

Welcome to BPEL Fundamentals. This introduction is Unit #1 of our course. This course will teach you about Active Endpoints and our products, the BPEL language itself and how to use the ActiveVOS Designer. In the files you download there will be:

Presentation Files: These have the class content that will teach you how to use the BPEL language and the ActiveVOS Designer. (In Powerpoint format.)

Lab Documents: These have step by step instructions for the lab assignments, which give you practical experience using the tool to create a BPEL process. (In PDF format.)

Class Files: These are the basic Schema (.xsd) and WSDL (.wsdl) files that you will start with and use to complete the coursework (in .xsd or .wsdl format.)

Solution Files: These are a set of files that have the labs completed up to that point in the course. For example, Lab #7 solution files will have all of the labs completed correctly up to and including Lab #7 (in .xsd or .wsdl format.)

Unit Objectives

- At the conclusion of this unit, you will be familiar with:
 - Active Endpoints and the ActiveVOS family of products
 - Business Process Execution Language (BPEL) features and benefits
 - Process definition that you will build during the labs

2 Copyright © 2004-2008 Active Endpoints, Inc.



In this unit we will introduce Active Endpoints and the ActiveVOS family of products, the BPEL language and the business process that will be developed during this course.

About Active Endpoints


Active Endpoints is the leading provider of BPEL solutions

Market proven	Flexible, open architecture
Platform portability	High productivity

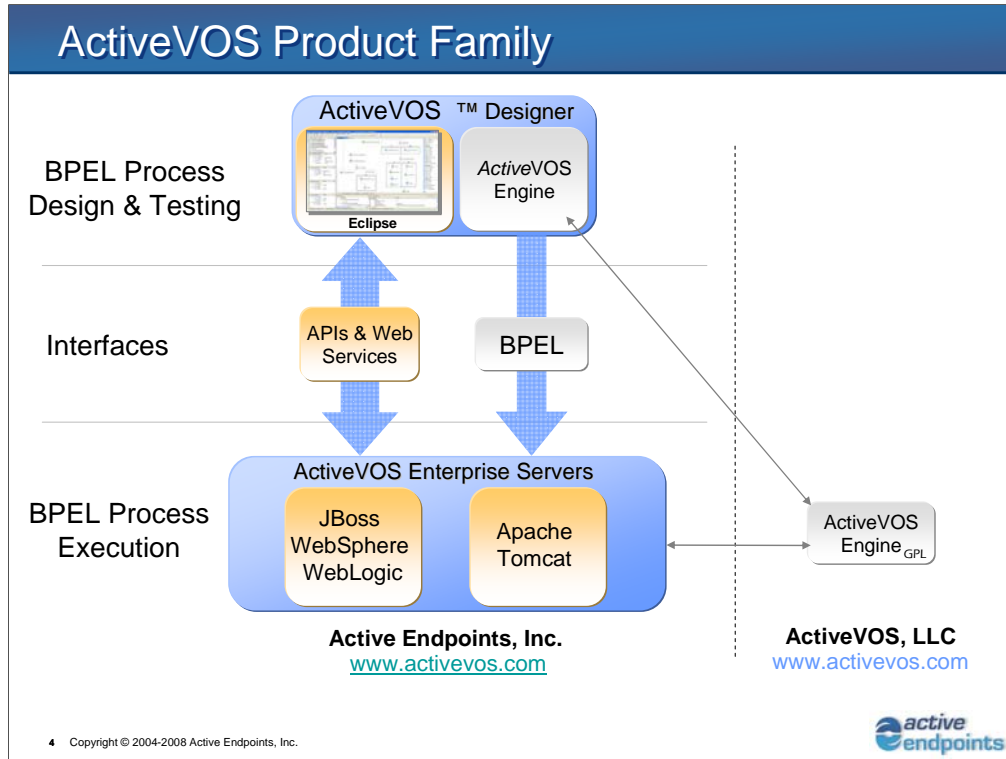
BPEL Market Firsts – Active Endpoints

- ✓ **Authoring tools for Eclipse**
- ✓ **Open source server**
- ✓ **Training programs**
- ✓ **Execution servers for Apache Tomcat, JBoss, WebSphere, WebLogic**

