Seminar
Web Services Orchestration

Thiago Rodrigues Colucci

10 de maio de 2010

Sumário

1 Web Service

1.1 What?

- Web services are application components

- Web services communicate using open protocols

- Web services are self-contained and self-describing

- Web services can be discovered using NEDI

- Web services can be used by other applications

- XML is the basis for Web services

1.2 Why?

1.2.1 Interoperability

When all major platforms could access the Web using Web browsers, different platforms could interact. For these platforms to work together, Web-applications were developed.

Web-applications are simple applications that run on the web. These are built around the Web browser standards and can be used by any browser on any platform.

1.2.2 Reusable application-components

There are things applications need very often. So why make these over and over again?

Web services can offer application-components like: currency conversion, weather reports, or even language translation as services.

1.2.3 Connect existing software

Web services can help to solve the interoperability problem by giving different applications a way to link their data.

With Web services you can exchange data between different applications and different platforms.

1.3   How?

The basic Web services platform is XML + HTTP.

XML provides a language which can be used between different platforms and programming languages and still express complex messages and functions.

The HTTP protocol is the most used Internet protocol.

Web services platform elements:

- SOAP (Simple Object Access Protocol)

- UDDI (Universal Description, Discovery and Integration)

- WSDL (Web Services Description Language)

2   Orchestration

2.1   What?

Orchestration describes how web services can interact with each other at the message level, including the business logic and execution order of the interactions. These interactions may span applications and/or organizations, and result in a long-lived, transactional, multi-step process model.

Orchestration is not defined as a W3C standard. It's standardized by OASIS. The co-chair of the Web Service Choreography working group of W3C (in 2005), Steve Ross-Talbot, said that "OASIS is a very different kind of organization. By contrast, it's much more vendor-led than the W3C. The W3C is responsible for all the core building blocks of the Web, for Web services and for the semantic Web. All the ancillary things you need to make Web services happen, such as management and orchestration, are in OASIS".

But the fact is that the industry is embracing orchestration as the way to specify, generate and control business process built upon web services. This was an initiative of IBM with Microsoft to develop a unified standard for Orchestration.

2.2   Why?

The principal motive is to embrace the use of web services and jump to the next level of the Internet. Depending basically on web services, the business process becomes interoperable and scalable. Besides, there's already a lot of open web services available to use, shortening the development time.

2.3   How?

Using open standards to connect web services together. Basically, we first describes the orchestration through a XML document, then we generate an executable process from this file, and finally we execute the process. That's all.

2.4   Standard

2.4.1   History

Historically (2002), three standards arose from the cooperation between companies (e.g. IBM, Microsoft, Sun, BEA, and others) interested on the development of the Web:

BPEL4WS   stands for "Business Process Execution Language for Web Services", the specification — called BPEL, for short — models the behavior of Web services in a business process interaction. It provides an XML-based grammar for describing the control logic required to coordinate Web services participating in a process flow. The WSDL interface defines the specific operations allowed, and BPEL defines how to sequence them.

WSCI   is the Web Service Choreography Interface. It defines a collaboration extension to WSDL, describing only the observable behavior between Web services. It does not address the definition of executables business process as BPEL does. But choreography is not the theme of this seminar.

BPML   the Business Process Management Language also incorporate WSCI support. BPML and WSCI share the same underlying process execution model, so developers can use WSCI to describe public interactions among business process and reserve PBML for developing private implementations.

In the beginning of 2003, all the companies supporting BPML discontinued it's support, migrating to BPEL4WS. It became a standard by OASIS, whom renamed the standard to WS-BPEL, the change was made to align with others Web Services standards.

Nowadays the main standard is still BPEL, with a lot of tools to deploy it's definition to executable process.

2.4.2   WS-BPEL

Before telling how to use BPEL, we first have to define some therms:

basic activity is an instruction that interacts with something external to the process itself. In a typical scenario, a BPEL executable process receives a message. Then it might invoke a series of services to gather additional data and subsequently respond to the requestor.

structured activities manage the overall process flow, specifying the sequence for referenced Web services. These activities also support conditional looping and dynamic branching. They are essentially BPEL's underlying programming logic for BPEL.

variables identifies the specific data exchanged in a message flow. When a BPEL process receives a message, it populates the appropriate variable so that subsequent requests can access the data. Variables are used to manage data persistence across Web services requests.

partnerLink can be any service that the process invokes or any service that invokes the process. Each partnerLink maps to a specific role that it fulfills in the business process.

scope tag To group a set of activities in a single transaction, BPEL uses a *scope* tag. The tag signifies that the enclosed steps should either all complete or all fail. Within this scope, a developer can specify compensation handlers that the BPEL orchestration engine can invoke in case of error.

Now, with our commands well known, we can analyze an BPEL description example.

2.4.3   Tools

Open Source   The main open source project for BPEL is surely Apache ODE (Orchestration Director Engine), which executes as executable process.

Intalio Works is an application that helps you describe a BPEL orchestration, then it uses ODE to deploy the executable. Intalio is build on Eclipse.

Proprietary   Oracle BPEL Process Manager are other graphical process creation tools for BPEL.

3   Bibliography

WST   Web Service Tutorial - `http://www.w3schools.com/webservices/`

WSO   C. Peltz. Web services orchestration: a review of emerging technologies, tools, and standards. Technical Report HP. January, 2003

WSOC   C. Peltz. Web Services Orchestration and Choreography. IEEE Computer Society, 20c.03.

NIMSA   Non-intrusive monitoring and service adaptation for WS-BPEL. 17th international conference on World Wide Web, 2008

IWSA   Gottschalk et al. Introduction to Web service architecture. IBM Systems Journal, Vol 41, NO 2, 2002.

WSBCBC   M. Stal. Web Services: Beyond Component-based Computing. Communications of the ACM, Vol 45, NO 10, 2002.

OASIS   OASIS(Organization for the Advancement of Structured Information Standards) - `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel`

BPEL   BPEL - `bpel.xml.org`

W3C   W3C - `www.w3.org/TR`

INTERVIEW   Interview with Steve Ross-Talbot - `http://searchsoa.techtarget.com/news/interview/`

BPSHO   Business Process Spec Handed Off to OASIS, Not W3C - `http://www.internetnews.com/d`

AHIB   A Hands-on Instruction to BPEL - `http://www.oracle.com/technology/pub/articles/m`

OPM   Oracle Process Manager - `http://www.oracle.com/technology/products/ias/bpel/ind`

WPS   WebSphere Process Server - `http://www-01.ibm.com/software/integration/wps/`

WID   WebSphere Integration Developer - `http://www-01.ibm.com/software/integration/wid/`

ODE   Apache ODE - `http://ode.apache.org/`

INTALIO   Intalio Works - `http://www.intalioworks.com/products/bpm/opensource-edition/#ov`