Orchestrating Web Services – Standards and Solutions

Elena Ivanova

Standards for orchestration of web services are considered. Brief overview of such technologies is presented. The main characteristics of available software tools for orchestration are derived and described. A case study about particular service orchestration software suits is presented.

Key words: *Web Services, Orchestration, BPEL, XPDL*

INTRODUCTION

Standards and technological solutions for orchestration appeared as a natural growth of web service technologies. They arose out of the need for composing several services into one and thus forming more complex services and business processes. There is another term and group of technologies, relevant to composing of services – choreography. Choreography refers to global, multiparty, peer-to-peer collaborations where the component entities interact in long-lived stateful and coordinated way regardless of any programming model or supporting platform used. Choreography languages (examples are WSCI, WS-CDL, etc) are mainly descriptive and cannot be directly executed. These languages are necessary to be mapped to an orchestration language in order to be executed. Orchestration focuses on the behaviour of the composed services as a whole. Orchestration languages (e.g. BPLM, BPEL, XPDL, BPELJ, jPDL, etc) are executable languages and define a runtime environment for their execution.

The paper is focused on orchestration and is organized as follows. The next section is devoted to the standards and technologies, related to orchestration of web services. After that the attention is paid on the software tools providing means for orchestration. The next is a case study about particular service orchestration software suits. The paper finishes with some concluding remarks.

ORCHESTRATION TECHNOLOGIES – BRIEF OVERVIEW

Term Orchestration refers to coordinating of multiple tasks. So, in terms of services Orchestration means coordinating of several services (i.e. their relations and consequence of execution) so that the result is a complex service or a process composed by these services.

There are several standardization bodies and organizations that aim at developing and adopting different orchestration technologies. Some of them are listed below in alphabetical order.

- **JSR 207:** (Java Specification Request), http://www.jcp.org/en/jsr/- is an initiative of the Java Community Process (JCP) that standardizes how to automate business processes on a J2EE server. The basic model of this JSR is to define a special type of ejb session bean that acts as the interface for a business process. The JSR wants to standardize a set of XML meta tags that should be specified as meta data (JSR175).
- **OASIS** (Organization for the Advancement of Structured Information Standards), http://www.oasis-open.org/home/index.php – is another international consortium working towards development, convergence, and adoption of common e-business standards.
- **OMG** (Object Management Group), http://www.omg.org - is an open membership, not-for-profit consortium that produces and maintains computer industry specifications for interoperable enterprise applications.
- **WfMC** (The Workflow Management Coalition), http://wfmc.org/ - is a consortium of about 300 organizations that defines a set of related standards based on an interesting reference model. The reference model describes the relation between a WFMS and other actors.

During the last five-six years several service orchestration languages have been developed and adopted. The next table presents short descriptions of these languages as well as their sources.

Table 1

Standard	Description
BEPLJ (BPEL for Java) *URL*: www.ibm.com	It is a combination of BPEL and Java, proposed by IBM and BEA, which allows sections of Java code (snippets) to be included in BPEL process definitions. Snippets are expressions or small blocks of Java code, used for: loop conditions, branching conditions, variable initialization, Web service message preparation, etc. BPELJ introduces a few minor changes to BPEL as well as several extensions in order to fit BPEL and Java conveniently together.
BPEL (Business Process Execution Language) *URL*: www.oasis-open.org	This is a convergence of WSFL and XLANG, written by developers from BEA, IBM, SAP, Siebel and Microsoft. It is an XML-based language, depending on WSDL, XML-Schema and XPath, providing a language for the formal specification of business processes and interaction protocols. It supports both abstract and executable processes. In BPEL, a business process is composed of elements ("activities") that define activity behaviors, including the ability to invoke Web services and control flow, and to compensate when errors occur. The resulting business process is exposed as one or more Web services.
BPML (Business Process Modelling Language) *URL*: www.bpmi.org	This is a textual service-oriented process modelling language, based on XML, XML-NS, XML-S, XPath and WSDL. It provides an abstracted execution model for collaborative and transactional business processes. It is a strict superset of BPEL, although BPEL have been endorsed by for BPMI phase 2.0 instead of BPML.
BPSS (Business Process Specification Schema) *URL*: www.ebxml.org	BPSS provides a standard framework (language) for business process specification. As such, it works with the ebXML Collaboration Protocol Profile (CPP) and Collaboration Protocol Agreement (CPA) specifications to bridge the gap between Business Process Modeling and the configuration of ebXML compliant e-commerce software.
jPDL (jBPM Process Definition Language) *URL*: jbpm.org	It is an XML based process execution language running in the Open Source JBoss jBPM (Java Business Process Management) workflow management system. Java actions can be triggered from the language.
JSR 207 (Java Specification Request) *URL*: www.jcp.org/en/jsr/	This is an initiative of the Java Community Process (JCP) that standardizes how to automate business processes on a J2EE server. The basic model of this JSR is to define a special type of ejb session bean that acts as the interface for a business process.
Orch *URL*: www.cs.utexas.edu/users/wcook/projects/orc/	This is a programming language and system for orchestrating distributed services. The Orc model [3] assumes that basic services, like sequential computation and data manipulation, are implemented by primitive sites. Orc provides constructs to orchestrate the concurrent invocation of sites to achieve a goal – while managing time-outs, priorities, and failure of sites or communication.
XLANG *URL*: www.microsoft.com	It is an extension of WSDL, providing both the model of an orchestration of services and collaboration contracts between orchestrations. XLANG, like BPML, were designed with an explicit -calculus theory foundation.
XPDL (XML Process Definition Language) *URL*: www.wfmc.org	It is an XML-based language from the Workflow Management Coalition (WfMC) for defining business processes. XPDL was based around a common set of functions for work distribution found in most workflow products. The XPDL's syntax is specified by an XML Schema.
WSFL (Web Services Flow Language) *URL*: www.ibm.com	It is an XML language, from IBM, for the description of Web Services compositions as part of a business process definition. It relies and complements existing specifications - SOAP, WSDL, XMLP, and UDDI.