3 Copyright © 2004-2008 Active Endpoints, Inc.



Active Endpoints is the leading BPEL Solutions provider, focused exclusively on market-leading, BPEL-related products. Thousands and thousands of developers and organizations have downloaded ActiveVOS Designer and the ActiveVOS engine. ActiveVOS Designer was the first Eclipse-based authoring tool to be used exclusively for BPEL development. Active Endpoints was the first to have processes that could be deployed on an open source Server – ours runs on Tomcat - or on any BPEL-compliant engine such as those that run on WebLogic, WebShpere, etc. Active Endpoints was the first to have online training programs – like this one – and they can be delivered online or on your organization's site.



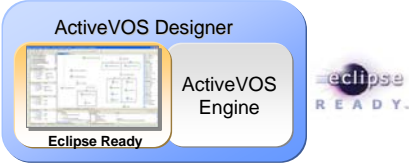
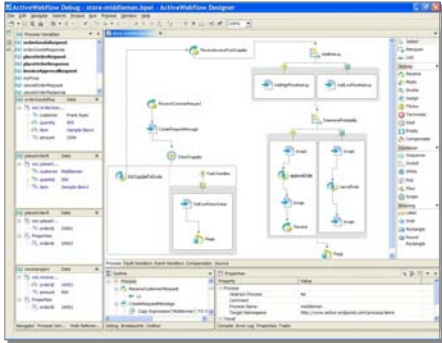
Active Endpoint products:

Designer is an Eclipse-based development tool, constructed as a set of plug-ins to the Eclipse platform and available in both free (community) and licensed (commercial) versions. The ActiveVOS Engine, which is open source, is included with the Designer for the convenience of debugging your processes. We also produce licensed business process execution engines, which have additional capabilities, beyond what is included in our open source Engine.


The ActiveVOS Enterprise Servers run on any ActiveVOS supported platform, and are for the deployment of developed processes. The bottom line is that the ActiveVOS Engine is open-source and the ActiveVOS Engine, the ActiveVOS Designer tool and the Tomcat Server you downloaded for this class are all free. The ActiveVOS Enterprise Servers are what you pay for, and are not open source.

ActiveVOS Designer

- Design
 - Eclipse Ready™
 - Top-down and bottom-up design patterns
 - Drag & drop process composition
 - Template-based mapping
 - Auto-generation of BPEL process definitions
 - Intelligent WSDL catalog
 - Interfaces View
 - Custom activities (BPELEts)
 - Pushbutton deployment
- Test
 - Design-time simulation & process testing through an embedded BPEL engine
- Debug
 - Diagram-based
 - Process debugging
 - Diagnostics of deployed processes



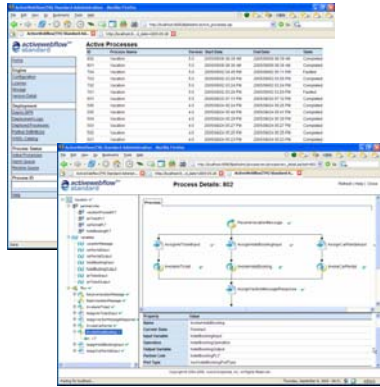
5 Copyright © 2004-2008 Active Endpoints, Inc.



Originally called “ActiveWebFlow”, and then “ActiveBPEL”, the ActiveVOS Designer is free, downloadable, and is configured as a set of Eclipse Ready plug-ins. We use Eclipse’s Drag & Drop graphic design interface, which can be used to support top-down or bottom-up development styles. The ActiveVOS Designer provides automatic generation of the BPEL code and BPEL process definitions. The intelligent cataloging of web service artifacts (i.e., WSDLs and Schemas) greatly simplifies the creation of process diagrams from your WSDL/Schema definitions. BPELEts are reusable components that can be added to the process editor palette and then re-used again and again. ActiveVOS products are used not just for the design and simulation of processes, but for their deployment and administration, as well. We can simulate a process locally or debug a deployed process remotely, all with full diagnostics.

