Building Portal Applications

Jana Polgar

NextDigital, Australia

Tony Polgar

Oakton, Australia

ABSTRACT

Software development methodology refers to a standardised, documented methodology which has been used before on similar projects or one which is used habitually within an organisation (Software development methodology). It can generally be applied to all kinds of software products. In portlet development process there are new circumstances which affect the methodology. A portal development manager must be aware of the technological properties and constraints, because there is a large (and very new) range of issues, risks and hidden costs that must be addressed in both the development and deployment processes. This article focuses on discussion of practical approaches to the resolution of development issues and risks in Portal environment. The discussed topics include implementation of Portals in enterprise environment, portlet applications' high availability, portlet disaster recovery, and the

cost of portlet deployment. An attempt is made to forecast future trends in portlet technology.

BACKGROUND

Enterprise Portals entered the business scene as a new generation of integration services, in a logical sequence of creating easier access paths to service oriented architectures. One can regard Portals as a happy marriage between network enabled access through Web, and specialised, business focused access to grouped information and functions. Development of portlets has been originally regarded as a yet another metamorphosis of J2EE or .NET technology. The expectations and promises of Portal suppliers included powerful user interfaces, fast development using rich APIs, compatibility of portlets originating from different suppliers, (JSR 168 & JSR286 define the portlet specification), integration of content, and document management

with functional portlets, single sign-on, and easy implementation of authorisation/authentication services as well as application services such as document management, and version management. The vendors attempt to ease the cost and time of the development by bringing new in new tools and environments. A number of questions arose immediately:

Is the development as mature as it would appear from the above promises? Can a development manager with experience in other Web technologies and distributed applications easily become a successful portal development manager? Is there anything specific that a portal development manager must know about the technology? Are the best practices in Web development applicable to portlet development? What are the hidden costs and pitfalls of portal development? Are the new development environments as efficient as claimed? How often can the development manager introduce new tools for the developers?

This article aims at answering these questions. We start with providing the foundations of portal technology, a brief note on the portal specification JSR 168, and WebSphere Portal 6 overview. Armed with the technology landscape, we will look at the issues associated with the development of portal applications. Please note that our practical experience is predominantly in IBM's WebSphere Portal 6.

Brief Look at Portal Technology Today

Portal applications follow typical J2EE multitier architecture. We can view portals as an extension of the Web tier. The portlet container (portal server) is an extension of a servlet container. The portal server manages portlets' lifecycle, provides a runtime environment, and redirects client requests to appropriate portlets.

A portlet is an application that displays some content in a portlet window. A portlet is developed, deployed, managed and displayed independently

of all other portlets. Portlets may have multiple states and view modes and they can also communicate with other portlets by sending messages. Physically, portlets are fragments of HTML placed on a portal page typically in rows and columns. Each portlet provides access to a business function or a Web service (*Polgar, J.(2006)*).

Portlets are organized into portal pages. In turn portal pages are organised as a set of navigation nodes. Each page is defined in terms of templates for the portal window, screen, and rows and columns. Each screen also contains look-and-feel components such as themes and skins.

Each portlet is characterised by its mode and state. Portlet modes are properties of the portal presentation model. This model can be implemented differently by different infrastructure vendors. Portlet modes allow the portlet to display a different "face" depending on its usage. There are four modes in IBM's Websphere Portal Server:

1. **VIEW** is the initial face of the portlet when created. It generates markup visualizing the portlet state and properties.
2. **HELP** supports the help mode (clicking on the control results in a help page being displayed for the user).
3. **EDIT** mode produces markup to enable the user to personalize the portlet.
4. **CONFIGURE** mode as provided in WebSphere Portal 5.1 and higher versions displays a face that allows the portal administrator to configure the portlet for a group of users or a single user.

The following examples demonstrate the results of changes in the EDIT mode. Figure 1 shows the portlet called *ActionEvent* before any EDIT was actioned. Figure 3 shows a view of a portal page after the EDIT mode button has been pressed. The EDIT mode button is the icon in the right upper corner of the portlet, in the shape of pencil—see Figure 3. The user selects a *Red Action* button or *Blue Action* button as shown in

14 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/building-portal-applications/43947

Related Content

A Goal-Oriented Representation of Service-Oriented Software Design Principles

Alireza Moayerzadehand Eric Yu (2011). *Non-Functional Properties in Service Oriented Architecture: Requirements, Models and Methods* (pp. 120-144).

www.irma-international.org/chapter/goal-oriented-representation-service-oriented/52232

Context, Gender and Intended Use of Mobile Messaging, Entertainment and Social Media Services

Anna Sell, Mark de Reuver, Pirkko Waldenand Christer Carlsson (2012). *International Journal of Systems and Service-Oriented Engineering* (pp. 1-15).

www.irma-international.org/article/context-gender-intended-use-mobile/64196

Service Risk Management in Emerging Economies

Yilmaz Akgunduz (2017). *Promotional Strategies and New Service Opportunities in Emerging Economies* (pp. 90-115).

www.irma-international.org/chapter/service-risk-management-in-emerging-economies/175550

Cost Effective for Erlang Traffic of Mobile Learning Over the Clouds

Khaing Sandar Htun (2014). *International Journal of E-Services and Mobile Applications* (pp. 1-11).

www.irma-international.org/article/cost-effective-for-erlang-traffic-of-mobile-learning-over-the-clouds/115861

Requirements Engineering for Cloud Application Development

Holger Schrödland Stefan Wind (2013). *Cloud Computing Service and Deployment Models: Layers and Management* (pp. 137-150).

www.irma-international.org/chapter/requirements-engineering-cloud-application-development/70138