



Software Group

# UML Modelling of Automated Business Processes with a Mapping to BPEL4WS

Tracy Gardner  
IBM UK Laboratories, Hursley Park

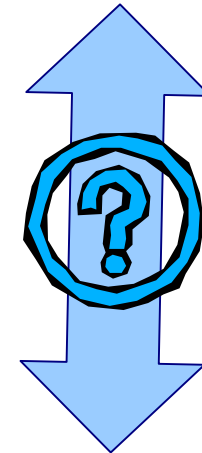
**First European Workshop on  
Web Services and Object Orientation,  
ECOOP 2003**

© 2003 IBM Corporation

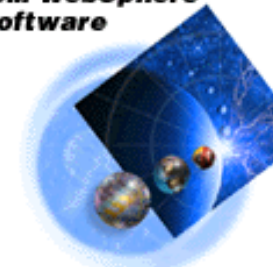
## UML to BPEL4WS

- UML is a widely used, standard modelling language for software design with a visual notation.
- BPEL4WS is a language for specifying business processes which can be executed on a BPEL4WS runtime
- **Goal: Support automated mapping from (a profile of) UML to BPEL4WS**

A profile is a customization of UML for modelling in a particular context.



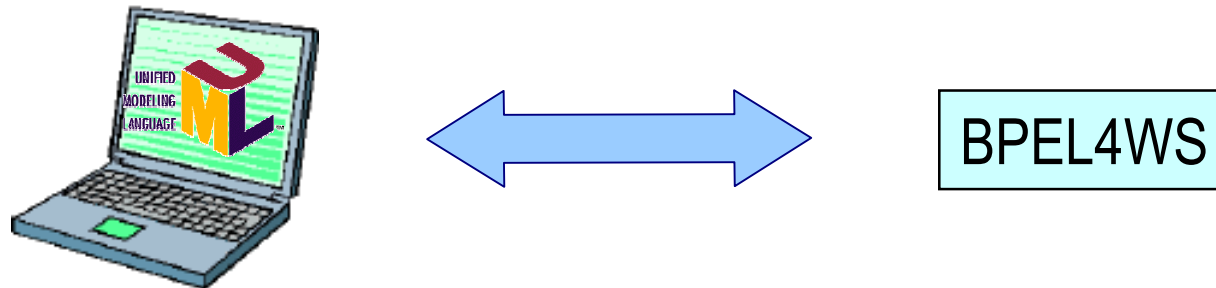
IBM WebSphere  
Software



**BPEL4WS,  
WSDL, XSD**

## Target Audience

- The profile subsets and extends UML 1.4 to allow the design of UML models that can be mapped to BPEL4WS
- The profile is aimed at:
  - UML-literate IT architects/developers who need to design systems that will be deployed to a BPEL4WS runtime



## BPEL4WS

- The Business Process Execution Language for Web Services (BPEL4WS) provides an XML notation and semantics for specifying business process behavior based on Web Services.
- A BPEL4WS process is defined in terms of its interactions with partners. A partner may provide services to the process, require services from the process, or participate in a two-way interaction with the process.