ActiveVOS Enterprise Server


- BPEL process execution
- Process persistence & versioning
- BPEL process analysis
- Diagram-based debugging
- Load balancing, clustering & fault tolerance
- Browser-based management consoles
- Web Services APIs
 - Event/debug
 - Administrative



ActiveVOS Enterprise Servers

JBoss
WebSphere
WebLogic

Apache
Tomcat



6 Copyright © 2004-2008 Active Endpoints, Inc.

The ActiveVOS Enterprise Server's mission is to execute the processes we design. The licensed version allows for both process persistence and process versioning. Process *persistence* supports the retention of execution information from past and current instances in a database. Process *versioning* supports the changeover of a process to newer versions, and the handling (migration) of instances that are currently running against older versions of the process. We can perform process analysis through diagram-based debugging, which is supported by the ActiveVOS Engine. Our servers support load-balancing, clustering and fault tolerance for high performance and reliability. Our browser-based Server Administration Console allows you to perform process administration, monitoring and analysis from anywhere that has internet access. Finally, our Administration and debugging features are enabled as web services, and can be used in your own development projects.

Standards Based Implementation

- Leverages many important industry standards
 - Business Process Execution Language versions 2.0 (WS-BPEL) and 1.1 (BPEL4WS)
 - For describing web service orchestration flows
 - Web Services Description Language (WSDL) 1.1
 - For describing the public interface to a web service
 - Extensible Markup Language (XML) 1.0
 - For data portability
 - XML Schema (XSD) 1.0
 - For describing message data definitions
 - Simple Object Access Protocol (SOAP) 1.1
 - For platform-neutral exchanges of information in a distributed environment
 - WS-I Basic Profile Version 1.1
 - Ensures greater interoperability

7 Copyright © 2004-2008 Active Endpoints, Inc.



The use of standard languages and protocols and the leveraging of common skillsets means that your development team comes to the table with at least some part of the basics of BPEL already mastered. There will certainly be some new material to learn, but a great deal of what we use comes from other – and probably familiar - areas. ActiveVOS Designer uses these standards-based technologies:

BPEL (v1.1 BPEL4WS and v2.0 WS-BPEL) describes web service orchestration flows

WSDL (v1.1) defines the public interface to a web service

XML (v1.0) defines the language BPEL is based on and supports data portability

XML (v1.0) Schema describes the data definitions used by our messages

SOAP (v1.1) is a platform neutral transport mechanism that is commonly used get data in and out of a process.

SOAP is not part of BPEL, and is not required, (i.e., you can use something like JMS) but it is commonly used by web services invoked by a BPEL process.

WS-I Basic Profile (v1.1) ensures wider interoperability.

Unit Objectives

- At the conclusion of this unit, you will be familiar with:
 - ✓ Active Endpoints and the ActiveVOS family of products
 - Business Process Execution Language (BPEL) features and benefits
 - Process definition that you will build during the labs

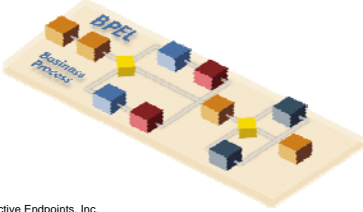
8 Copyright © 2004-2008 Active Endpoints, Inc.



We've looked at Active Endpoints and the ActiveVOS product line, now let's take a look at the BPEL language itself.