The chronological developments of these languages are tried to be represented on fig.1.

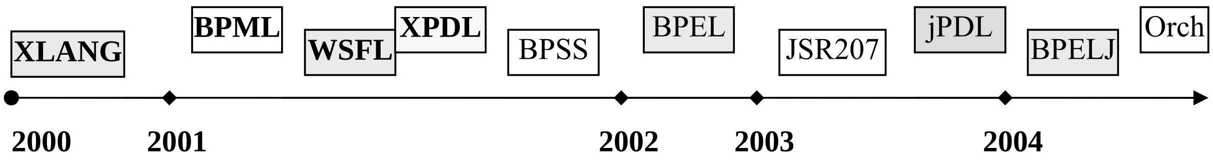


Fig.1 Chronology of Orchestration Technologies

Available orchestration technologies can be divided into three groups, according to their basic language: XML based, programming language based (mostly java based), and other based. This classification is represented on the following figure 2.

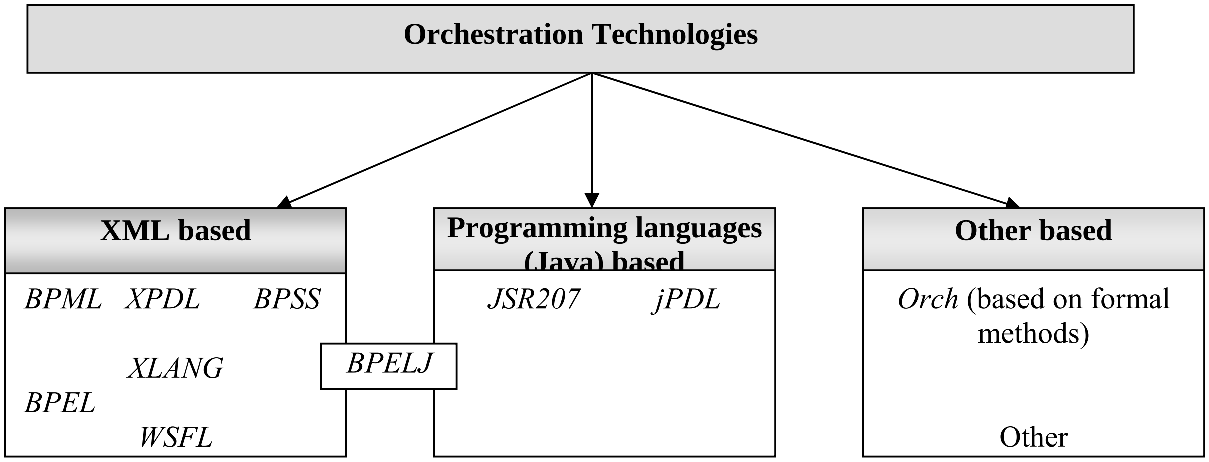

Orchestration Technologies
XML based
Programming languages (Java) based
Other based
BPML
XPDL
BPSS
XLANG
BPEL
WSFL
BPELJ
JSR207
jPDL
Orch (based on formal methods)
Other

Fig.2 Types of Orchestration Technologies

SOFTWARE TOOLS FOR SERVICE ORCHESTRATION – MAIN FEATURES

Service orchestration tools are usually part of complex workflow applications (applications for automation of business processes). A large amount of service orchestration software tools is available. Our studies came across more than 90 products. The list, of course, is not complete, but it is sufficient to illustrate the importance of the developments in such application area. The products have different level of maturity and they are presented both as market available products and as an open source software suits.