# A BPEL 'program'

state

interaction  
points

behaviour

```
<process name="purchaseOrderProcess" ...>

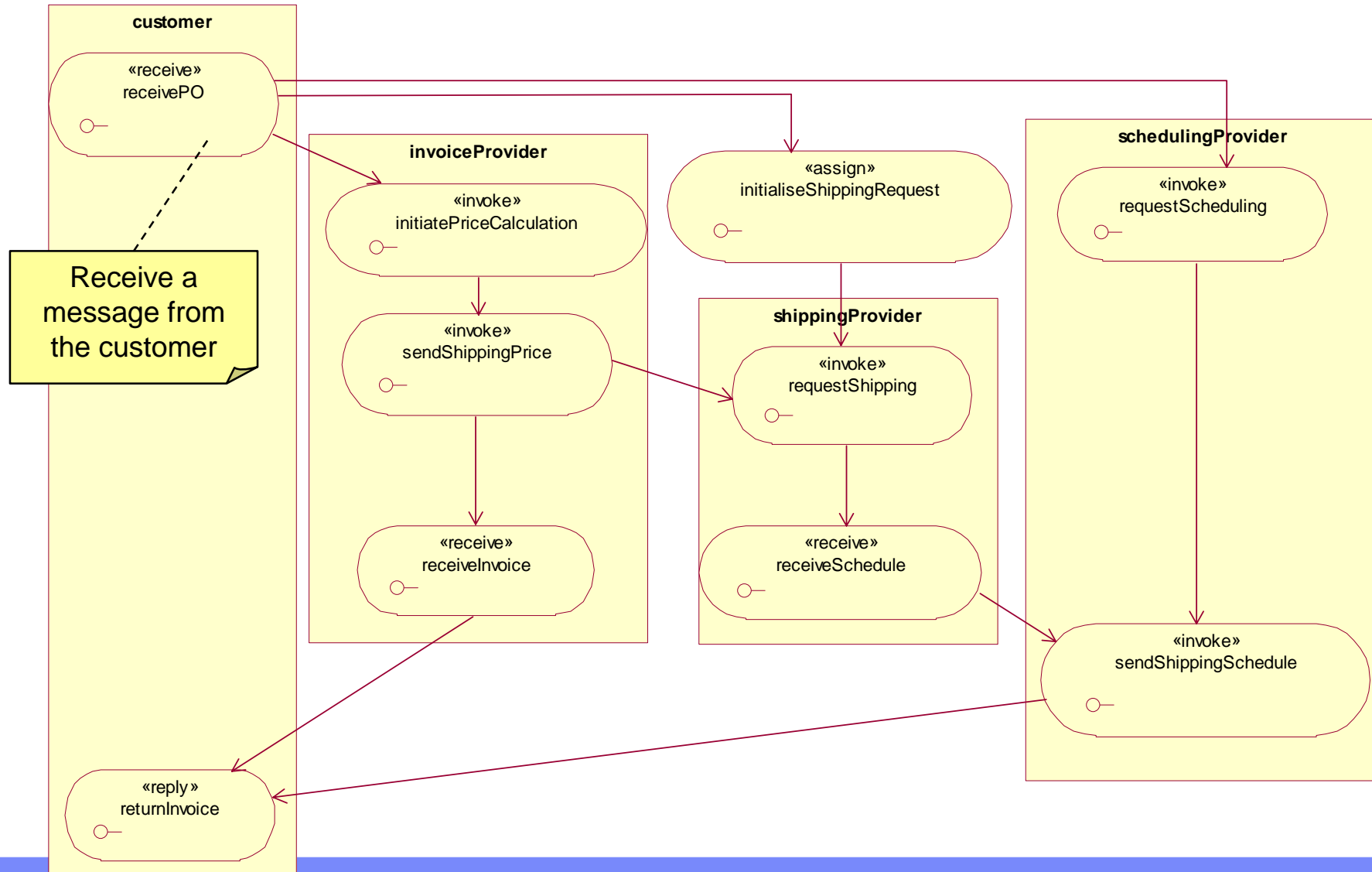
  <containers>
    <container name="PO" messageType="lns:POMessage"/>
    <container name="Invoice" messageType="lns:InvMessage"/>
    ...
  </containers>

  <partners>
    <partner name="customer" serviceLinkType="lns:purchaseLT"
