WS-BPEL
*Web Services Business Process Execution Language*

John Evdemon
Architect, Microsoft
Co-Chair, OASIS WS-BPEL Technical Committee
The Importance of Architecture

The Winchester “Mystery” House

- 38 years of construction – 147 builders, 0 architects
- 160 rooms – 40 bedrooms, 6 kitchens, 2 basements, 950 doors
- 65 doors to blank walls, 13 staircases abandoned, 24 skylights in floors
- No architectural blueprint exists
Abstract

- Business Processes play a key role in Business-to-Business (B2B) and Enterprise Application Integration (EAI) scenarios by exposing service invocation and interaction patterns.
- Business Processes are the fundamental basis for building heterogeneous and distributed workflow applications.
- The Web Services Business Process Execution Language (WS-BPEL) language is designed to specify business processes that are composed of and exposed as Web services.
- An abstract WS-BPEL representation can be use to represent publicly observable behaviors in a standardized fashion.
- This presentation provides an overview of the WS-BPEL language and illustrates how it can be used to compose web services into complex service interaction patterns.
- A review of Microsoft’s BPEL capabilities will also be discussed.
Agenda

- Motivation
- OASIS and WS-BPEL
- Main Concepts
- Examples
- Status and support
WS-BPEL? BPEL? BPEL4WS?

How do you pronounce it?

Why is there no standard graphical notation?

What is the difference between orchestration and choreography?
Motivation

- Application integration is a key problem facing businesses
  - Intra enterprise integration (Enterprise Application Integration)
  - Integrating with partners (Business Process Integration)

- Web services enable service-orientation
  - 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?
Two-level Programming Model

- **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 observable 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 the services it is interacting with
  - Recursive, type-based composition, enables …
  - third-party composition of existing services
  - providing different views on a composition to different parties
  - inter-workflow interaction
  - increased scalability and reuse
- Separation and composability of concerns
  - Decoupled from the supporting mechanisms (quality of service, messaging frameworks)
- Stateful conversations and lifecycle management
  - Can carry multiple stateful long-running conversations
- Recoverability
  - Business processes, and in particular long running ones, need a way to build-in fault handling and compensation mechanisms to handle and recover from errors
WS-BPEL

- WS-BPEL enables …
  - Defining business processes as coordinated sets of Web service interactions, recursively into new aggregated Web services
  - Defining both abstract and executable processes
    - Abstract processes for B2B style specifications
    - Executable processes provide a model to integrating enterprise applications
  - Creating compositions of Web services
    - Composition based on abstract descriptions

- WS-BPEL was designed to provide an interoperable process model

- WS-BPEL comes from …
  - Strong roots in traditional flow models
  - Plus many concepts from structured programming languages
  - All laid on top of WSDL and core XML specifications
  - Merges WSFL and XLANG concepts
WS-BPEL Specifications

- **BPEL4WS 1.0 (7/2002)**
  - Original proposal from BEA, IBM, Microsoft
  - Combined ideas from IBM’s WSFL and Microsoft’s XLANG

- **BPEL4WS 1.1 (5/2003)**
  - Revised proposal submitted to OASIS
  - With additional contributions from SAP and Siebel

- **WS-BPEL 2.0**
  - Currently in OASIS undergoing standardization
  - Committee Draft specification available
WS-BPEL in the WS-* Stack

- **WS-BPEL**
- **WSDL, Policy, UDDI, Inspection**
- **Security**
- **Reliable Messaging**
- **Transactions**
- **Coordination**
- **SOAP (Logical Messaging)**
- **XML, Encoding**
- **Other protocols**
- **Other services**
- **Quality Of Service**
- **Transport and Encoding**

**Business Processes**

**Description**
Outline

- Motivation
- OASIS and WS-BPEL
- Main Concepts
- Examples
- Status and support
Getting the Players Together

BPEL4WS 1.1

(©) BPEL4WS 1.1 authors
OASIS Technical Committee

- 288 committee members
  - Largest technical committee at OASIS
  - 30 actively voting members, attending weekly calls, working on subcommittees, etc.