The main functional characteristics of service orchestration tools can be derived from the Workflow Reference model [1], which defines the main components and interfaces within the workflow architecture. In short the five WfMC interfaces can be characterized as follows. For details refer to [2]. Interface 1 refers to exchange of workflow descriptions. Interface 2 considers the invocation of workflow engines by workflow clients. Interface 3 supports the invocation of applications performing the specified workflow activities. Interface 4 supports workflow collaboration. Interface 5 supports the monitoring and administration of workflows including the life cycle management of distributed workflows. The following functionalities are specific to the service orchestration tools:

- specification of executable processes – creating and specifying processes, in accordance with some orchestration standard like BPEL and XPDL;
- simulation – simulating and testing the process execution using sample data and breakpoints;

- deployment – deploying the process onto the appropriate server location;
- execution – executing of the deployed processes;
- administration and monitoring – managing and monitoring of the running processes.

There are tools, supporting all these functions, and others, supporting only the first or/and the second function. Thus, service orchestrating products can be divided into three categories, according to their major functionality: editors (supporting mainly the first function – specification and/or simulation), engines (supporting mainly the second function – deployment and execution) and complex tools (supporting all aforementioned functions).

It can be made another classification of the examined tools, according to the supported orchestration standard: BPEL and XPDL, the most widely used standards. Some of the relevant tools are given in the tables below.

Table 2a BPEL based tools

ActiveBPEL Engine	Bexee	MidOffice
ActiveBPEL Designer	Biztalk Server	BEA WebLogic
ADONIS	Cape Clear Orchestrator	Parasoft BPEL Maestro
Apache Agila	iGrafx BPEL	PXE
Oracle BPEL Process Manager	OpenLink - Virtuoso Universal Server	SAP NetWeaver Exchange Infrastructure

Table 2b XPDL based tools

Aspose.Workflow	Newgen	Vignette Process Workflow Modeler
Agentflow	ObjectWeb BONITA	
Enhydra JaWE	OFBiz Workflow Engine	wfmOpen
Enhydra Shark	Open Business Engine	

Among the more than 90 found products, approximately 16% of them are based upon XPDL standard, approximately 24% of them are based upon BPEL standard, 4% of them are based on other standard (like WSFL and jPDL), and the rest are based on other (proprietary) technology. The most commercial tools are based on BPEL standard.

CASE STUDY

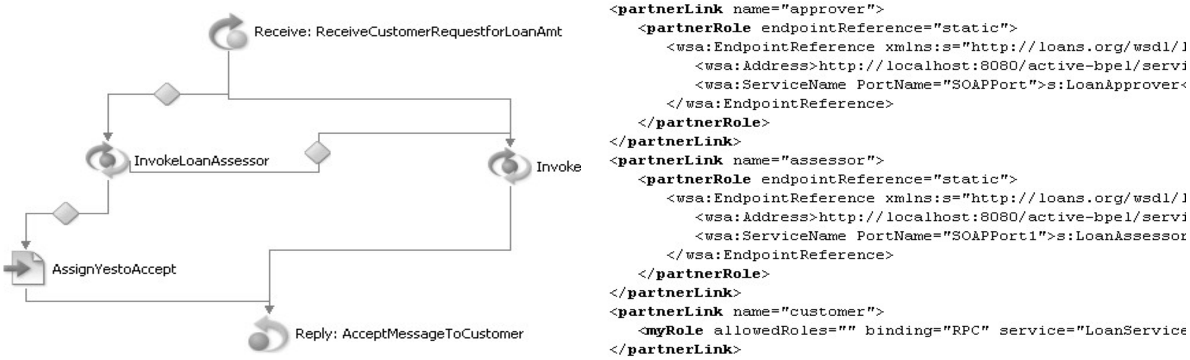
A case study in this section will illustrate the described functionalities of particular service orchestration tools: ActiveBPEL Designer and ActiveBPEL Enterprise, the products of Active Endpoints [4, 5], one of the leading companies in this area.

