



WebSphere software

WebSphere Portal: An on-ramp to a service oriented architecture.

Robert Duffner
IBM Software Group

Contents

- 2 Introduction**
- 3 Demystifying SOA**
- 5 Starting with WebSphere Portal
as your on-ramp to an SOA**
- 6 The evolution of portals**
- 8 Introducing composite
applications**
- 17 Putting it all together**
- 18 IBM SOA foundation**
- 19 Conclusion**
- 19 For more information**

Introduction

The use of portals and portal technology continues to mature across the technology adoption lifecycle as pragmatists in global organizations implement technologies with proven track records and demonstrable benefits. In fact, a survey from Morgan Stanley of FORTUNE 1000 CIOs indicates that portals continue to remain at the top of technology spending priorities.¹ How do you make portal investments pay off? Successful portal projects must be succinctly defined so that both information technology (IT) and business stakeholders agree on the scope and the expected business value to be delivered. Although this sounds like a relatively simple principle for organizations undergoing portal projects, it is much more difficult in practice.

Recent data from Forrester Research shows that portal projects struggle to deliver business value primarily due to the lack of alignment between IT and urgent business needs.² This problem is further compounded by the resulting complexity that exists in many IT shops today, namely, disparate legacy systems, applications and databases. The costs associated with maintaining this IT infrastructure leaves fewer resources available for spending on strategic portal initiatives.

As organizations search for a way to leverage a service oriented architecture, many can use portal products as a first step.³

On Demand Business defined

An enterprise whose business processes – integrated end-to-end across the company and with key partners, suppliers and customers – can respond with flexibility and speed to any customer demand, market opportunity or external threat.

The imperative for a cogent portal strategy could not be timelier as organizations seek to make IT not only responsive to new business initiatives and challenges but also a strategic weapon in gaining competitiveness and growth. As such, service oriented architectures (SOAs) are gaining traction with many CIOs as they seek to simplify and standardize their IT infrastructures in an effort to make IT a strategic component of the business strategy.

This white paper from IBM presents an overview of the capabilities of IBM WebSphere® Portal as a strategy and logical path into an SOA. IBM WebSphere Portal is uniquely positioned to deliver a product that best equips your organization to compete in a world of On Demand Business. SOAs are part of IBM’s blueprint designed to help your organization pinpoint and build the right kind of IT infrastructure that best supports your needs.

Demystifying SOA

The need for taking a more componentized approach to software and infrastructure development is nothing new to many IT professionals. The concept of drawing boundaries around software with well-defined interfaces dates back to the 1970s, when software applications were first designed. What makes componentizing infrastructure any different today?

A service oriented architecture (SOA) is an approach that enables business processes to be assembled from reusable components or “services” that are independent of applications and the computing platforms on which they run. Services are the building blocks for reuse!



The emergence of SOA based on Web services represents a shift in the dominant software paradigm, comparable to the shift from terminal-based to client/server architectures.⁴

First, industry standards with which to create these components and have them communicate have evolved and have been agreed upon by major vendors. In conjunction with broad industry support, we have now moved beyond simple connectivity and are tackling business level interoperability, which results in a much clearer return on investment. Second, the infrastructure to support self-defined, loosely coupled services has emerged. And finally, tools are now available to incorporate existing assets instead of reimplementing them.

Technology has evolved to a point where, for the first time in the evolution of the business/technology relationship, it's the business principles and the needs of the organization that can drive the way in which technology can execute on those needs. It is through a framework called service oriented architecture (SOA) that this relationship comes alive. If achieving On Demand Business is the organizational goal, then an SOA is the answer to the question, "How do I get there?"

IBM defines an SOA as an approach that enables business processes to be assembled from reusable components or "services" that are independent of applications and the computing platforms on which they run. It is an architecture that leverages open standards to represent software assets as services. A service in SOA is an application function packaged as a reusable component for use in a business process (e.g., processing a purchase order or verifying a credit card transaction). What makes an SOA come to life is when these services are loosely coupled (i.e., the service consumer has no knowledge of the technical details of the service provider), have well-defined interfaces that are preferably standards-based and are reusable.

Table 1. Web services standards

Extensible Markup Language (XML)

XML is a W3C-recommended general-purpose markup language for creating special purpose markup languages capable of describing many different kinds of data.

XML Schema

XML Schema is an XML language for describing and constraining the content and legal structure of an XML document.