The WS-BPEL TC Charter

- OASIS Standardization
- Define common concepts for a business process language for usage patterns
  - Scope: process interface descriptions and executable process models
- Explicitly **does not** address
  - Bindings to specific hardware/software platforms
  - Mechanisms required for a “complete” runtime environment for process implementation
OASIS Technical Committee

Issues Process
- List of all issues available at
- Issue discussion
  - Weekly calls
  - Quarterly face to face meetings

Status
- Deadlines (need 2/3 majority to override)
  - No new feature issues since Aug 15, 2004
  - No new feature issue resolution proposals since April 1, 2005
  - Feature issues that are not resolved are marked as revisitable
- Latest approved committee draft: September 1, 2005
- Targeting Spring 2006 completion
WS-BPEL Design Goals

- Business processes defined using an **XML-based language**
- **Web services** are the model for process decomposition and assembly
- The **same orchestration concepts** are used for both the **external** (abstract) and **internal** (executable) views of a business process
- Both **hierarchical and graph-like** control regimes are used, reducing the fragmentation of the process modeling space
- An **identification mechanism for process instances** is provided at the application message level
- The **basic lifecycle mechanism** is in implicit creation and termination of process instances.
- A long-running transaction model is defined to support **failure recovery** for parts of long-running business processes
- Language built on **compatible Web services standards in a composable and modular manner**
OASIS WS-BPEL 2.0 To-Do List

- Open issues still under discussion
- Miscellaneous specification clarifications
- Abstract processes
  - Common base (syntax)
  - Profiles (semantics)
    - Externally observable behavior (as in BPEL4WS 1.1)
    - Templating
Outline

- Motivation
- OASIS and WS-BPEL
- Main Concepts
- Examples
- Status and support
WS-BPEL Language Structure

- Process
- Partner links
- Data handling
- Properties and correlation
- Basic and structured activities
- Scopes
BPEL and WSDL

- BPEL processes are exposed as WSDL services
  - Message exchanges map to WSDL operations
  - WSDL can be derived from partner definitions and the role played by the process in interactions with partners
Recursive Composition

- BPEL processes interact with WSDL services exposed by business partners

Interfaces exposed by the BPEL process

Interfaces consumed by the BPEL process
Composition of Web Services

Service A

Service P

Service B

A’s WSDL

P’s WSDL

BPEL

B’s WSDL

Partner Link Type

Partner Link Type
Partner Links

- Partner link: instance of typed connector
- Partner link type specifies required and/or provided portTypes
- Channel along which a peer-to-peer conversation with a partner takes place

![Diagram showing Partner Link Type with Process 1, Port Type 1, Partner Link Type, Port Type 2, and Process 2]
Scoped variables typed as WSDL messages or XML Schema elements/types

Activities’ input and output kept in scoped variables

Assignment activities move data around
Properties and Correlation

Messages in long-running conversations are correlated to the correct process instance.

Typed properties defined in WSDL are named and mapped (aliased) to parts of several WSDL messages used by the process.
<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>receive</td>
<td>Do a blocking wait for a matching message to arrive</td>
</tr>
<tr>
<td>reply</td>
<td>Send a message in reply to a formerly received message</td>
</tr>
<tr>
<td>invoke</td>
<td>Invoke a one-way or request-response operation</td>
</tr>
<tr>
<td>assign</td>
<td>Update the values of variables or partner links with new data</td>
</tr>
<tr>
<td>validate</td>
<td>Validate XML data stored in variables</td>
</tr>
<tr>
<td>empty</td>
<td>No-op instruction for a business process</td>
</tr>
<tr>
<td>throw</td>
<td>Generate a fault from inside the business process</td>
</tr>
<tr>
<td>rethrow</td>
<td>Forward a fault from inside a fault handler</td>
</tr>
<tr>
<td>exit</td>
<td>Immediately terminate execution of a business process instance</td>
</tr>
<tr>
<td>wait</td>
<td>Wait for a given time period or until a certain time has passed</td>
</tr>
<tr>
<td>compensate</td>
<td>Invoke compensation on an inner scope that has already completed</td>
</tr>
<tr>
<td>extensionActivity</td>
<td>Wrapper for language extensions</td>
</tr>
</tbody>
</table>
### Structured Activities

