Conductor: An Enabler for Web Services-based Business Collaboration

David Burdett, Qiming Chen, Meichun Hsu
Commerce One Inc.
19191 Vallco Parkway, Cupertino, CA 95014

Abstract

To enable intra- and inter-enterprise application integration, enterprises have invested heavily in EAI (Enterprise Application Integration) and B2B (Business to Business) technologies. Recent momentum in Web Services based on the SOAP, WSDL and UDDI specifications promises a standardized connectivity at a lower cost that would dramatically transform the business connectivity paradigm. In this article, we introduce Commerce One Conductor, an enabler for business collaboration based on robust web services, and provide an overview of 2 specific design points: (a) Conductor's web service extensions that address security, guaranteed delivery, routing, choreography, and document and service version interoperation, and (b) a Conversation Manager-based mechanism to extend the business process engine to provide collaborative web services.

1 Introduction

E-business is moving toward a paradigm in which enterprises interact with each other through exchanging XML documents based on well-defined protocols such as SOAP, WSDL, UDDI, and BPSS. These emerging Web Service standards enable businesses to interoperate in a dynamic and loosely coupled way. Based on Forrester Research [5], firms will expand their use of partners, private, and public Web Services in the next five years, and Global 3,500 firms are expected to spend about USD 7m on a private hub to enhance visibility and improve supplier collaboration.

Commerce One has focused on eliminating friction within enterprises and between trading partners through information exchange via a Marketplace platform called MarketSite. Taking the lessons learned in connecting large corporations together to trade, Commerce One is transforming MarketSite into a business collaboration platform, called Commerce One Conductor, based on emerging web service standards and the service oriented application architecture. It abstracts both human activities and business applications into "services": human activities are known as Portal Services, while business applications expose Web Services. It enables new business processes to be rapidly constructed by linking services offered by new and existing business systems. Conductor fuses three key technologies - Service Registry, Web Service Engine, and Collaboration Engine - to enable public or private hubs (Figure 1):

- *Service Registry*: All Portal Services and Web Services are described and published in the Conductor Service Registry. A *Conductor Community Hub* is a server that administers the registry as well as managing

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

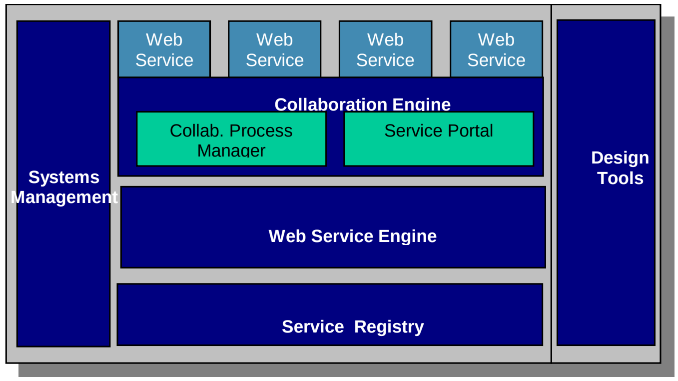

Figure 1: Key technology components of Conductor.

service provisioning, subscription, and interoperability. A Service Registry defines business organizations and business services in a "community". A community can be an enterprise, an extended enterprise, or a public marketplace. The Service Registry significantly extends from UDDI-based registries to include specifications of organizational structures, trading partners, and related security information. Cross-community interoperation (e.g., between a buyer enterprise and a supplier enterprise) is supported through Service Registry interoperation. Service Registry contents can also be uploaded to, or aggregated into, a standard UDDI registry to be searched by UDDI-based tools for partner and service discovery.

- *Web Service Engine*: Web Service Engine provides the underlying connectivity fabric for web service messages. Instead of just supporting simple web service invocations, Web Service Engine supports Soap extension for robust business interactions (see next section); it obtains information needed for these extensions from the Service Registry.
- *Collaboration Engine*: Collaboration Engine allows new business processes to be quickly constructed out of new or existing business systems while weaving in human activities, creating new business models and partner interactions. It consists of a *Collaborative Process Manager (CPM)* and a *Service Portal*. CPM links and orchestrates human actions (portal services) as well as business applications (web services) across organizations. End users access activities assigned to them in the context of a business process via a Service Portal accessible to users within an enterprise as well as to trading partners.