      myRole="purchaseService"/>
    ...
  </partners>

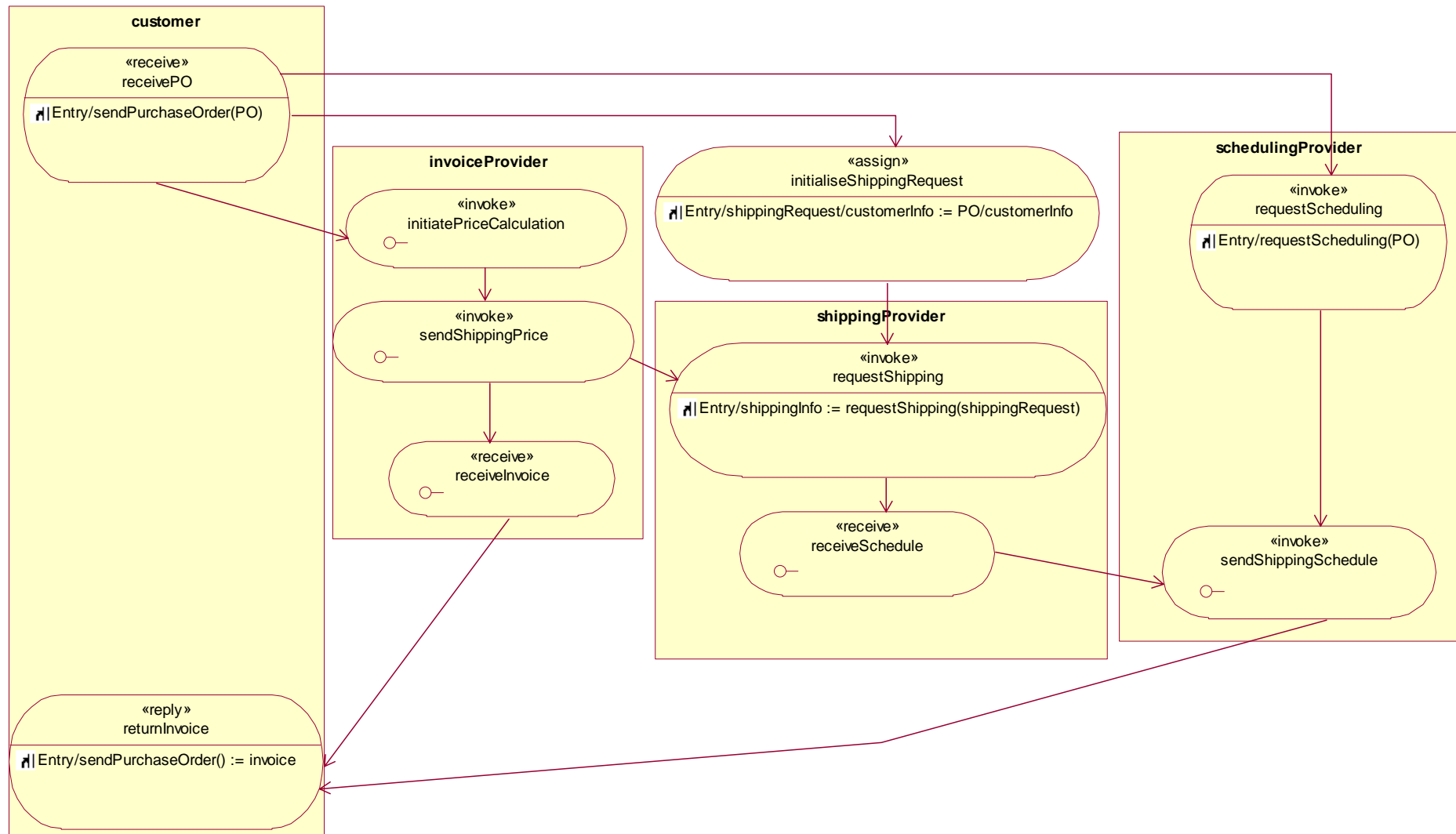
  <sequence>
    <receive partner="customer"
      portType="lns:purchaseOrderPT"
      operation="sendPurchaseOrder"
      container="PO">
    </receive>
    ...
    <reply partner="customer" portType="lns:purchasePT"
      operation="sendPurchaseOrder"
      container="Invoice"/>
  </sequence>

</process>
```

