Orchestrating Web Services: The Case for a BPEL Server

An Oracle White Paper
June 2004

ORACLE

Orchestrating Web Services: The Case for a BPEL Server

Orchestrating Web Services: The Case for a BPEL Server

EXECUTIVE OVERVIEW

Web services and the Business Process Execution Language for Web Services (BPEL) are changing the economics of integration. This white paper describes this new approach for implementing service-oriented architectures and how the Oracle BPEL Process Manager provides a mature and reliable implementation of these standards in an integration platform available today.

BUSINESS PROCESS INTEGRATION GOES MAINSTREAM

An enterprise's business processes provide the most important point of competitive differentiation. The definition and flawless execution of processes enable an organization to provide more competitive products or services, reduce costs, improve customer service, and react quickly to changing market conditions.

Traditional integration solutions are proprietary, expensive, and have only been able to address the high end of the integration market. While some standards such as J2EE Connector Architecture (JCA), Java Messaging Service (JMS), and RosettaNet have been developed and adopted to solve different aspects of this problem, a standard for comprehensive process orchestration has been lacking. Until now.

Implementing an industry standard for orchestrating business processes and Web services will not only speed the implementation and deployment of new integration projects, but will also reduce the overall cost of management, modification, extension, and redeployment of existing processes. In addition to tactical time and cost savings, this provides a strategic advantage: superior responsiveness to changing market conditions.

WEB SERVICES AND BPEL

Industry Standards for Interoperability

The term "Web services" refers to a set of interoperability standards (WSDL, XML and XML Schema, SOAP, JMS, JCA, etc.) that simplify integration with heterogeneous systems throughout the extended enterprise. The same way standards like SQL revolutionized access to structured data and HTTP and

HTML standardized the way people access content and applications, Web services have the potential to transform the internet into a true distributed computing platform and allow heterogeneous systems to cooperate simply and reliably.

Business Process Execution Language for Web Services (BPEL) provides enterprises with an industry standard for business process orchestration and execution. From a technical perspective, BPEL offers a standard language for defining how to: send XML messages to remote services, manipulate XML data structures, receive XML messages asynchronously from remote services, manage events and exceptions, define parallel sequences of execution and undo parts of processes when exceptions occur. These are the constructs needed to compose a set of services into collaborative and transactional business processes. BPEL is based on XML Schema, SOAP and WSDL.

Unlike process standards that have been proposed in the past, Business Process Execution Language for Web Services, driven by the OASIS standards body, has achieved the critical support and endorsement from the industry's leading vendors. While earlier fragmented efforts fell short in developing a single, comprehensive standard that meets the needs of customers, BPEL is a comprehensive standard that satisfies real-world requirements and has the support of major infrastructure and application vendors such as Oracle, Microsoft, IBM, SAP, and Siebel.

Oracle BPEL Process Manager

Oracle BPEL Process Manager enables organizations to model and deploy business processes based on the Business Process Execution Language for Web Services (BPEL) standard. The cornerstone of process orchestration and execution within a Service-Oriented Architecture, the BPEL standard provides an enterprise blueprint for reducing the cost and complexity of integration projects – while increasing their strategic value. Available now, Oracle BPEL Process Manager delivers:

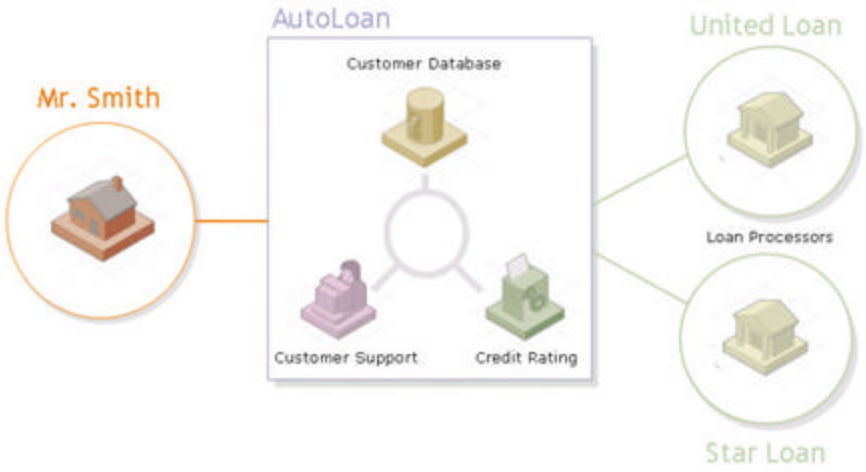
- The first native BPEL engine, ensuring 100 percent process portability
- A production-level BPEL process management solution that customers can use today
- A deployment-proven solution for implementing the BPEL standard

Making Web Services Work

Business and people that work together need their applications to work together. IT applications today typically must be built upon, and integrated with, a wide variety of existing systems, services and business processes. This has turned most J2EE or Microsoft .NET developers into "integration developers" and made multi-language and protocol interoperability a mainstream requirement.

