WS-BPEL 2.0

Web Services Business Process Execution Language
Technical Introduction

Frank Ryan, Active Endpoints, Inc.
V.P. Technical Services
frank.ryan@active-endpoints.com
WS-BPEL 2.0

- BPEL is the Web Services Orchestration standard from OASIS
  - bee•pel’, beep’•uhl, bip’•uhl
- An XML-based grammar for describing the logic to orchestrate the interaction between Web services in a business process

BPEL Historical Timeline

Dec 2000
Microsoft publishes XLANG

March 2001
IBM publishes WSFL

July 2002
IBM, Microsoft and BEA converge WSFL & XLANG into BPEL4WS 1.0

March 2003
BPEL4WS is submitted to OASIS

May 2003
OASIS publishes BPEL4WS 1.1

1stH 2007
WS-BPEL 2.0 released
Motivation

- Integration continues to be a key problem facing businesses
  - Intra-enterprise integration (Enterprise Application Integration)
  - Integrating with partners (Business-to-Business Integration)
  - Syndication

- Web services move towards service-oriented computing
  - Applications are viewed as “services”
  - Loosely coupled, dynamic interactions
  - Heterogeneous platforms
  - No single party has complete control

- Service composition
  - How do you compose services in this domain?
Integration
Why the Need For BPEL?

- WSDL defined Web services have a stateless interaction model
  - Messages are exchanged using
    - Synchronous invocation
    - Uncorrelated asynchronous invocations
- Most “real-world” business processes require a more robust interaction model
  - Messages exchanged in a two-way, peer-to-peer conversation lasting minutes, hours, days, etc.
- BPEL provides the ability to express stateful, long-running interactions
Two Programming Levels

- Programming in the large
  - Non-programmers implementing flows
    - Flow logic deals with combining functions in order to solve a more complex problem (such as processing an order)

- Programming in the small
  - Programmers implementing functions
    - Function logic deals with a discrete fine-grained task (such as retrieving an order document or updating a customer record)
Process Usage Patterns

- Aiming for a single approach for both
  - Executable processes
    - Contain the partner’s business logic behind an external protocol
  - Abstract processes
    - Define the publicly visible behavior of some or all of the services an executable process offers
    - Define a process template embodying domain-specific best practices
Process Model Requirements

- Portability and Interoperability
- Flexible Integration
  - Rich, and easily adaptable to changes in underlying services
- Recursive, type-based composition, enables
  - Third-party composition of existing services
  - Providing different views on a composition to different parties
  - Increased scalability and reuse
- Stateful conversations and lifecycle management
  - Supports multiple stateful long-running conversations
- Recoverability
  - Long running business processes need fault handling and compensation mechanisms to manage and recover from errors
Benefits of BPEL

- Industry standard language for expressing business processes
  - Leverage a common skill set and language

- Designed to fit naturally into the Web services stack
  - Expressed entirely in XML
  - Uses and extends WSDL 1.1
  - Uses XML Schema 1.0 for the data model

- Portable across platform and vendor
  - Will run on any BPEL-compliant engine

- Interoperable between interacting processes
  - Layering on top of Web services stack
Relationship with WSDL

- BPEL is layered on top of and extends the WSDL service model
  - WSDL defines the specific operations allowed
  - BPEL defines how WSDL operations are orchestrated to satisfy a business process
  - BPEL also specifies extensions to WSDL in support of long-running asynchronous business processes
WS-BPEL in the WS-* Stack

- You are here
- WS-BPEL
- WSDL, Policy, UDDI, Inspection
- Security
- Reliable Messaging
- Transactions
- Coordination
- SOAP (Logical Messaging)
- Other protocols
- XML, Encoding
- Other services
- Business Processes
- Description
- Quality Of Service
- Transport and Encoding