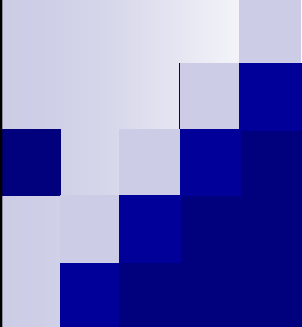


Universität  
Paderborn





# MeWA

## Chapter 10: Web Service Processes

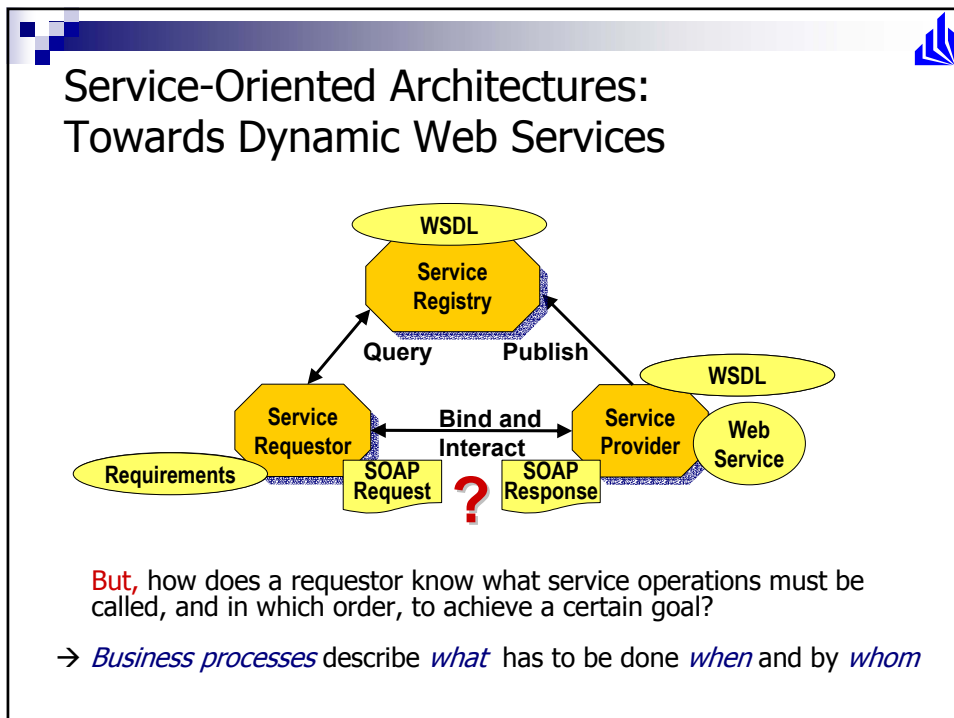
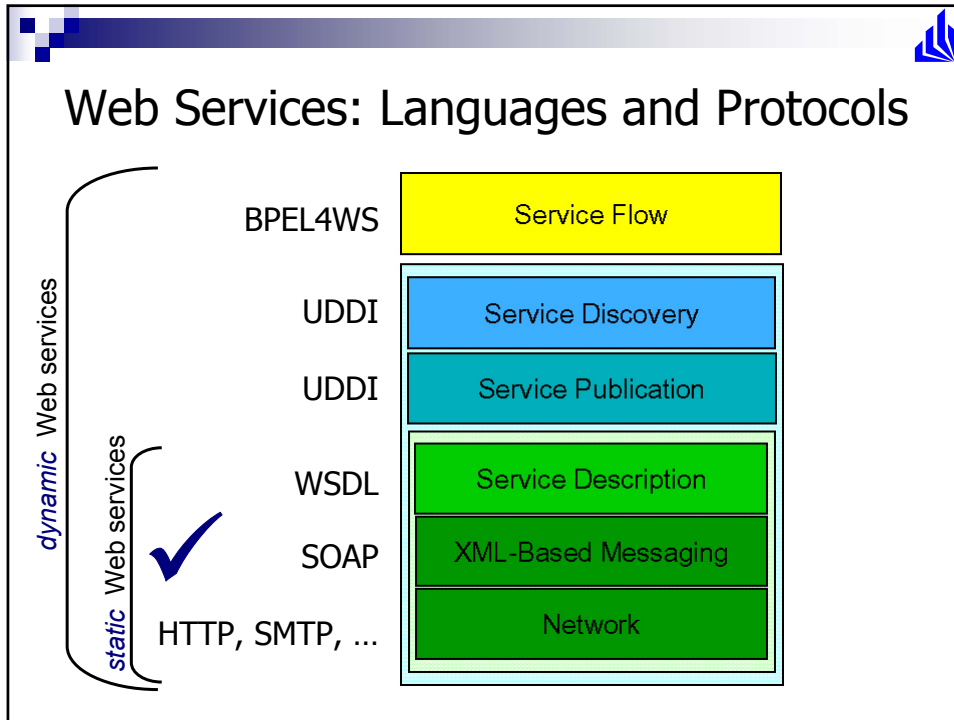
Dr Reiko Heckel

Data Base and  
Information Systems Group



## Outline of the Lecture

1. Introduction ✓
2. Interactive Web Applications ✓
3. Graph Transformation ✓
4. Service-Oriented Architectures ✓
5. Introduction to XML ✓
6. Modeling XML Languages ✓
7. Model-Based Data Integration ✓
8. Messaging and RPC ✓
9. Describing and Publishing Web Services ✓
10. Processes for Web Services



## Business Process

- is a set of logically related activities performed to achieve a defined business goal
    - has a customer (internal or external) who benefits from the process
    - has clearly defined start and end points
    - requires cooperation between organizational units or humans
  - is defined by answering the questions
    - *Who is performing a activity?*
    - *What activity are performed?*
    - *When is a activity performed?*
- Requester performs business process in cooperation with providers, combining the results of their processes.