Let's take the example of a loan procurement application implemented by a company named "AutoLoan." Through its online portal, AutoLoan offers loans to consumers who apply for financing of used car purchases. The AutoLoan application will leverage several trading partners who provide the actual financing as well as existing information systems and legacy applications for customer information, credit ratings, etc. In addition, AutoLoan needs the system to support interactions with people, such as customer service representatives.

The standards emerging around Web service orchestration such as SOAP, WSDL, XML Schema and BPEL enable AutoLoan to address their integration and business process management requirements in a vendor-independent fashion. And beyond just leveraging these standards, AutoLoan wants to build their system with a loosely coupled, service-oriented architecture so that they will be able to get the efficiency of highly integrated systems while minimizing the cost, time and resources required to build and maintain them.



What makes this example particularly interesting is that it requires both integration of existing functionality and new application development. Hence, implementing the system requires AutoLoan to integrate disparate developer skills, methodologies and infrastructures into a maintainable application. It is these needs that are driving AutoLoan – and the general market – toward Web services as a standard service interface and BPEL for process orchestration. Note, by the way, that the AutoLoan example incorporates both internet/B2B style integration and A2A/intranet integration. Many uses of Web services and BPEL today are for purely intranet-based integration and the requirements, standards, and products described here fit this model equally well.

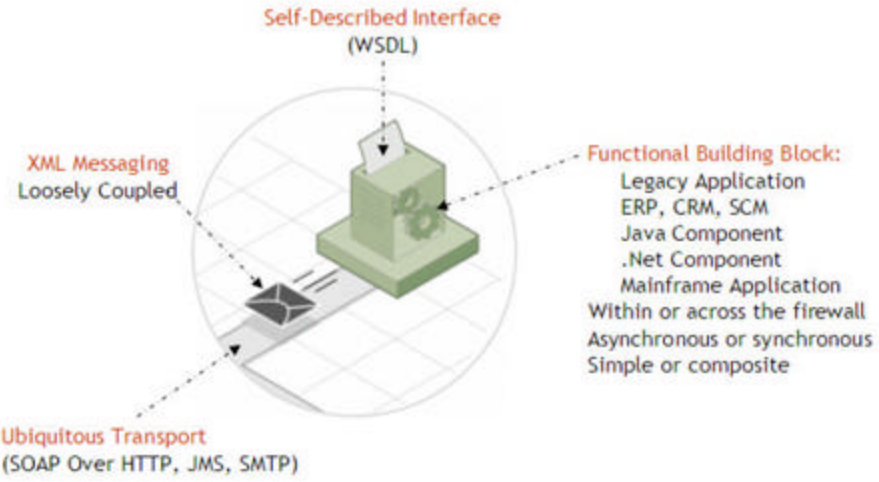
Making Web services work is a two-step process: first you publish them and then you orchestrate them. Publishing means making the services available through

some standard interface/API. Orchestration means assembling and coordinating those services into a manageable business application.

Publishing Web Services

Publishing a service takes a piece of functionality that already exists, such as within an ERP system, a legacy application, a Java component or a .Net component, and making it available over the network so that it can be easily integrated into applications. The standards surrounding Web services include a standard way of describing the interface to a service (WSDL), a data model (XML and XML Schema) and are flexible enough to support nearly any protocol.

You can think of a published Web service as a building block that receives an XML request message, does some processing and generates a set of XML response messages. The details of the service interface are defined using a WSDL file contract. The actual transport and exchanges of messages can be implemented using ubiquitous protocols such as HTTP, JMS, JCA, Java and SMTP.



In our AutoLoan example, American Loan and Star Loan publish their loan financing capabilities as Web services that accept loan applications, return loan offers, and issue loan policies upon acceptance of an offer. American Loan uses the open source Axis toolkit from Apache and Star Loan uses Microsoft .Net for publishing their respective services.