Simple Object Access Protocol (SOAP)

SOAP is a standard for exchanging XML-based messages over a computer network, usually using HTTP.

Web Services Description Language (WSDL)

WSDL is an XML format published for describing the public interface to Web services.

Universal Description, Discovery and Integration (UDDI)

UDDI is a platform-independent, XML-based registry that enables businesses (or systems) to discover each other and define how they interact over the Internet. It is designed to be interrogated by SOAP messages and to provide access to WSDL documents.

Standards are critical to achieving On Demand Business because they provide the rules that ensure interoperability between the disparate technologies and products of vendors. Moreover, support for open standards and adoption of the technologies supporting SOA, namely Web services, are helping gain traction with many IT executives today (see Table 1). SOA's big promise is the ability to work very well in heterogeneous IT environments. One of the biggest challenges many IT organizations face is the inability for IT to be responsive to the business because their infrastructure and business processes are in a highly inflexible state. These legacy applications cannot be modified in a timely fashion to support the changing business requirements. SOA can fundamentally transform IT infrastructure from a barrier to business to an enabler of business change. Gartner predicts that by 2008, SOA will provide the basis for 80 percent of all new IT development projects.⁴

Starting with WebSphere Portal as your on-ramp to an SOA

Organizations that have deployed a portal solution on a portal platform over the past three to four years did so for many of the same reasons responsible for SOA's emergence today, namely, providing a component-based model, delivering reusable software assets, simplifying integration and increasing the ability to leverage existing IT assets. In addition, several key portal standards have been emerging over the same time period that provide the technology that enables service oriented applications. This is what has been fueling adoption of portals by large and midsize businesses as they continue to roll out deployments of customer-, partner-, citizen- and employee-facing enterprise portals. And IBM has extensive global experience working with clients on a variety of successful portal initiatives that have standardized on the industry's leading portal product, IBM WebSphere Portal.⁵ Identifying a specific portal project offers a great opportunity to leverage a tactical entry point into the service oriented capabilities of WebSphere Portal.

Table 2. A brief history of the portlet

Portlets are reusable Web components that display relevant information to portal users. Their emergence began in the late 1990s as an adjunct to Internet search engines that were popularized by consumer portals (such as Yahoo!). Common examples include e-mail, weather, stock quote and news portlets.

In 1999, the enterprise portal market was created as vendors began releasing portal server products to the market. The portlet concept was introduced into the enterprise portal as the primary interaction point for end users. Since then, portlet standards have emerged such as JSR 168 and WSRP that allow for the “plug-n-play” of portals.

Today, the portlet acts as a key integration point for accessing, aggregating and orchestrating data, applications and business processes at the user interface. As such, portlets are a rudimentary component of an SOA.

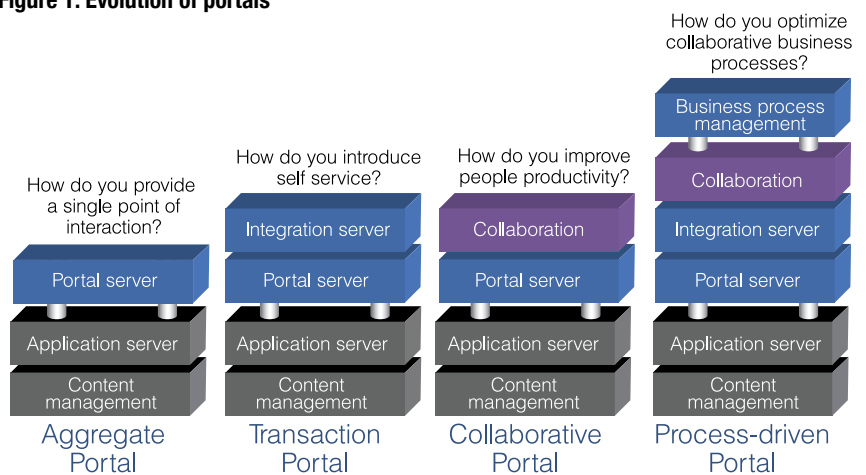
The evolution of portals

Portals continue to undergo a natural evolution and transformation as the needs of the business user mature and become more demanding (see Figure 1). Portals started out as a way to enable users to gain personalized access to content and applications. Many of these early portals were built on proprietary application servers and provided capabilities to simplify departmental portal deployments by providing a “portal-in-a-box” solution.

