WebSphere Process Server: IBM's new foundation for SOA

Wolfgang Kulhanek
Carol Serna

September 12, 2005

WebSphere Process Server V6 is a next-generation business process server that supports all styles of integration based on SOA and open standards, and automates business processes that span people, workflows, applications, systems, platforms, and architectures. This article introduces the major WebSphere Process Server capabilities, features, and solutions, and provides references so you can learn more.

What is WebSphere Process Server?

Service Oriented Architecture provides you with the ability to develop and modify integration applications dynamically. It also lets you integrate existing applications with newer applications, so that they work together transparently.

IBM® WebSphere® Process Server V6 is a comprehensive Service Oriented Architecture integration platform, which is based on WebSphere Application Server V6.

You can use WebSphere Process Server to develop and execute standards-based, component-based business integration applications in a Service Oriented Architecture. Because it is based on J2EE 1.4 infrastructure and platform services provided by WebSphere Application Server V6, WebSphere Process Server includes capabilities such as business process automation.

You can use WebSphere Integration Developer to develop applications that run on WebSphere Process Server V6. WebSphere Integration Developer, based on the Eclipse 3.0 platform, delivers role-based development for integration projects. You can use WebSphere Integration Developer and WebSphere Business Modeler V6 together with IBM Rational Application Developer V6 or IBM Rational Software Architect V6 as an integration development platform. J2EE developers, business analysts, or integration developers can use the tooling perspective based on their unique roles, so that each developer can focus on the editors and tools needed for these roles thereby maximizing productivity.
Why would you use Service Component Architecture and Business Objects with WebSphere Process Server?

What is Service Component Architecture (SCA)?

WebSphere Process Server supports two capabilities required for an efficient Service Oriented Architecture: a universal invocation model, implemented as a Service Component Architecture (SCA), and a universal data representation implemented, as Business Objects (BOs).

Conceptually, SCA describes all integration artifacts as service components with well defined interfaces. SCA also introduces the concept of a module, which groups together service components and provides further specification and encapsulation of services. Integration developers can use the Assembly Editor in WebSphere Integration Developer to group service components into modules, and to specify which service interfaces you want exposed by the module to outside consumers.

You can use Services that include imported components, such as Java Beans or Web Services, along with service components that WebSphere Process Server provides. You can then connect the modules together to form complete integration solutions. SCA concepts let you encapsulate integration logic within modules. This means that you can make a change to a service component within a module without affecting any of the other modules in the overall solution as long as the interface of the module you changed stays the same. This concept applies throughout WebSphere Process Server. All integration artifacts in WebSphere Process Server -- processes, business rules, human tasks, and so forth -- are represented as SCA service components.

This creates a very flexible environment that you can use, for example, to replace an approval module that contains a human task with a module containing a business rule. As long as the interface to the module is identical, you can deploy the updated module, and it will be picked up automatically by all consuming modules without changes.

You can use SCA to invoke service components through both synchronous and asynchronous programming styles. This set of options lets you assemble the modules into overall solutions so that asynchronous channels between service components and modules can increase the overall throughput and flexibility of the system.

What are Business Objects (BO)?

You can use Business Objects, extensions of Service Data Objects (SDO), to provide an abstraction layer for data access. While SDOs provide a universal means of describing disparate data, such as JDBC ResultSet, XML Schema described data, and so forth, Business Objects include some extensions that are very important for integration solutions; you can use them to further describe the data that is being exchanged between SCA services. This includes metadata-like change history or information on the context of the data, such as update, create, delete, and so on.

While the concepts of Service Component Architecture and Business Objects are new, the implementation is standards based. For example, interfaces to any SCA service component are
described through Web Service Definition Language (WSDL) or Java interfaces and Business Objects are described through XML Schema (XSD).

What are the new features?

WebSphere Application Server V6

As the foundation for WebSphere Process Server capabilities, WebSphere Application Server V6 is a native Java Messaging Service provider that includes full interoperability with existing WebSphere MQ-based networks and support for Web services. Java™ 2 Connector Architecture 1.5 renders key connection services for WebSphere Process Server.

Figure 1. WebSphere Process Server platform

Service Oriented Architecture core

Service Component Architecture, Business Objects, and the Common Event Infrastructure (CEI) are part of the SOA core. In addition, to allow for the invocation of native SCA service components, various SCA bindings allow for interaction with external services such as Enterprise Java Beans, Web services, JMS Messages or legacy applications through WebSphere Adapters. You can use CEI to capture events you can use to monitor applications, for example, in IBM WebSphere Business Monitor or IBM Tivoli products. WebSphere Process Server builds on and leverages CEI to emit a specific set of events for each SCA service component.

Supporting services

On top of the runtime infrastructure and the SOA core, WebSphere Process Server offers a variety of service components. Supporting services are components that are needed in any integration solution including data transformation and synchronization services.

- **Interface maps**: It is possible for interfaces of existing components to match semantically but not syntactically (for example, updateCustomer versus updateCustomerInDB2). This is especially true for components that already exist and services that need to be accessed. Interface maps let you invoke these components by translating these calls. Additionally, you can use Business Objects to translate the actual Business Object parameters of a service invocation.
• **Business Object maps**: You use a Business Object map to translate one type of Business Object into another type of Business Object. You can use these maps in a variety of ways, for example, in an interface map to convert one type of parameter data into another.