<table>
<thead>
<tr>
<th>Flow</th>
<th>Contained activities are executed in parallel, partially ordered through control links</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>Contained activities are performed sequentially in lexical order</td>
</tr>
<tr>
<td>While</td>
<td>Contained activity is repeated while a predicate holds</td>
</tr>
<tr>
<td>RepeatUntil</td>
<td>Contained activity is repeated until a predicate holds</td>
</tr>
<tr>
<td>Pick</td>
<td>Block and wait for a suitable message to arrive (or time out)</td>
</tr>
<tr>
<td>ForEach</td>
<td>Contained activity is performed sequentially or in parallel, controlled by a specified counter variable</td>
</tr>
<tr>
<td>If Then Else</td>
<td>Select exactly one branch of activity from a set of choices</td>
</tr>
<tr>
<td>Scope</td>
<td>Associate contained activity with its own local variables, fault handlers, compensation handler, and event handlers</td>
</tr>
</tbody>
</table>
Nesting Structured Activities

<sequence>
  <receive .../>
  <flow>
    <sequence>
      <invoke .../>
      <while ... >
        <assign>...</assign>
      </while>
    </sequence>
  </flow>
  <reply>
</sequence>
Scopes and Handlers

- **Scope**
  - Local variables
  - Local partner links
  - Local correlation sets
  - Set of activities (basic or structured)

- **Handlers**
  - Event handlers
    - Message events or timer events (deadline or duration)
  - Fault handlers
    - Dealing with different exceptional situations (internal faults)
  - Compensation handler
    - Undoing persisted effects of already completed activities
  - Termination handler
    - Dealing with forced scope termination (external faults)
Process Instance Lifecycle

- Business processes defined in BPEL represent stateful Web services.
  - When a process is started, a new instance is created according to the process definition.
  - The creation and destruction of BPEL process instances is by design implicit.
Outline

- Motivation
- OASIS and WS-BPEL
- Main Concepts
- Examples
- Status and support
1. A customer asks for a loan, providing name and amount info
2. Two services are involved:
   a) A risk assessor which can approve the loan if the risk is low
   b) A loan approver which checks the name and approves/disapproves the loan
3. The reply is returned to the customer
Fault Handling and Compensation

- Charge credit card
- Refund customer
- Ship goods
- Rethrow
- Notify manager
- Compensate

Error!
1. Assign EPR from partnerLink (myRole) to variable

2. Pass variable with A's callback EPR to process B

3. Receive variable with A's callback EPR from process A

4. Assign EPR from variable to partnerLink (partnerRole)

5. Invoke A's callback using the received EPR

6. Receive the callback from the other process
Public View (abstract)

Traveler Process
- submit to agent
- receive confirmation
- receive tickets

Agent Process
- get itinerary
- order tickets
- receive confirmation
- send confirmation
- send tickets

Airline Process
- get order
- confirm flight
- send tickets
Private View (executable)

**Traveler Process**
- plan trip
- submit to agent
- receive confirmation
- receive tickets

**Agent Process**
- get itinerary
- select airline
- order tickets
- receive confirmation
- send confirmation
- send tickets

**Airline Process**
- get order
- reserve seats
- charge credit card
- confirm flight
- send tickets

**Agent Process**
- plan trip
- submit to agent
- receive confirmation
- receive tickets

**Airline Process**
- get order
- reserve seats
- charge credit card
- confirm flight
- send tickets

**Traveler Process**
- get itinerary
- select airline
- order tickets
- receive confirmation
- send confirmation
- send tickets

**Airline Process**
- get order
- reserve seats
- charge credit card
- confirm flight
- send tickets
Outline

- Motivation
- OASIS and WS-BPEL
- Main Concepts
- Examples
- Messaging, support and status
**WS-BPEL**

- Interoperable process model for long running business processes

- Flexible integration of Web services
  - WSDL abstract interfaces alone used to define composition
    - Enables two levels of adaptive behavior
      - Abstract partners can be bound to actual services at runtime
      - The process can choose a protocol for communicating with the service at runtime

- Services whose data definitions do not match can be composed
  - Data transformations can be inlined in process definition
What’s new since BPEL4WS 1.1?