2 From Simple Web Services to Collaborative, Enterprise-grade Web Services

While simple web services as currently defined by SOAP and WSDL standards are useful in intra-enterprise information aggregation, or for readily available inter-enterprise queries, they lack security, reliability, and other key collaboration features needed for general business transactions. Conductor directly supports the following extensions to standard web services:

- Security, reliability, routing and addressing: A service provider can specify if collaboration parties are required to provide signing, encryption, and/or other security information when invoking a service instance. It can also specify if reliable transport must be used, and if other delivery features are expected, such as delivery receipts or error returns. Furthermore, services can be logically routed within a community or cross communities, eliminating the restriction and impracticality of point-to-point service invocations.
- Business document schemas and document family: Business document schemas are explicitly supported in the Registry and they are grouped into document families. For example, XML schemas for xCBL 3.5 PurchaseOrder and 4.0 Purchase Order are two versions of documents in the PO document family. A service definition may refer to xCBL PO as the message content, while the service instance can further specify which version or versions of PO are supported in the service implementation.

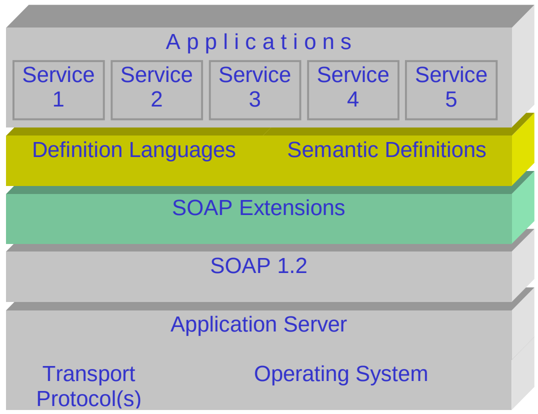

Figure 2: Standards Stack for Service-based Applications in Conductor.

- Choreography and collaborative (or long-running) services: Choreography specifies the sequence of service invocations expected of collaboration parties. For example, a service definition "OrderManagement_SupplierSide" service may contain definitions of 2 operations: "receive_order" and "receive_order_confirmation". A choreography specification may require that the first service invocation occurs before the second one. It is expected that all service instances that conform to this service definition also conform to the choreography specification.

The standards stack that a Conductor application (a web service) is built on is shown in Figure 2. It outlines 2 major extensions on top of standard Application Server and SOAP 1.2: (a) Definition Language and Semantic Definition extensions, and (b) SOAP extensions.

3 Implementing Web Service Extensions

Service Registry supports extended definitions and service descriptions. Like WSDL, service descriptions are partitioned into a definition part ("types") and implementation and binding part ("instances"). The Registry provides structured storage for all elements in the service definitions and descriptions, promoting reusability while offering the ability to "materialize" the descriptions in WSDL documents. Service Registry supports the following definitions in addition to standard WSDL information:

- Service instance attributes such as security, reliability, delivery properties, routing flexibilities
- Document schemas (based on XSD definition language) and semantic document family (based on published xCBL [10] and UBL [9] standards); transformation maps (based on XSLT but also accommodating other high performance variations) among members of document family and between document families.
- Service choreography definitions.

3.1 Enterprise-grade Web Services

Conductor's Web Service Engine processes service invocation messages based on the extended service standards. It embeds an interoperability resolver and supports Soap Extensions. Interoperability resolver is capable of computing document transformation rules and routing rules based on service descriptions in the Service Registry. For example, if a service invoker invokes an OrderManagement_Supplier service with a version 3.5 of the PO document, while the service only supports version 4.0 of PO, a transformation map is applied directly in the Web Service Engine before the invocation is sent to the service.

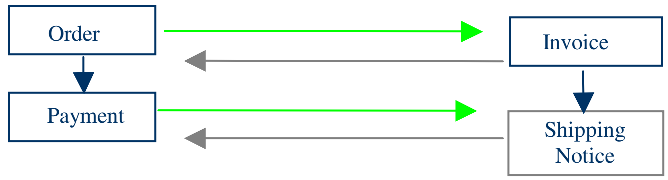

Figure 3: Choreographed document exchange as conversation process.

Soap Extensions define SOAP modules that provide enhanced functionality over simple SOAP to provide security, reliability, and collaboration extensions. In defining such extensions, we adopt small, self-contained, reusable SOAP module specifications; each specification provides useful functionality on its own. We also define profiles that describe how specifications are used in combination. For example, in a B2B scenario, signing, encryption, and authentication are needed in the protocol, which can be ommitted in an intra-enterprise query. This approach is taken in contrast to a single monolithic specification of complex web service, which will be too big for lightweight solutions, and will take too long to develop. The Soap Extensions supported in the Web Service Engine are summarized below, with references to other related evolving standards where appropriate:

- Security: Digital signatures, encrypted messages, and authentication are supported according to WS-Security [8]. In addition, a SAML-based mechanism [6] is used to provide the ability to pass authorization between servers.
- Reliability: Reliable messaging is achieved in Web Service Engine by adding a reliable queue-based transport. This is done in contrast to developing a reliable messaging extension in Soap. Therefore, reliable messaging is possible only when both ends of messaging use the same reliable queue implementation. On top of the reliable transport, delivery receipts (evidence that the message was delivered to the application), message expiry (how to determine if a message has ”expired”), and return messaging (how to specify the types of message to send back) are supported which strengthens audit and diagnosis required by most business transactions. The emerging WS-Transaction [8] standard, which attempts to provide atomic commit semantics, is an alternative.
- Addressing and Routing: Instead of direct, physical addressing to the service, a logical addressing scheme in the form of <Collaboration Party, Service Identifier, Action> has been defined. When sending a message to a community, the logical address uniquely identifies the server that handles the message within that community. In addition, the protocol includes a provisioning for specifying a ”return address”, and for dynamic routing. The latter differs from the WS-Routing [8] spec which is a mechanism based on computing the complete route of the message in advance. Dynamic routing offers more flexibility when crossing multiple communities, especially relevant in the B2B scenarios.
- Message Metadata: Message Metadata specifies message identification and correlation. It also provides references to agreements and choreographies. This is a key extension to support collaborative or long-running web services.

3.2 Collaborative Web Services

In order for enterprises to collaborate at the business-process level, web services must be extended to allow long-running interactions. Point activities in peer services are linked together in the context of a long-running business process instance that we call a conversation. Figure 3 illustrates a conversation.

Conductor Soap Extensions support a correlation identifier field in Metadata, which identifies the message with a long-running conversation. This field is automatically populated and passed along during the course of the conversation when collaborative services interact. Collaborative services are driven by process descriptions in addition to service interface descriptions. We distinguish between public conversation models, specifying inter-service document flows, and private business process models, specifying local workflows or compositions

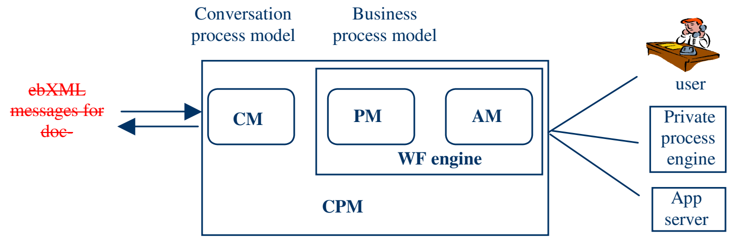

Figure 4: CPM - conversation manager, process manager and action manager.

of local services. Emerging standards for public conversation models fall into 2 categories: those specifying the global interface of a conversation (e.g. BPSS [2]), and those specifying the local interface of a conversation (e.g. WSCI [7], BPEL4WS [1]). In Conductor, a collaborative service definition can refer to a BPSS-based public conversation specification as its Choreography. By specifying a choreography, the service defines the expected behavior of collaborating services. Support for languages other than BPSS can be added. Collaborative services are implemented by the CPM component (see Figure 4) within Collaboration Engine [3, 4]. CPM consists of a Process Manager (PM) and a Conversation Manager (CM). PM executes the private business processes, while CM is responsible for validating and enforcing the choreography in the context of a conversation. The Action Manager (AM) dispatches and invokes local applications, services or processes to perform process tasks.

4 Conclusions

In this article we provided an overview of a web services-based business collaboration platform. We described key web service extensions required to serve the needs of business collaborations; these requirements are compiled based on Commerce One's extensive experience in developing B2B business applications and conducting B2B transactions on Market Place platforms. Conductor represents a wave of next-generation collaboration platforms to power standards-based application networks for extended enterprises.

References

[1] Business Process Execution Language for Web Services, Version 1.0, July 2002. http://www-06.ibm.com/developerworks/library/ws-bpel/

[2] Business Process Specification Schema, V1.01, 2001. ebXML.org.

[3] Qiming Chen, Meichun Hsu, Inter-Enterprise Collaborative Business Process Management, Proc. of ICDE 2001, Germany.

[4] Qiming Chen, Meichun Hsu, CPM Revisited - An Architecture Comparison, Proc. of CoopIS 2002, USA.

[5] The Web Services Payoff, Forrester Report, Dec. 2001. Which Industries will adopt Web Services?, Forrester Brief, Dec. 2001.

[6] Committee working draft, Binding and Profiles for the OASIS Security Assertion Markup Language (SAML). http://www.oasis-open.org/committees/security/docs/draft-sstc-bindings-model-09a 10 Jan. 2002

[7] Web Service Choreography Interface, Tech Report by Italio, SAP, BEA, Sun Microsystems. 2002.

[8] WS-specs http://msdn.microsoft.com/webservices/understanding/specs/default.aspx

[9] http://www.oasis-open.org/committees/ubl/

[10] http://www.xCBL.org.