# Purchase Order Process

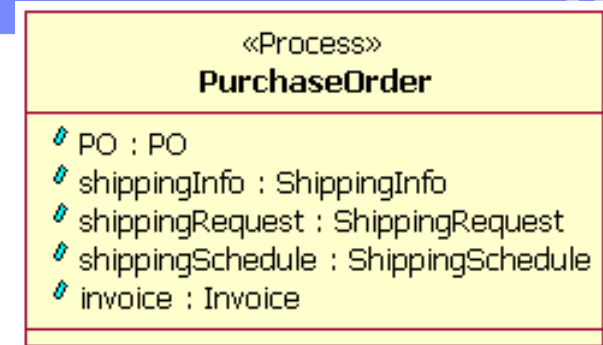


# Purchase Order Process – Detailed Behaviour

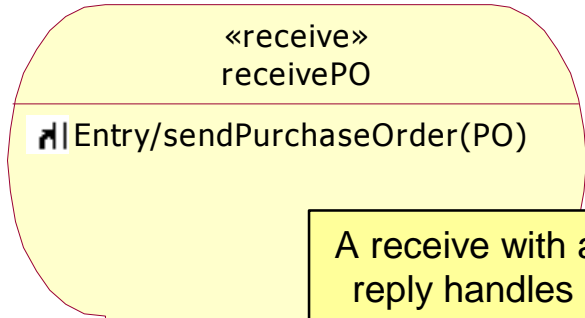


# Basic Interactions

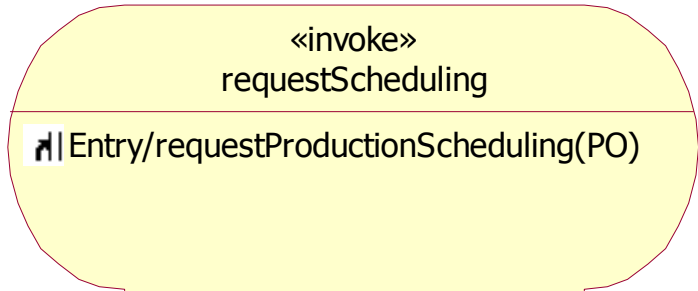
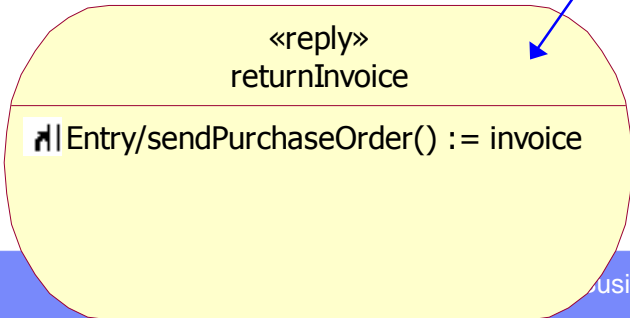
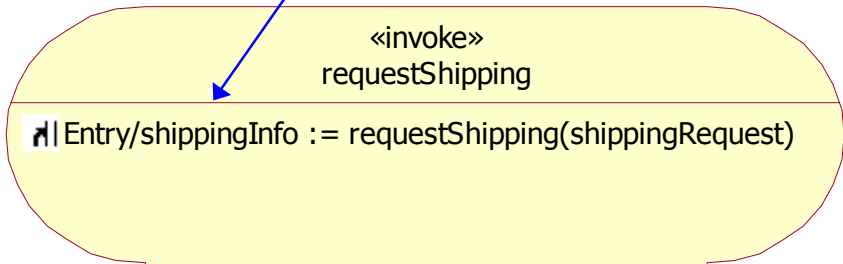
- Action syntax based on XPATH which BPEL uses as its default expression language