- New activity types
  - if-then-else – replacing the BPEL4WS 1.1 switch activity
  - repeatUntil – like BPEL4WS 1.1 while activity with at least one iteration
  - validate – explicit XML schema validation of WS-BPEL variable content
  - forEach – sequential or parallel iteration controlled by a counter variable
  - extensionActivity – designated WS-BPEL extension point for new activity types
- Completion condition in forEach activity
- Variable initialization
- XSLT for variable transformations
  - New XPath extension function bpws:doXslTransform(...)
- XPath access to variable data
  - XPath variable syntax $variable[.part]/location
- XML schema variables in Web service activities
  - Usability enhancement for WS-I compliant doc/lit-style WS interactions
- Locally declared messageExchange
  - Internal correlation of receive and reply activities
- Abstract processes
  - Common base (syntax) and profiles (semantics)
BPEL in Production

General Recommendation:
- BPEL is still under development at OASIS and should not yet be considered for a production environment

Observations:
- Many BPEL implementations are using it like a programming language
- At this point in time there are no portable (multi-platform) BPEL executable implementations
  - Please correct me if I’m mistaken!
Executable BPEL

- Organizations evaluating executable BPEL use it like a programming language
  - Missing some core capabilities to be a programming language for business processes
  - Some tools add proprietary extensions to BPEL

- A BPEL / SQL analogy

- Prediction:
  - The portability of executable BPEL will be relatively low to non-existent. This is because the spec is missing some fundamental concepts
BPEL in Products

- Microsoft BizTalk Server 2004
- Microsoft BizTalk Server 2006
- Microsoft Windows Workflow Foundation (under development)
- Active Endpoints ActiveWebflow Server
- ActiveBPEL Engine (open source)
- bexee BPEL Execution Engine (open source)
- Cape Clear Orchestrator
- FiveSight PXE
- IBM WebSphere Business Integration – Server Foundation 5.1
- IBM WebSphere Process Server 6.0
- OpenLink Virtuoso Universal Server
- OpenStorm ChoreoServer
- Oracle BPEL Process Manager
- Parasoft BPEL Maestro
- SeeBeyond eInsight BPM
- Twister (open source)
BPEL in BizTalk

- BizTalk 2004 was one of the earliest implementations of BPEL
  - Pre-OASIS
  - Implementation is unchanged in BizTalk 2006

General Recommendation:
- BPEL is still under development at OASIS and should not yet be considered for a production environment
BPEL Activities for WF

- BPEL 1.1 compliant activities on WF platform
  - Demonstrates WF platform can interop with BPEL 1.1 workflow processes
  - Demonstrates that workflow industry standards can be implemented on WF
Demos
WS-BPEL Opportunities

- Lack of a “global model”
- Compensation patterns
More opportunities

- Business Process Reification
- “Just-in-time” business processes
- Peer-to-peer packaging and sharing of workflow fragments (subprocesses)

How will orchestrated services will be managed?
- how are services provisioned to users?
- how is access control handled?
- how do you guarantee certain levels of quality?
WS-BPEL Application Areas

- Business Process Design
- Human Workflow
- Sub-processes
- Static Analysis
- Autonomic Computing
- Grid Computing
- Semantic Web
- Regulatory Compliance
- Process Analytics
- Better guidance for Abstract to Executable representations
- What about REST and other types of services?
Known extensions, related work