Business Process Execution Language (BPEL)


- BPEL is the Web Services Orchestration standard from OASIS
 - bee•pel', beep'•uhl, bip'•uhl
- An XML-based grammar for describing the logic to coordinate and control Web services in a business process



9 Copyright © 2004-2008 Active Endpoints, Inc.

BPEL Historical Timeline

- Dec 2000**
Microsoft publishes XLANG
- March 2001**
IBM publishes WSFL
- July 2002**
IBM and Microsoft converge WSFL & XLANG into BPELWS 1.0
- March 2003**
BPEL4WS is submitted to OASIS
- May 2003**
OASIS publishes BPEL4WS 1.1
- 1stH 2007**
WS-BPEL 2.0 released



Pronunciation – we use “beep-uhl” (rhymes with people) in house, but any pronunciation is acceptable. BPEL is an XML-based grammar for describing the logic to coordinate and control web services in a business process. The roots of BPEL begin in December of 2000, when Microsoft published XLANG. This was followed in March of 2001 when IBM published WSFL. BPEL itself was first released on July 2, 2002, as BPEL4WS v1.0, jointly by BEA Systems, IBM and Microsoft. BPEL4WS v1.0 merged the flat-graph process definition approach of IBM’s Web Services Flow Language (WSFL) with the structural constructs approach of Microsoft’s XLANG. In May of 2003, BPEL4WS v1.1 was released with a set of revisions, and that was the version of the specification that was submitted to the OASIS organization. OASIS just finished the standardized version of the language, known as WS-BPEL 2.0., and it was ratified as an OASIS standard in April of 2007.

Concerns Addressed by BPEL

- BPEL processes can automate both simple and complex interactions between Web services
 - Supporting long running business transactions
 - Correlating message exchanges
 - Implementing parallel processing of activities
 - Mapping data between partner interactions
 - Providing consistent exception and recovery handling

10 Copyright © 2004-2008 Active Endpoints, Inc.



So, why the new language? Why do we need BPEL?

BPEL provides:

- Support for web services relationships and interactions that are engaged in both short and long-term business transactions.
- Message exchange correlation for long running message exchanges, not just over a minute or two, but over days, weeks or months.
- Implementation for the parallel processing of activities, which allows us to execute non-dependent actions concurrently to improve process performance.
- For the mapping of data between partner interactions, so that we can, for example, take the result from one web service and use it to invoke another web service.
- Consistent Exception & Recovery handling for our deployed business processes.

Benefits of BPEL

- Industry-wide language for expressing business processes
 - Leverage a common skill set and language
- Designed to fit naturally into the Web services stack
 - Is 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

11 Copyright © 2004-2008 Active Endpoints, Inc.



What do we get out of using this language? What are some of the benefits of BPEL?

BPEL is SOA (Service Oriented Architecture) compliant, meaning that it is based on web services, which are the set of protocols by which such services can be published, discovered and used in a technology neutral, standard form.

BPEL allows us to leverage existing standards and skill sets, and a common language, as we mentioned earlier.

BPEL deployed orchestrations are web services themselves, and therefore fit naturally into the existing Web Services stack.

BPEL is expressed entirely in XML, uses and extends the WSDL 1.1 definitions and uses XML Schema 1.0 for the data model.

BPEL is platform and vendor agnostic, and so a BPEL process will run on any engine that is BPEL-compliant.

BPEL processes are interoperable between and among existing/running web services because they are themselves web services.

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 this stateful interaction model

12 Copyright © 2004-2008 Active Endpoints, Inc.



So why do we need BPEL? What problems does it solve?

Web services use a "stateless" interaction model, meaning that the conversation between the two services has no context and no persistent data set. Most business processes require a better interaction model to handle conversations that can last hours, days, even weeks or months.

BPEL provides this "stateful" interaction model to support your business processes.

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

The diagram consists of three blue rectangular boxes. The top box is labeled 'BPEL 2.0'. Below it are two smaller boxes: 'WSDL 1.1' on the left and 'BPEL-defined WSDL extensions' on the right. This visualizes BPEL 2.0 as a layer that builds upon both the standard WSDL 1.1 and its own specific extensions.