As portals gained in popularity and portlets emerged as the primary component of interaction (see Table 2), transaction portals were built to provide deeper integration with back-end systems, allowing for self-service applications to be deployed. At the same time, Web-based collaborative applications emerged as a way for people to be productive online with messaging, document sharing, Web conferencing and shared team workspaces.

But as organizations begin rationalizing their infrastructures and improving their operational efficiencies, process-driven portals are emerging as the preferred approach to optimizing collaborative business processes across the enterprise. They accelerate the adoption of best practices by making them the easiest route to accomplish the user’s task. This new generation of portals requires advanced integration architecture and a complete portal framework to automate existing business processes through technology.

Figure 1: Evolution of portals



IDC analysts believe that a new user work environment will emerge in the next five years that will be supported by a new, unified, modular enterprise software stack. This “enterprise workplace” will dramatically improve interactions across applications and between workers.⁶

Table 3. Understanding composite applications

The best way to understand composite applications is through the use of this simple formula:

CA = Assembly + Orchestration

Assembly requires the aggregation of components or reusable parts into services.

Orchestration determines how a process will flow within an application.

It is the composite applications that are built using reusable, loosely coupled components with well-defined interfaces that provide the foundation of an SOA.

IBM WebSphere Portal is the ideal solution for delivering on this new generation of process-driven portals. WebSphere Portal allows you to design a portal based on the perspective of the audience of users interacting with the portal. Understanding the audience means providing the right context within the portal and delivering the right content, applications and processes based on the user’s role. This approach reduces the burden on the user by hiding the underlying complexity of the various interfaces and applications being accessed. The portal adapts to the user’s needs and delivers a composite view that allows the user to interact with these multiple back-end systems as if they were one. By consolidating access to the content, applications and processes relevant to the user – all in one portal view – you simplify the user’s tasks and provide audience value, resulting in improved productivity and adoption of the portal.

Introducing composite applications

Process-driven portals deliver roles-based composite applications by aggregating business logic components into the user interface as a composite view. Composite applications go beyond just aggregating information into a single view. Their real value is to pick up the assembly of the components and orchestration of the processes that automate and optimize the common business transactions and decision-making processes where typical enterprise applications leave off. Composite applications are allowing organizations to span extended processes that involve multiple constituents both inside and outside the enterprise. It is the human component involved in these processes that give portals an indispensable role in the development and deployment of composite applications.

IBM WebSphere Portal delivers a complete set of capabilities that enable the assembly and orchestration of composite applications. The list of capabilities in the first column of Figure 2 is referred to as “on the glass” because the assembly and orchestration are at the user interface layer. In the second column, “under the covers” refers to the capabilities provided in the application and business logic layer. These capabilities are provided by the IBM WebSphere application infrastructure platform. WebSphere Portal takes full advantage of these capabilities as it is built atop the IBM WebSphere Application Server.

Figure 2: WebSphere Portal capabilities that enable composite applications

	“On the glass”	“Under the covers”
Assembly	<ul style="list-style-type: none">• Portlets<ul style="list-style-type: none">-JSR 168-WSRP-Mainframe integration• Web services• Web clipping• JavaServer Faces	<ul style="list-style-type: none">• Adapters• Message broker• Web services• Service Data Objects• Information services (e.g., XQuery, SQL)
Orchestration	<ul style="list-style-type: none">• Human task list (BPEL)• Cooperative portlets (property broker)	<ul style="list-style-type: none">• Choreography engine (BPEL)• Business rules

Composite applications have the potential to break down the barriers that operational applications impose and enable cross-functional and interorganizational processes.⁷

WebSphere Portal as a portal technology contains more than the “basics” of just information integration, Web content and document management, and application access. It provides an integrated portal business framework that encompasses the capabilities necessary to truly impact an organization’s daily business processes through the development and deployment of composite applications.

The following summarizes the key capabilities of WebSphere Portal that enable the assembly and orchestration of composite applications.