Synchronous requests have a response



A receive with a corresponding reply handles a synchronous request





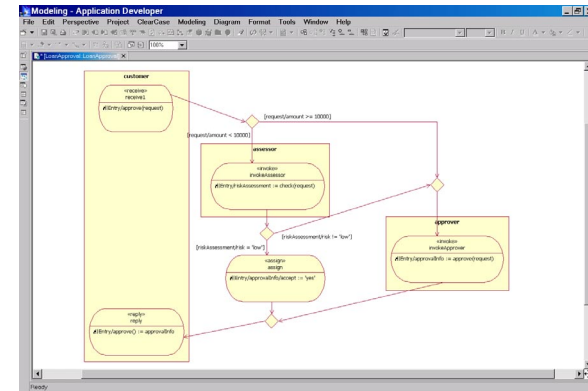
## BPEL Mapping Overview

<<process>> class	BPEL process definition
Activity graph on a <<process>> class	BPEL activity hierarchy
<<port>> associations	BPEL partner declarations
<<process>> class attributes	BPEL containers
Hierarchical structure and control flow	BPEL sequence and flow activities
Decision nodes	BPEL switch activities and transition conditions
<<receive>>, <<reply>>, <<invoke>> activities	BPEL receive, reply, invoke activities
<<protocol>> package with <<role>> classes	BPEL service links types and roles

# UML to BPEL alphaWorks Demo

- Part of the Emerging Technologies Toolkit
- Available now
- Supports an end-to-end scenario: Loan Approval
- Version supporting full forward generation from user-defined models coming in July 2003

Rose/XDE



XMI

Model-Model  
Mapping

```

<?xml version="1.0" encoding="UTF-8"?>
<process abstractProcess="no" containerAccessSerializable="
enableInstanceCompensation="no" name="LoanApproval"
suppressJoinFailures="yes"
targetNamespace="http://www.bpel-examples.ibm.com/LoanA
xmlns="http://schemas.xmlsoap.org/ws/2002/07/business-pr
xmlns:LoanApprovalProtocols="http://www.bpel-examples.i
xmlns:LoanApprover="http://www.bpel-examples.ibm.com/Lo
xmlns:LoanAssessor="http://www.bpel-examples.ibm.com/Lo
xmlns:LoanDefinitions="http://www.bpel-examples.ibm.com
<partners>
  <partner name="Approver" name="customer" serviceI
  <partner name="assessor" partnerRole="Assessor" ser
</partners>
<containers>
  <container messageType="LoanDefinitions.CreditInfor
  <container messageType="LoanAssessor.RiskAssessmen
  <container messageType="LoanApprover.Approval" name
  <container messageType="LoanDefinitions.LoanRequest
</containers>
<faultHandlers>
  <catch faultContainer="error" faultName="LoanAssesse
  <reply container="error" faultName="invalidReque
  <name="throwInvalidRequest" operation="approve
  <partner="customer" postType="LoanApprover.L
</catch>
</faultHandlers>
<flow>
  <link name="receive_to_invokeApprover"/>
  
```

BPEL

BPWS4J  
Runtime

The top screenshot shows the 'Configure Processes' dialog in IBM Business Process Execution Language for Web Services Java Runtime. It displays the process name 'LoanApprovalServiceBP' and provides options to 'List' or 'Deploy' the process. The bottom screenshot shows the 'Launch Configurations' dialog, where the 'Main' tab is selected, showing the program arguments for running the process.

## Conclusions

- The experiences of modelling with UML can be applied to the development of systems that will be deployed using emerging web services standards
- It is possible to specify a profile of UML with sufficient detail that it can be translated automatically to a language such as BPEL4WS
- The approach provides an integration mechanism for multiple standards and specifications which need to be used to build a complex solution

This is especially relevant in business integration scenarios

## Resources and Contact Details

- UML to BPEL on IBM alphaWorks:  
<http://www.alphaworks.ibm.com/tech/ettk>
  - Profile definition
  - Example models
  - End-to-end demo
  
- BPEL4WS Specification:  
<http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>
- BPWS4J on IBM alphaWorks:  
<http://www.alphaworks.ibm.com/tech/bpws4j>
  
- Contact: Tracy Gardner <[tgardner@uk.ibm.com](mailto:tgardner@uk.ibm.com)>