- BPEL4People – white paper
  - Human user interactions – as known from existing workflow engines

- BPEL Subprocesses – white paper
  - Based on a coordination protocol

- BPEL-J – white paper
  - Java bindings

- Business Process Extension Language (BPXL)
  - OMG/BPMI

  Transaction semantics
  - Aligned with WS-Transaction specifications

  Currency with related standards
  - WSDL 2.0, XQuery, etc.
WS-BPEL Resources

- OASIS Technical Committee
  [http://www.oasis-open.org](http://www.oasis-open.org)
- BPEL4WS 1.1
  [http://dev2dev.bea.com/technologies/webservices/BPEL4WS.jsp](http://dev2dev.bea.com/technologies/webservices/BPEL4WS.jsp)
  [http://www.siebel.com/bpel](http://www.siebel.com/bpel)
- WS-BPEL 2.0 – latest approved committee draft (September 2005)
- Info aggregator sites
  - Wikipedia
  - BPEL Resource Guide
    [http://bpelsource.com](http://bpelsource.com)
- Numerous books and conference papers
- Analyst reports
Web services orchestration/choreography is about connecting web services together

Key technical requirements that must be met:
- asynchronous support with correlation
- strong transactions semantics and exception handling
- set of programming constructs for describing workflow

A number of standards have been introduced
- W3C chartered to define choreography languages
- OASIS expected to finish BPEL this year (finally!)

Availability of platforms, tools and documentation will be vital to widespread adoption of these standards
Thanks!

john.evudemon@microsoft.com
Addendum:

Microsoft’s Workflow and Business Process Strategy
How Does Microsoft Define Workflow?

- **Workflow** is a set of activities that describe the implementation of a business process.

- **Workflow** can be short or long running and can involve people and systems.
Windows Workflow Foundation

Key Concepts

- Host Process
- Windows Workflow Foundation
- Runtime Engine
- A Workflow
- An Activity
- Runtime Services
- Base Activity Library
- Custom Activity Library
- Visual Designer: Graphical and code-based construction
- Workflows are a set of Activities
- Workflows run within a Host Process: any application or server
- Developers can build their own Custom Activity Libraries
- Components: Base Activity Library, Out-of-box activities and base for custom activities, Runtime Engine, Workflow execution and state management, Runtime Services, Hosting flexibility and communication

Ordering State Machine Example - Microsoft Visual Studio
Workflow and BizTalk Server

**BizTalk Server**
- Accelarators
- Workflow
- Messaging
- Transformation
- Adapters

**Design Tools**
- Business Activity Monitor And Admin Tools

**Visual Studio Designer**
- Future version will migrate to Windows Workflow Foundation for orchestration

**Windows Workflow Foundation**
- Workflow framework
- Exposed via WinFX
- Broad set of scenarios
- Used to build solutions
- Enables manageability and scale-out in solutions
- Use for building workflow into apps or workflow-enabled servers

**WinFX**
- Premium BPM server
- Distinct server product
- Use in B2B, EAI, BPM scenarios
- Deployable solutions
- Manageability, Scale-out
- Future version will migrate to Windows Workflow Foundation for orchestration
A Unified Workflow Strategy

- ...provide a Windows Workflow Foundation as a part of the developer framework for Windows

- ...and enterprise applications that build on top of Windows Workflow Foundation for specific scenarios.

Example

A ISV developer building a line of business application chooses to use Windows Workflow Foundation within the business logic of their application.

Example

An enterprise customer uses BizTalk Server to provide system workflow across their existing line of business applications and trading partners to reduce their cycle-times and increase business visibility in a high-volume environment.
A Unified Workflow Technology

- A framework, not a product
- Key features
  - Long running workflows in any application or server
  - Extensible activity framework
  - Flexible control flow
    - human and system scenarios
  - Transparency at runtime
  - Visual designer for graphical and code-based authoring
    - Embeddable, extensible
  - Exposed via WinFX
    - Powers Office 2007
  - Future orchestration engine in BizTalk Server