Assembly: Portlets

The portlet probably best embodies the idea of a reusable part. In the case of WebSphere Portal, it is a Web component managed by the Java® 2 Enterprise Edition® (J2EE) container, WebSphere Application Server. It provides the integration point to process requests and generate dynamic content, thereby acting as the user interface component at the presentation layer of the portal. WebSphere Portal supports JSR 168, a standard application programming interface (API) for creating portlets as the integration component between applications and portals on a J2EE platform. This standardization enables these portlets to “plug-n-play” across various platforms. For example, Business Objects has created JSR 168 portlets that enable you to integrate Crystal Reports inside a WebSphere Portal environment for true “plug-n-play” operation. The ability to rapidly develop and deploy portlets into a portal has been enhanced by IBM Rational® Application Developer, the design time assembly tool for portlets developed and deployed in WebSphere Portal. It is these portlets and the underlying framework that drive organizations to standardize on a common enterprise portal platform.

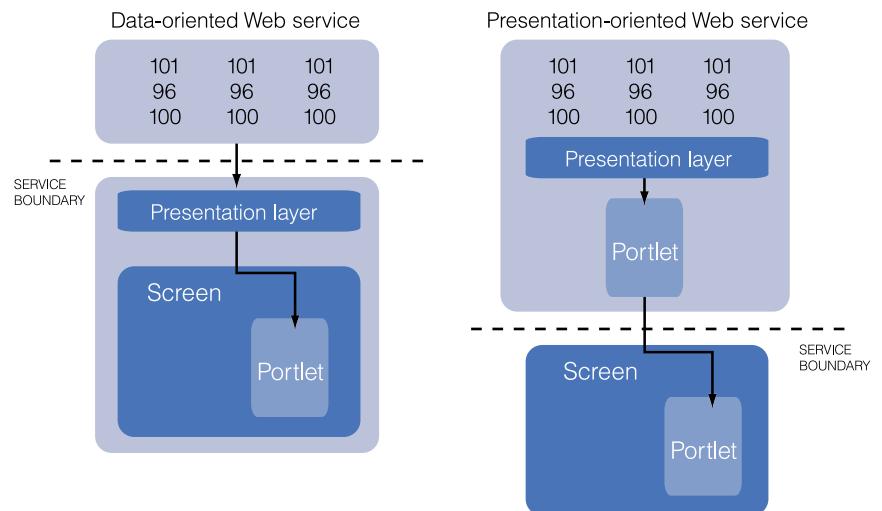
Assembly: Web services

Web services are becoming the predominant method for making information and applications available programmatically via the Internet. Portals are playing an important role in the utilization of Web services for composite application assembly. There are two key ways for using Web services in conjunction with WebSphere Portal.

- Portlets running on the portal server can access a Web service to obtain information or perform a function provided by the Web service.
- Portals can publish portlets as remote portal Web services to make them available to other portals in a way that allows them to be more easily found and integrated.

This allows WebSphere Portal to leverage both data-oriented Web services and presentation-oriented Web services (see Figure 3).

Figure 3: Web services “on the glass”



Data-oriented Web services contain business logic but lack presentation logic. Therefore, every portal that runs a data-oriented Web service must implement its own presentation logic because it is not defined within the service boundary. Presentation-oriented Web services provide both application logic and presentation logic within the service boundary. Web Services for Remote Portlets (WSRP) is a Web services standard created by the OASIS standards body. The WSRP approach is best suited for dynamically integrating business applications. For example, to integrate an order status Web service using this approach, there is no need to develop the presentation logic. WebSphere Portal allows you to simply request the order status service to display as a portlet in the portal in a predefined location. To end-users, it appears that the portlet is running locally within their portals but in reality, the portlet resides in a remotely-running portlet container, and interaction occurs through the exchange of SOAP messages. Leveraging WSRP within an SOA provides a powerful combination whereby presentation-oriented portlet applications can be discovered and reused without engaging in additional development or deployment activities.

WSRP is an important cross-platform standard that will allow organizations to federate their portals and freely interoperate and share resources across the enterprise. For example, WSRP provides a strategy to integrate WebSphere Portal with Microsoft® Office SharePoint™ Portal Server 2003 and Windows® SharePoint Services Web sites. WSRP provides interoperability, portability and options for flexible portal deployments. As such, one of the stated goals of WSRP is to make the Internet a marketplace of visual Web services that are ready to be integrated into portals.

Assembly: Mainframe integration

WebSphere Portal also fully leverages IBM WebSphere Host Access Transformation Services (HATS) to quickly and easily extend legacy applications into a portal via reusable portlets. HATS makes 3270 and 5250 “green screen” applications available as portlets in order to convert host screens into a composite view inside a portal. This provides a significant improvement to the workflow and navigation of host applications without the need for access or modification to the host application source code. HATS also fully supports the ability to create WSRP portlets with WebSphere Portal as a presentation-oriented Web service.