Keep in mind that the definition of Web services, as used here, includes non-SOAP/XML building blocks to accommodate performance requirements and direct integration with existing messaging infrastructures and applications. In our AutoLoan example, the credit rating functionality is a CICS transaction, which is already published on an MQSeries message bus. In this case, we can use the Java Messaging Service (JMS) API to access and consume that published functionality

reliably and asynchronously. By using a flexible binding framework like WSIF from Apache, AutoLoan can use a 100 percent standard BPEL process to orchestrate both the Web services/SOAP operations and the JMS messages.

Now that we have seen what publishing Web services entails, we can move on to the requirements associated with orchestrating Web services and how they are addressed by the BPEL standard and the Oracle BPEL Process Manager.

ORCHESTRATING WEB SERVICES WITH BPEL

Asynchronous Messaging

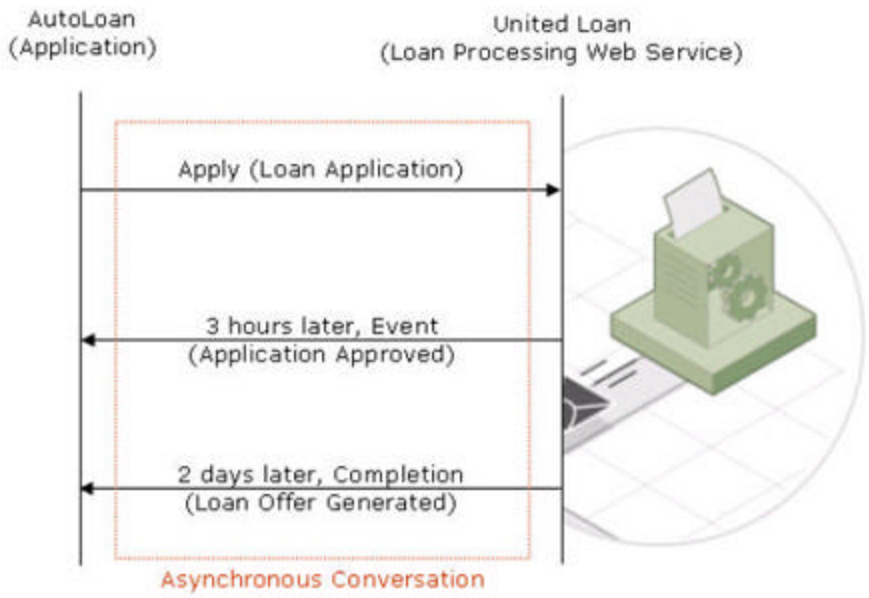
Part of the challenge of building the loan procurement application is to assemble the presentation logic, local business logic, and the published services into a manageable service-oriented application.

In order to achieve reliability, scalability, and adaptability, interactions with Web services will have to support both synchronous and asynchronous messaging styles. In the AutoLoan example, the application interacts with the credit rating system through JMS and with the loan processors through SMTP and HTTP. In addition, an EJB might be used to interface to a custom J2EE loan servicing application and synchronous Web services as interfaces to. NET-based systems.

Generally speaking, some of the services in an enterprise will already be implemented and may define only synchronous interfaces. Other services may exist as asynchronous messages or be implemented as part of the development of a new application using asynchronous Web services protocols. In all of these cases, a process or service which will integrate with other services, within or outside of an enterprise, needs to be able to gracefully handle situations where services may become unavailable at unpredictable times. This requires that a process flow use asynchronous messaging styles and effective exception management if it is to avoid the brittleness of a tightly coupled architecture.

The Application is the Orchestration™

The emergence of asynchronous messages and Web services as building blocks for applications introduces new challenges. In particular, the synchronous request-reply programming model is giving way to a conversational model based on asynchronous interactions across loosely coupled Web services.



Here, we define orchestration logic as the business logic that sequences, coordinates and manages conversations among Web services. Such orchestration logic can be as simple as a single two-way conversation or as complex as a non-linear, multi-step business transaction with exception handling and compensation logic.