13 Copyright © 2004-2008 Active Endpoints, Inc.

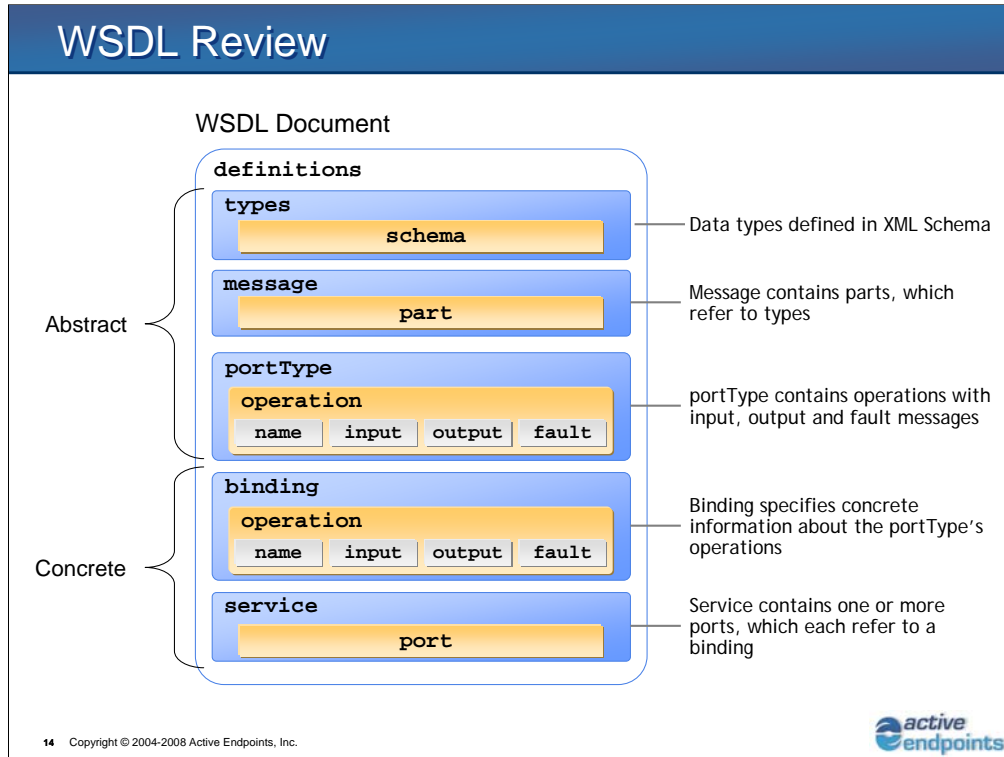
So, what is BPEL's relationship with WSDL?

WSDL defines the operations (i.e., the methods) and the defined message types that are used by a web service.

WS-BPEL coordinates the actions of one or more WSDL operations, as a means of orchestrating the interaction of web services.

BPEL specifies extensions to WSDL that allow long-running, asynchronous business processes.

Remember that when we create BPEL processes we are not integrating applications. BPEL deals with "black box" services and their exposed APIs. We do not care about the underlying implementations because we don't need to. This is how BPEL turns the promise of a Service Oriented Architecture into a reality.



On the next two slides is a brief review of WSDL files and their contents.

We see on this slide that a WSDL file can be divided between the abstract and the concrete sections.

The Abstract section includes the data and process type definitions:

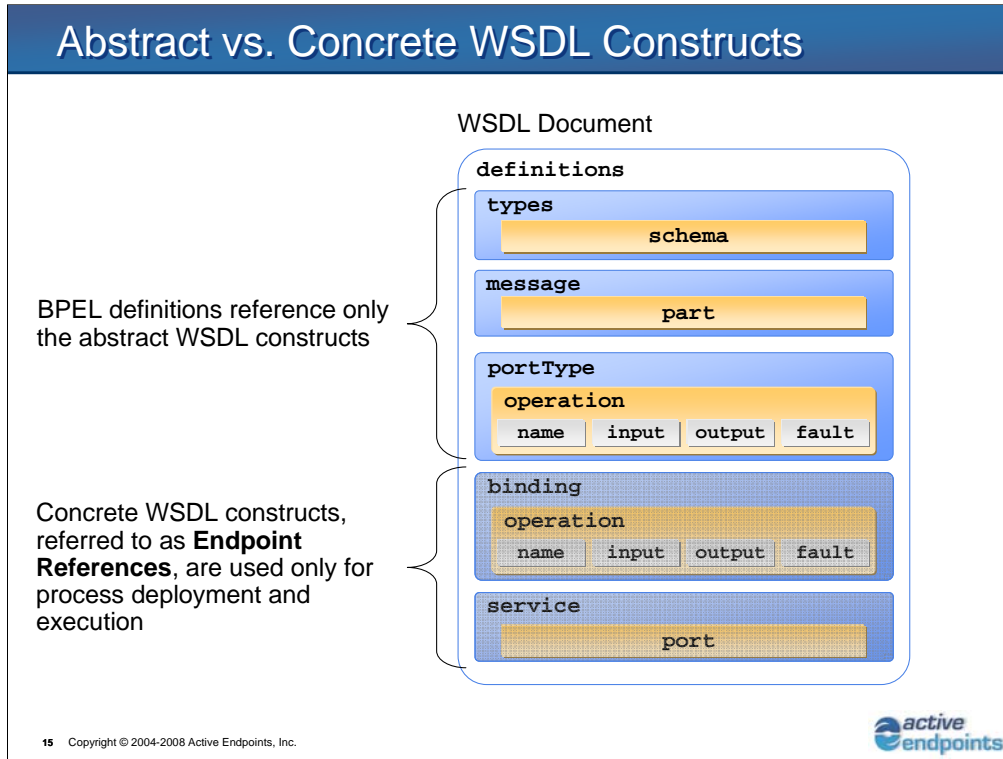
- The Data Types defined in the schema (i.e., string, integer, boolean, etc.)
- The MessageType definitions, which use the schema definitions and that move data between a process and its partner services.
- The PortType definitions, which include the operations that define the port's functionality and their input, output and fault messages.

These abstractions, taken together, define the web service's design.

The Concrete section includes the deployment & runtime definitions:

- The actual Ports and their Services
- The Binding information, which includes the transport mechanisms and the payloads they carry.
- The Endpoint References, which are the service's URL(s).

These concretes, taken together, define how the web service is actually implemented at execution time.



So, to summarize, BPEL is concerned with abstract during the design phase and with the concrete at the time of deployment and execution. BPEL definitions are completely abstract during design and refer to the WSDL definitions. Concrete WSDL constructs – such as the Binding and Services - exist only during the actual deployment and execution of a process.

Unit Objectives

- At the conclusion of this unit, you will be familiar with:
 - ✓ Active Endpoints and the ActiveVOS family of products
 - ✓ Business Process Execution Language (BPEL) features and benefits
 - Process definition that you will build during the labs

16 Copyright © 2004-2008 Active Endpoints, Inc.



Now that we've looked at Active Endpoints and our products, and at BPEL and its features and benefits, let's turn our attention to the process that you will be creating during your lab work.