Table 4. JSF and SDO defined

JavaServer Faces (JSF)

JSF is an official J2EE standard, JSR 127, which defines a standard for building user interfaces for server-side applications.

Service Data Objects (SDO)

SDO is an impending J2EE standard, JSR 235, and a new concept that simplifies data transfer between the persistence and presentation layers of an application by using disconnected data graphs.

Assembly: Web clipping

Another useful tool available in WebSphere Portal is the Web clipping capability that takes advantage of existing Web assets, allowing the identification and extraction or “clipping” of specific portions of Web content from existing sites and displaying them in portlets or mobile devices. This allows you to develop and deliver a single, consistent interface to existing Web resources all within your portal by reusing, rather than reimplementing, existing content and legacy Web sites.

Assembly: JavaServer Faces (JSF) and Service Data Objects (SDO)

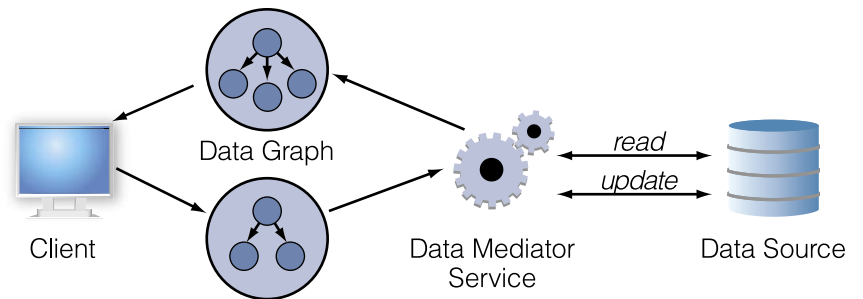
A popular term in the pre-Web days was “rapid application development,” or RAD for short. The main goal of RAD was to enable you to build powerful applications with a set of reusable components. Developers who used tools like Microsoft Visual Basic®, PowerBuilder or Delphi know that they were a major leap forward in application development productivity. For the first time, it was easy to develop complex user interfaces (UIs) and integrate them with data sources. However, these proprietary tool sets never contributed to providing a foundation for an SOA because they did not support cross-platform, open standards that are loosely bound and dynamically linked. JavaServer Faces (JSF) and Service Data Objects (SDO) were created to bring the same design concepts and the simplicity of RAD into J2EE application development while avoiding the shortcomings of the proprietary RAD tools due to the emergence of open standards that provide well-defined interfaces in a loosely coupled construct that supports an SOA.

JavaServer Faces is an official J2EE standard, JSR 127, which simplifies the development of user interfaces (UI) for server-side applications. JSF technology provides a reusable component architecture that governs event handling and the use of standard user interface components like input fields, buttons, hyperlinks and checkboxes. This separation of logic from the presentation layer allows developers of various skill levels to quickly build Web applications by assembling these reusable UI components on a portal page. JavaServer Faces has proven to be one of the most successful methods of reducing the complexity of developing J2EE applications and has seen incredible adoption by tool makers and by developers.

Service Data Objects (SDO) is a new concept that helps enable an SOA by simplifying data transfer between a wide variety of service and resource types (such as an ERP system and a relational database). SDO provides loose coupling via disconnected data graphs that allow data browsing and updates while the application is disconnected from the data source (see Figure 4). Using the disconnected data graphs architecture, a client retrieves a data graph from a data source, makes changes to the data graph, and can then apply the data graph changes back to the data source utilizing a Data Mediator Service (DMS). The data graphs can be dynamically created from just about any data source, including XML files, Enterprise Java Beans (EJBs), XML databases and relational databases – or from Web services, Java Connector Architecture resource adapters and Java Message Service packages.

The combination of JSF and SDO can have profound benefits for developers creating data-rich portal applications using J2EE. JSF provides a visual metaphor that speeds the creation of the presentation layer of an application by replacing tedious programming with drag and drop coding. SDO simplifies and abstracts how those portal applications use data sources by utilizing a single data graph architecture. Both of these assembly constructs embody the attributes of an SOA: loosely coupled and reusable, with well-defined standards-based interfaces.