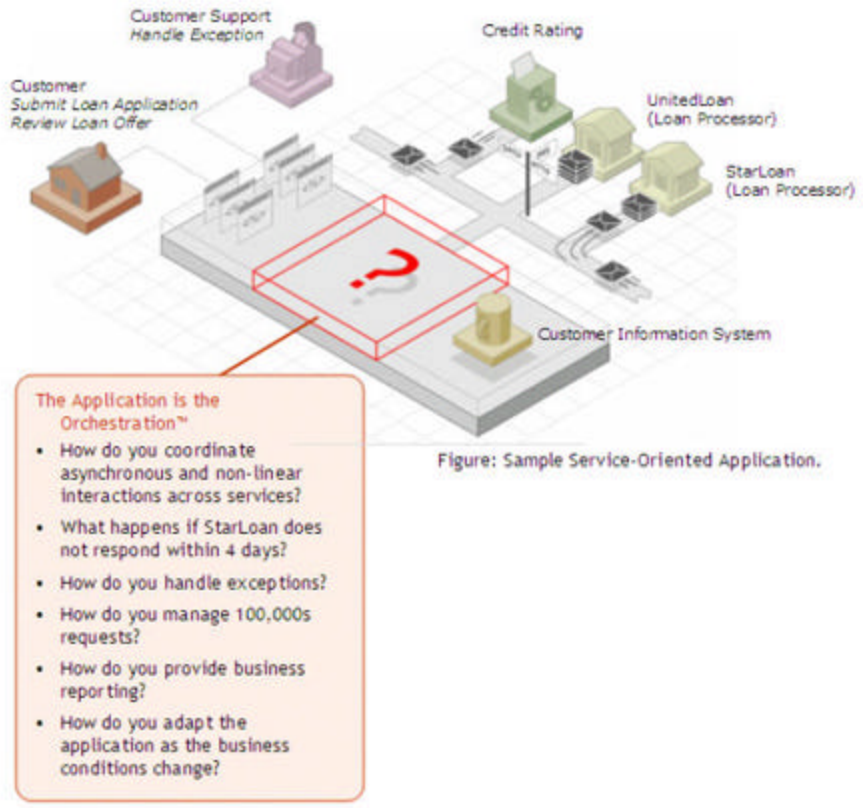


Figure: Sample Service-Oriented Application.

In the AutoLoan example, the orchestration logic includes extracting the customer profile from an existing database, requesting the credit rating from an internal service and then asking the two loan processors in parallel to process the loan application.

A Common Set of Requirements

Implementing and managing the orchestration logic of service-oriented applications entails a consistent set of infrastructure-level requirements. In this section, we will review those requirements in the context of the sample AutoLoan application and see how the BPEL standard and the Oracle BPEL Process Manager address them.

Req. #1: **Open standards (Java/J2EE, JMS, XML, SOAP, WSDL)**

How do you leverage your existing investment and knowledge in Java and J2EE? How is each conversation marshaled into SOAP, XML, and JMS messages so that the other side of the communication is independent of your implementation? How do you interoperate with systems and services implemented with many different languages and technologies?

The Web services standards today offer unprecedented interoperability and a fraction of the cost of proprietary EAI platforms. The Oracle BPEL Process

Manager is built from the ground up with native support for these standards, enabling both interoperability and portability for applications and business processes.

Req. #2: **State and context management**

How do you coordinate, store and manage the state of each conversation while still utilizing asynchronous messaging? How do you correlate each response to a set of actions? The Oracle BPEL Process Manager solves these problems at the infrastructure level, with seamless support for the "dehydration" of long-running processes and correlation of asynchronous messages.

Req. #3: **Loosely-Coupled Services**

How do you model each conversation so that the overall application (or service implementation) can be easily adapted as the business conditions change? How easily can you add a new loan processor to the loan procurement application? BPEL and the Web services standards enable a loosely coupled, coarse-grained design pattern that lends itself to efficient implementation of services-oriented architectures (SOA).

Req. #4: **Parallel Processing**

How do you design and coordinate parallel conversations? How do you make sure that the loan application is simultaneously submitted to United Loan and Star Loan? How do you implement sophisticated join patterns ("Cancel the conversation with United Loan if the user selects Star Loan")? BPEL is the most mature process/workflow language standard to date and draws upon the rich history of its predecessor languages (specifically XLANG from Microsoft and WSFL from IBM) to provide rich process flow capabilities.

Req. #5: **Exception Management**

System and business level exceptions tremendously increase the variability and the complexity of the orchestration logic. What happens if American Loan refuses to process the application because the application is invalid? BPEL has robust support for exception handling and the Oracle BPEL Process Manager includes both design-time and run-time capabilities for managing, monitoring and administering faults and exceptions.

Req. #6: **Events/Notifications**

Conversations can span a long period of time and include multiple responses. How do you handle notifications such as the one generated by American Loan when the application is approved? How do you specify and handle time-outs when services do not respond in a timely fashion? Again, BPEL includes support for these requirements, allowing the implementation of highly complex, non-deterministic flows.