• **Relationships**: You may want to access the same data within business integration scenarios, for example, customer records, or in various backend systems, such as an ERP system and a CRM system. A common problem for keeping Business Objects in sync is that different backend systems use different keys to represent the same objects. You can use the relationship service in WebSphere Process Server to establish relationship instances between objects in these disparate backend systems. These relationships are typically accessed from a Business Object map when translating one Business Object Format into another.

• **Selector**: You can use a selector component for dynamic selection and invocation of different services, which all share the same interface. For example a customer support process could use different human tasks implementations during holidays than during regular working days. WebSphere Process Server offers a Web-based interface to enable dynamic updates to the selection criteria and target services, which means that a module that has been deployed at a later time can still be called by this selector component enabling dynamic changes to the integration solution.

• **Java**: You can use Java components to invoke Java code.

### Service components

WebSphere Process Server provides four service components:

• **Business processes**: The business process component in WebSphere Process Server implements a Web Services Business Process Execution Language (WS-BPEL) compliant process engine. You can develop and deploy business processes that support long and short running business processes and a compensation model within a scalable infrastructure. You can create WS-BPEL models in WebSphere Integration Developer or import from a business model that you can create in WebSphere Business Modeler.

• **Human Tasks**: Human Tasks are standalone components in WebSphere Process Server that you can use to assign work to employees or to invoke any other service. Additionally, the Human Task Manager supports the ad hoc creation and tracking of tasks. You can use existing LDAP directories (as well as operating system repositories and the WebSphere user registry) to access staff information. WebSphere Process Server also supports multi-level escalation for human tasks including e-mail notification and priority aging. WebSphere Process Server includes an extensible Web Client that you can use to work with tasks or processes. This Web Client is based on a set of reusable Java Server Faces (JSF) components that you can use to create custom clients or embed human task functionality into other Web applications.

• **Business State Machines**: Business State Machines provide another way of modeling a business process. This lets you represent your company's business processes based on states and events, which sometimes are easier to model than a graph-oriented business process model. One example would be an ordering process where you can modify or cancel the order at any time during the order process until the order is actually fulfilled.

• **Business Rules**: Business Rules are a means of implementing and enforcing business policy through externalization of business function. This enables dynamic changes of a business
process for a more responsive businesses environment. Business rule authoring is supported with Eclipse-based desktop tooling. Business Analysts can use the Web-based runtime tooling included in WebSphere Process Server to update business rules as business needs dictate without affecting other services.

You can configure and administer all of WebSphere Process Server's features using extensions to the WebSphere Application Server administration console and configuration capabilities. This provides one place to administer the entire application stack.

**What can you do with WebSphere Process Server?**

**Transaction, security, clustering and workload management:** WebSphere Process Server solutions leverage WebSphere Application Server capabilities, thus providing you with a scalable and reliable business integration environment you can use for transaction, security, clustering and workload management.

**Full ACID transaction support:** WebSphere Process Server offers full ACID transaction support for business processes, both for short-running (one transaction end to end) and long-running processes (multiple transactions). You can modify transaction boundaries in the tooling to group multiple steps in a business process into one transaction. Additionally, it supports flexible compensation of business processes as defined in the WS-BPEL specification.

**Recovery Manager as well as a Recovery Console:** WebSphere Process Server includes a Recovery Manager as well as a Recovery Console. In case of a failure in the middle of executing a business integration application, the server will detect this failure and allow you, the administrator to manage the failed applications from the Recovery Console.

**Encapsulation of Business Functions:** The unique architecture in WebSphere Process Server allows the encapsulation of business functions into separate modules that you can then update independently of each other. For example, you can use an approval module that contains a human task for the actual approval and later later replace it with another approval module containing a business rule. This change is completely transparent to the consumer of this module. Additionally, the concept of encapsulation ensures that data and interface definitions are encapsulated where they are used. For example, you can hide the details of how a customer is represented in a backend system inside of a module, while the module itself exposes a generic interface with a generic business object as data. This canonical data representation also enables a high degree of reuse in any given integration application.

**How does it work with existing systems and new applications?**

WebSphere Process Server offers many options you can use to integrate your company's existing systems with new integration applications.

**WebSphere Application Server messaging resources:** WebSphere Process Server uses the messaging resources in the WebSphere Application Server to tie into an existing WebSphere MQ network. You can configure the native Java JMS provider that comes with WebSphere Application Server.
Server V6 to connect into an existing WebSphere MQ Queue Manager, so you can create an extension to the WebSphere MQ network.

**Native Web services Interoperability**: WebSphere Process Server contains full support for Web services including support for SOAP over HTTP or SOAP over JMS. WebSphere Process Server also includes support for import and export for WebSphere Process Server components, JMS, and Enterprise Java Session Beans.

**WebSphere Adapters**: WebSphere Adapters include both application-specific adapters, for example, for Siebel, SAP, or PeopleSoft as well as technology adapters, for example, for relational databases or flat files.
Related topics

- Read the WebSphere Process Server announcement.
- Visit the WebSphere Integration Developer product site to learn more about WebSphere Integration Developer.
- The article IBM WebSphere Developer Technical Journal: Introducing the WebSphere Integration Reference Architecture (developerWorks WebSphere, August 2005) provides an overview of the WebSphere Information Integration architecture.
- Learn more about WebSphere Business Integration, visit the IBM Business Integration product site. You'll find technical documentation, how-to articles, education, downloads, product information, and more.
- Get support for products on the IBM Software Support Lifecycle site.

© Copyright IBM Corporation 2005
Trademarks
(www.ibm.com/developerworks.ibm/trademarks/)