Figure 4: Service Data Objects



Orchestration: Human task management

In order to successfully leverage business processes to their full potential, they must be surfaced effectively to all participants in a business. IBM WebSphere Portal enables distributed components, services and business processes to be presented to users of the portal as tasks through a variety of role-specific interaction points: the Web browser, mobile devices and rich-clients. This capability in WebSphere Portal allows internal and external portal users to participate and collaborate in all aspects of business processes in a portal environment without having to know details about the underlying processes.

Human tasks within these business processes (e.g. filling in forms, approving requests or fulfilling orders) need to be presented to the right people at the right time in a way that allows them to process their tasks timely and efficiently. The portal becomes the user interface for all user-facing tasks delivered via the human task portlet. For example, in Figure 5, the “My Tasks” portlet allows users to view, claim and process tasks in a workflow generated by the workflow engine all in one portlet. As each task is selected, a corresponding page is launched enabling action on the task. This allows the users of the portal to work in context from the task portlet and its integration with workflow because pages and portlets only pop up onto their screens when they are working on an activity that needs to be completed. This saves screen real estate and alleviates viewing unnecessary portlets and pages.

Figure 5: Orchestrating business processes into the portal

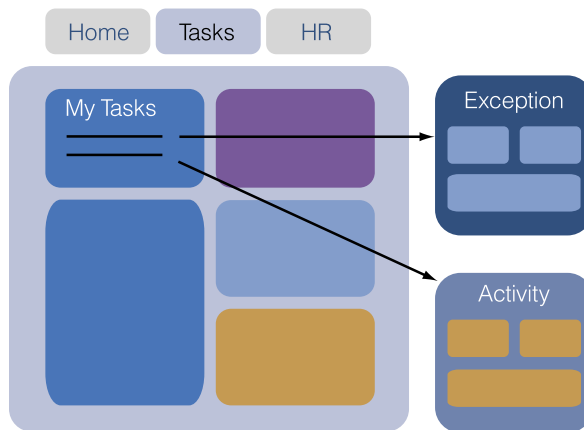


Table 5. Understanding BPEL4WS

Business Process Execution Language for Web Services (BPEL4WS) allows the definition of both business processes that make use of Web services and business processes that externalize their functionality as Web services. This includes specifying how a business process makes use of Web services to achieve its goal, as well as specifying Web services that are provided by a business process.

In April 2003, IBM teamed with Microsoft, BEA Systems, SAP AG and Siebel Systems to submit BPEL4WS to OASIS for standardization via the Web Services BPEL Technical Committee.

With the use of process integration in WebSphere Portal it is possible to implement a variety of scenarios, including:

- *Production Workflows: highly automated, with human involvement restricted to exception handling in case of a failure.*
- *Collaborative Workflows: human tasks are a key part of the workflow and enable collaboration in context with colleagues to create, share and modify forms, documents, communications and other content.*

The ability to orchestrate workflow into the portal is provided by the WebSphere Portal foundation architecture. This architecture enables WebSphere Business Integration Server Foundation to deliver integrated process choreography and J2EE workflow capabilities. The WebSphere Business Integration Server Foundation utilizes Business Process Execution Language for Web Services (BPEL4WS) to specify the business process behavior (workflow) and describe how individual Web services can be orchestrated to create composite services. BPEL for process choreography is a key component in the development and deployment of process-driven portals because of its strong ability to integrate heterogeneous IT environments by transforming applications into services.

Orchestration: Cooperative portlets

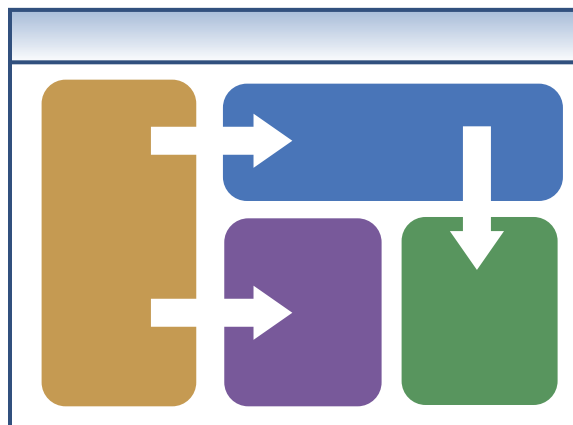
Cooperative portlets provide a method of orchestrating composite applications and is one of the more powerful and widely used features of WebSphere Portal. The term “cooperative portlet” refers to the capability of portlets on a portal page to interact with each other by interchanging data and messages. One or more cooperative portlets on a portal page can automatically react to changes from a source portlet triggered by an action or event in the source portlet. Portlets that are targets of the event can react so that users are not required to make repetitive changes or actions in other portlets on the page. This provides a coordinated and consistent behavior between the portlets on the page while the user interaction with the portlet provides the user context and situation that is the basis for dynamically adapting the portal and application for the user.