Req. #7: **Open Nested Transactions**

How do you combine multiple non-linear conversations into a business transaction? How do you track the history of the conversations so that they can be compensated if necessary? BPEL includes a compensation mechanism for the implementation of long-running transactions – even when the component services do not use a common transaction protocol.

Req. #8: **Scalability and Reliability**

What happens when you need to upgrade a server but cannot afford to stop running your integration applications? What happens when the loan procurement application becomes successful and needs to scale to handle hundreds of thousands transactions per day? The Oracle BPEL Process Manager can be easily clustered for both fault-tolerance/failover and to handle increasing transaction volume. A single BPEL process instance can be created on one server, automatically relocate to another server if a server failure occurs (including during the execution of a process), and then complete on a third server based on distribution of load.

Req. #9: **Management, Administration and Business Visibility**

What happens if a customer support rep needs to cancel a submitted loan application request? How do you provide administrators, executives and customer service reps with aggregate process statistics and instance level business visibility? The Oracle BPEL Process Manager includes unparalleled management and administration capabilities, supporting the easy development of custom dashboards and views on top of process statistics.

Req. #10: **Version Control**

Imagine that you have 10,000 active loan application requests and that you need to update the orchestration logic to reflect some new policy. How do you gracefully phase in and out multiple versions of your orchestration logic? The Oracle BPEL Process Manager supports "side-by-side versioning" of processes so that flow logic can be upgraded for new instances while existing instances execute against the flow logic that existed at their time of creation.

Req. #11: **Audit Trailing**

How do you trace the history of all the conversations related to a specific loan request? Can you provide non-repudiation and view the messages exchanged with specific services after the fact? The Oracle BPEL Process Manager automatically maintains audit trail information, supporting both a graphical and textual representation of process status and history. In addition, the audit trail can be easily customized to ensure that it is meaningful and complete based on the business semantics of the process.

Req. #12: **Support for Existing Infrastructure**

Do you have existing application server and messaging infrastructure and expertise that you want to maintain? Would you like to be able to run BPEL processes on different platforms on a project-by-project basis? The Oracle BPEL Process Manager runs on top of all the major application servers, including Oracle Application Server, WebSphere, WebLogic and JBoss. Production deployment platforms range from Windows and Linux to Solaris and IBM Z/OS. The database used as a dehydration store can be Oracle, SQL Server, DB2 or others.

Mapping out the common infrastructure requirements for service-oriented applications resembles the process of identifying the common requirements for building self-service Web applications. The latter class of applications started with implementation of CGI programs where developers repeatedly encoded infrastructure requirements for each application (e.g. session management, database connection pooling, multi-threading, etc). Identifying these common infrastructure requirements resulted in the software infrastructure known as an Application Server.

In the same way, the BPEL standard, supported by a "BPEL server" as infrastructure software, enables integration developers to work at a higher level of abstraction when implementing services-oriented applications. A BPEL server provides portability and a broad developer network for the problem of process integration, especially as other vendors such as IBM, Microsoft, and SAP have committed to ship BPEL servers and are expected to do so in commercial releases by 2005.

CONCLUSION

Businesses and people that work together need their applications and services to work together. This is driving the industry move to Web services and BPEL, which promise significant benefits in terms of adaptability, ease-of-integration, portability, and interoperability.

Making Web services work is a two-step process involving publishing services and orchestrating them. Publishing means making the Web services available through a supported interface/protocol but does not require that all existing systems be "wrapped" with a new XML/SOAP Web service layer. Orchestration means assembling and coordinating these services into a manageable business application.

However, implementing, executing and managing orchestration logic is complex and entails a consistent set of infrastructure-level requirements. This created the need for the BPEL standard and software infrastructure to support the definition and implementation of integration processes. This software infrastructure should be built from the ground up around Web services and BPEL and support the

native design and execution of BPEL processes – if portability and vendor independence are goals for an enterprise.

The Oracle BPEL Process Manager provides exactly this infrastructure – a 100 percent, native BPEL engine that coexists happily with existing middleware technologies and platforms. For more information and to download a trial version of the Oracle BPEL Process Manager, visit http://otn.oracle.com/bpel.

Orchestrating Web Services:
The Case for a BPEL Server
June 2004
Author: Dave Shaffer and Brian Dayton

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com