Lab Application Overview – Order Process

```

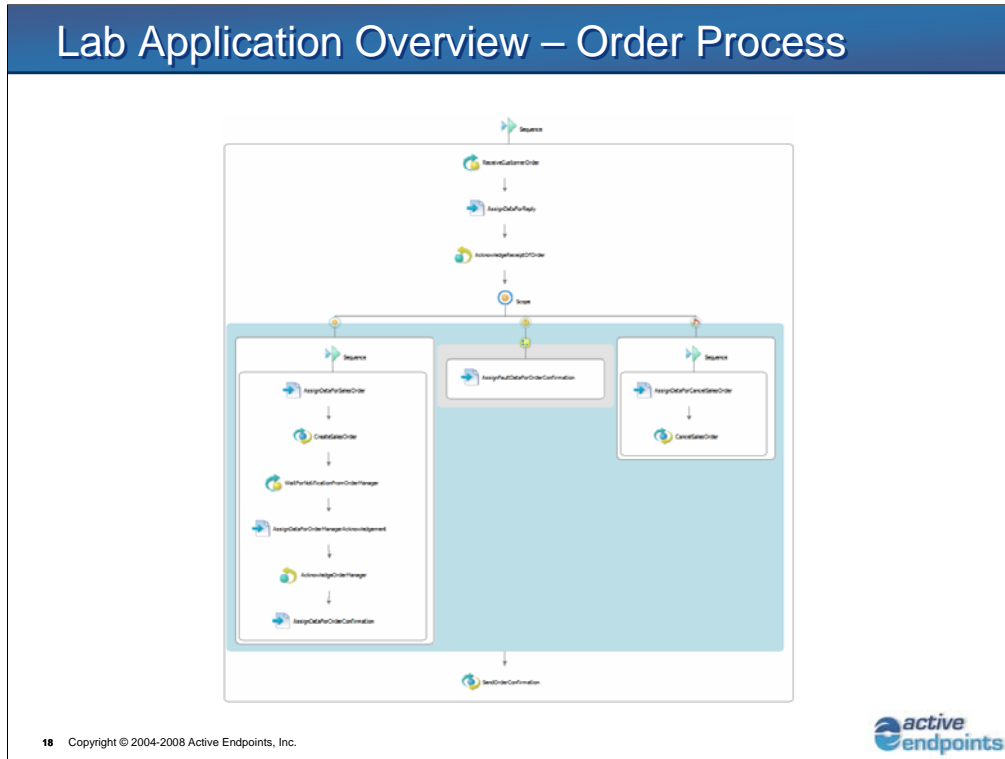
<process name="OrderProcess" suppressJoinFailure="yes" ...>
  <import importType=http://schemas.xmlsoap.org/wsdl/
    location="resources/wsdl/BPELRetailer.wsdl"
    namespace="http://docs.active-endpoints.com/activebpel/sample/... " />
  <partnerLinks>
    <partnerLink myRole="Seller" name="Customer"
      partnerLinkType="brws:PurchasingPLT" partnerRole="Buyer" />
  </partnerLinks>
  <variables>
    <variable messageType="rws:SubmitOrderRequest" name="SubmitOrderRequest" />
  </variables>
  <correlationSets>
    <correlationSet name="CustomerCorrelationSet"
      properties="brws:poNumber brws:customerNumber" />
  </correlationSets>
  <eventHandlers>
    <onEvent messageType="rws:CheckOrderStatusRequest" operation="CheckOrderStatus" ... >
      <scope ... </scope>
    </onEvent>
  </eventHandlers>
  <sequence>
    <receive createInstance="yes" name="ReceiveCustomerOrder" operation="SubmitOrder"
      partnerLink="Customer" portType="rws:RetailerPortType"
      variable="SubmitOrderRequest" />
    ...
  </sequence>
</process>

```

17 Copyright © 2004-2008 Active Endpoints, Inc.



This training will help you build a process for Order Processing that will invoke multiple web services, provided by our business partners. The final process we construct will be end up being deployed as a web service itself. We will use ActiveVOS Designer to create the process, and it will automatically create the BPEL code for us. Shown here is a portion of the BPEL code for this process.



This is a typical BPEL process diagram, showing a graphical representation of the BPEL code we saw on the last slide. We will use most of the basic and structured BPEL activities during the course of our labs, and will also use a good part of the functionality of the ActiveVOS Designer tool.

Unit Summary

- Now you are familiar with:
 - Active Endpoints and the ActiveVOS family of products
 - Business Process Execution Language (BPEL) features and benefits
 - Process definition that you will build during the labs

19 Copyright © 2004-2008 Active Endpoints, Inc.



This completes the Introduction to Active Endpoints, our product line and the BPEL training. The next unit deals with the concept of Partner Interaction.