The intercommunication between portlets simplifies what would otherwise be a complicated manual process potentially involving multiple graphical user-interface clients for different applications, logging into the systems, copying and pasting information or retyping it after writing it down.

Figure 6 represents cooperative portlet behavior with arrows illustrating where data and messages from one portlet is used to query other portlets, whereby additional information is retrieved and presented to the user.

This cooperation between source and target portlets is facilitated by a WebSphere Portal runtime entity called the property broker. Portlets on a page can cooperate in this way even if they were developed independently, without the programmer's awareness of the existence of the other cooperative portlets. Cooperative portlets provide a key capability to develop a more loosely coupled, reusable portlet strategy when developing and deploying composite applications.

Figure 6: Orchestrating business processes into the portal

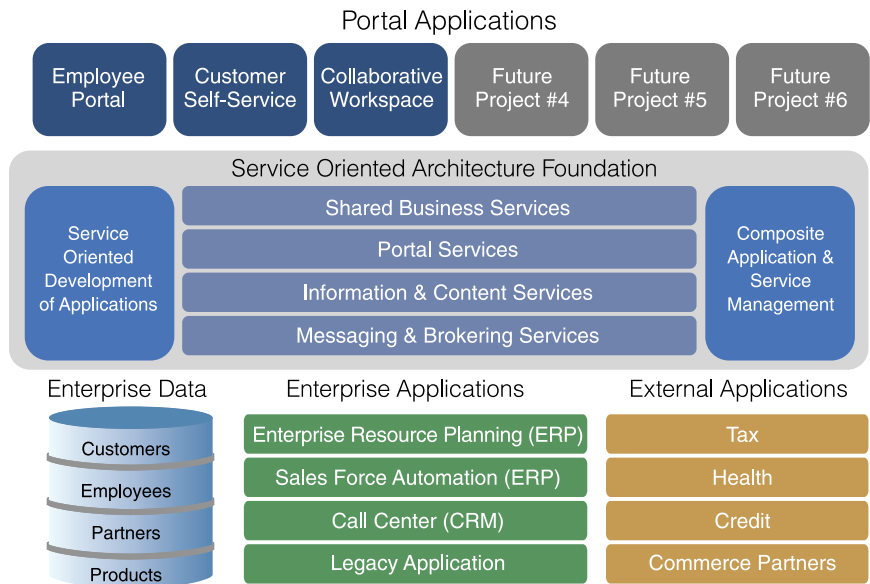


Putting it all together

The real value of an SOA can be seen over time. For example, the value attained by a leading auto manufacturer is not tied into delivering a single model to the market; it's the ability to deliver on multiple models by leveraging high levels of parts reuse and process optimization across models and brands. This level of parts reuse and flexibility reduces engineering time, saves money, improves quality and serves global ambitions. So, too, is the value derived in how IT deploys composite applications. While it is always important to derive value and return on investment from your first portal project, the true value is when you deploy the next two, three, four projects and so on.

Figure 7 shows an ideal end state for an IT organization that has deployed an SOA. It is in the SOA foundation where we begin to see the advantages of an SOA: reduced development times, cost savings, faster time-to-delivery on projects and an IT organization that is responsive to change and strategic to the business.