ActiveBPEL Designer is a comprehensive visual tool (editor) for creating, testing and deploying composite applications, based on the BPEL standard. It is a plug-in to the Eclipse (www.eclipse.org) integrated development environment. Having in mind the five specific functions of orchestrating tools, outlined above, the tool covers three of them - specification, simulation and deployment of executable complex business processes.

Specification. Processes are built by choosing partners, services and operations, and defining how data flows among those entities. It is done by organizing icons on the Process Editor canvas, and at the same time the tool constructs valid BPEL code. Figure 3 presents a typical picture of an ActiveBPEL process on the left hand side, and the corresponding BPEL code snippet as generated by ActiveBPEL Designer on the right hand side.

Simulation. The created process is simulated by using sample data and setting break points.

Deployment. The necessary files for deployment are automatically packaged and deployed onto the appropriate server location.


Receive: ReceiveCustomerRequestforLoanAmt
InvokeLoanAssessor
Invoke
AssignYestoAccept
Reply: AcceptMessageToCustomer

```
<partnerLink name="approver">
   <partnerRole endpointReference="static">
      <wsa:EndpointReference xmlns:s="http://loans.org/wsdl/l
         <wsa:Address>http://localhost:8080/active-bpel/servi
         <wsa:ServiceName PortName="SOAPPort">s:LoanApprover<
      </wsa:EndpointReference>
   </partnerRole>
</partnerLink>
<partnerLink name="assessor">
   <partnerRole endpointReference="static">
      <wsa:EndpointReference xmlns:s="http://loans.org/wsdl/l
         <wsa:Address>http://localhost:8080/active-bpel/servi
         <wsa:ServiceName PortName="SOAPPort1">s:LoanAssessor
      </wsa:EndpointReference>
   </partnerRole>
</partnerLink>
<partnerLink name="customer">
   <myRole allowedRoles="" binding="RPC" service="LoanService
</partnerLink>
```

Fig.3 ActiveBPEL process

ActiveBPEL Enterprise is a complete BPEL engine running either on top of a J2EE application server or standalone with a web servlet container. **ActiveBPEL Enterprise** servers are high performance BPEL servers that deliver many enterprise features including static analysis, process persistence, process versioning, extensive runtime web console, programmable Web Service and Java APIs plus diagram-based diagnostics. This tool covers three of specific functions – deployment, execution, and administration and monitoring of executable business processes.

Deployment. In order for the engine to execute a BPEL process, several files must be packaged in a business process archive (.bpr) file for deployment.

Execution. ActiveBPEL Enterprise provides an enterprise-level business process execution engine running as an application in the Tomcat server. It can be used to execute BPEL compliant processes, to manage the server's configuration, to suspend or resume running process instances.

Administration and monitoring. Web-based consoles allow the configuration and administration of the server's execution environment and deployed and running processes. Statistics data for deployed, active, completed, terminated processes are available.

CONCLUDING REMARKS

In this paper, a brief overview of orchestration standards as well as software tools providing means for orchestration was presented. A case study, describing functionalities of the particular service orchestration products: ActiveBPEL Designer and ActiveBPEL Enterprise, was illustrated.

As can been seen, many technologies are involved in the area of service orchestration. Two of them - BPEL and XPDL, have found wide acceptance, having in mind the software products, which are based on them. The list of software tools, supporting service orchestration, is quite long, and shows the importance of the developments in such application area. There are at least five major functional characteristics, specific to such kind of products – specification, simulation, deployment, execution and administration of executable processes.

REFERENCES

[1] WfMC: *Workflow Reference Model*; available at http://www.wfmc.org/standards/model.htm
[2] Hollingworth D.; *The Workflow Reference Model –10 Years On*; Workflow handbook 2004
[3] Misra J.and W. Cook, 2004, *Computation Orchestration: A Basis for Wide-Area Computing,* Lecture Notes for NATO summer school, held at Marktoberdorf in August 2004
[4] http://www.active-endpoints.com/
[5] http://www.activebpel.org/

Contacts:

Assistant prof. Elena Ivanova, Department "Hierarchical Systems", Istitute of Computer and Communication Systems, BAS, E-mail: elena@hsi.iccs.bas.bg

This research is partly supported by the European Commission, project №FP6-027178, and National Scientific Fund of Bulgaria, project № ВУ-МИ-108/2005.