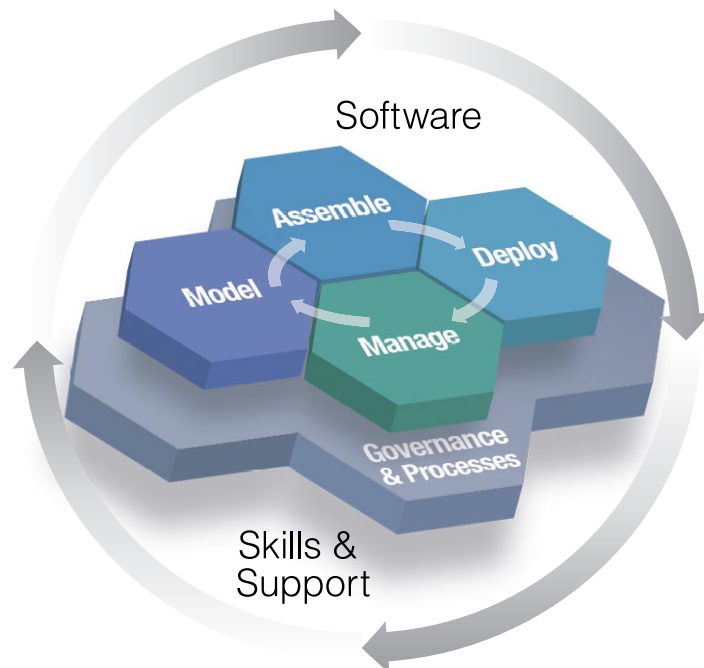
Figure 7: An ideal SOA end state



IBM SOA foundation

WebSphere Portal is just one part of an overall IBM SOA foundation. Although this white paper focuses on portals as an on-ramp to a service oriented architecture, IBM's SOA foundation supports a modular approach to a complete SOA lifecycle as an integrated, open set of software, best practices and patterns that provides you with what you need to get started with SOA. IBM's SOA foundation provides comprehensive support for the SOA lifecycle (model, assemble, deploy and manage) through an integrated set of tools and runtime components that allow you to leverage skills and investments across the common runtime, tooling and management infrastructure (see Figure 8).

Figure 8: The IBM SOA foundation



Conclusion

IBM WebSphere Portal provides an on-ramp to a service oriented architecture for organizations seeking a highly pragmatic approach to their IT infrastructure and the business solutions that run on them. As your organization embarks on a new portal or Web application project, WebSphere Portal can provide the tools and technology for the assembly and orchestration of composite applications. And as you continue to expand your adoption of SOA, WebSphere Portal provides a tactical entry point and a step-wise approach to deliver a unified user experience for services and processes within and beyond your enterprise.

In the era of On Demand Business, there is an acute awareness of the need to deliver tangible progress and quantifiable results. IBM is helping customers today build their strategic business design and business case for successful portal projects. WebSphere Portal represents an excellent opportunity to enable organizations to improve access and reduce the complexity of today's computing environments—in their own time, at their own pace, aligned with their organization's short- and long-term business goals.

IBM WebSphere Portal delivers value to your organization by providing a portal platform for:

- *Composite applications—reusable assets to reduce engineering costs and portal deployment times.*
- *Productivity—dynamic, role-based composite views delivered in context.*
- *Enterprise agility—engineered to support open standards and deliver process-driven portals on an SOA.*

For more information

To learn more about the IBM WebSphere Portal family of offerings and services, visit ibm.com/websphere/portal and ibm.com/soa



© Copyright IBM Corporation 2005
IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
10-05
All Rights Reserved

IBM, the IBM logo, Rational and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries or both.

Microsoft, SharePoint, Visual Basic and Windows are a trademarks of Microsoft Corporation in the United States, other countries or both.

Other company, product and service names may be trademarks or service marks of others.

References in this publication to IBM products and services do not imply that IBM intends to make them available in all countries in which IBM operates.

- 1 Morgan Stanley Equity Research: Enterprise Technology. "Morgan Stanley CIO Survey Series: Release 5.1," D. Togut, L. Kennedy. February 9, 2005.
- 2 Forrester Research, Inc. "Portal Projects In Search Of A Purpose," L. Ramos. April 21, 2004.
- 3 Gartner, Inc. "Portals Provide a Fast Track to SOA," G. Phifer. July 15, 2005.
- 4 Gartner, Inc. "Positions 2005: Service-Oriented Architecture Adds Flexibility to Business Processes," S. Hayward. February 16, 2005.
- 5 Gartner, Inc. "Magic Quadrant for Horizontal Portal Products, 2005," G. Phifer, R. Valdes, D. Gootzit, K.S. Underwood, L.F. Wurster. May 18, 2005.
- 6 IDC. "The Enterprise Workplace: How It Will Change the Way We Work," B. McDonough, R.P. Mahowald, J. Duhl, A. Crawford, M. Levitt, S. Feldman, E. Quinn. February 1, 2005.
- 7 Giga Research, a wholly owned subsidiary of Forrester Research, Inc. "Packaged Composite Applications Emerge—Slowly," E. Kinikin, L. Ramos. January 22, 2004.