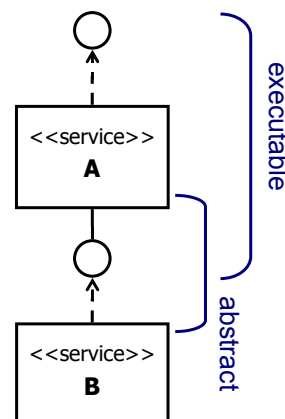
## Business Processes for Web Services

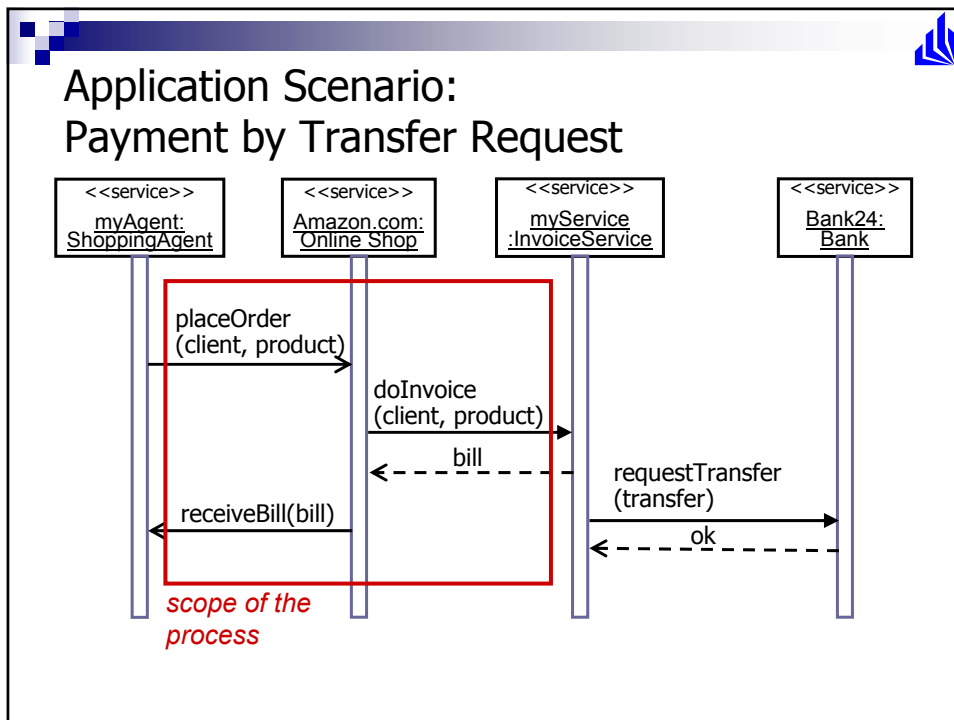
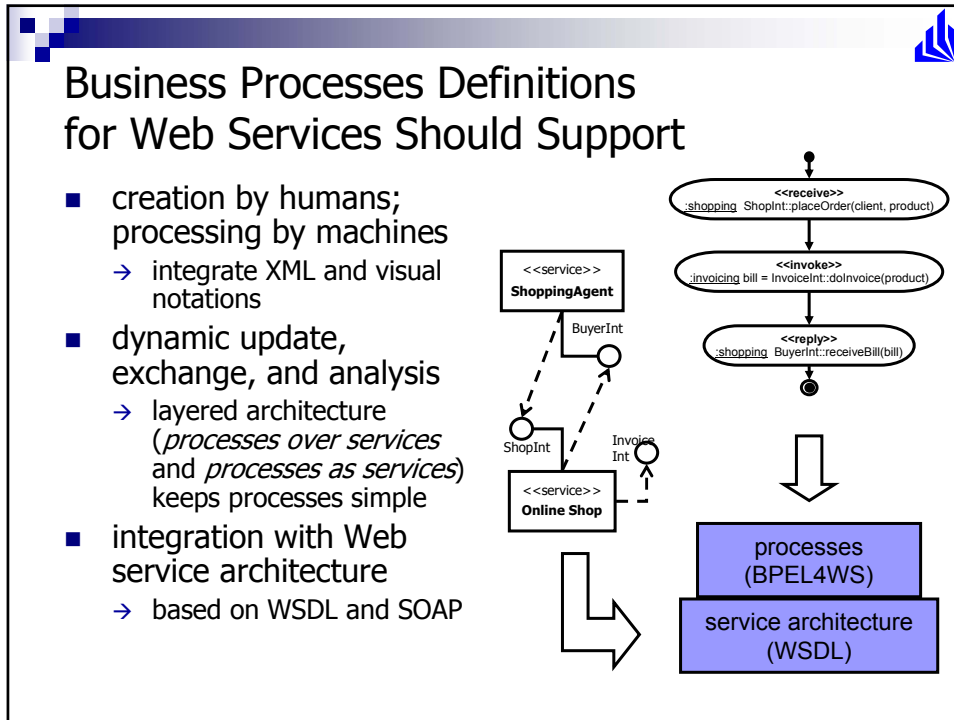
**Executable processes:**  
define implementation of provided service in terms of required ones

→internal view

**Abstract processes:**  
specify protocol that guide the interaction between (or the use of) services

→external view

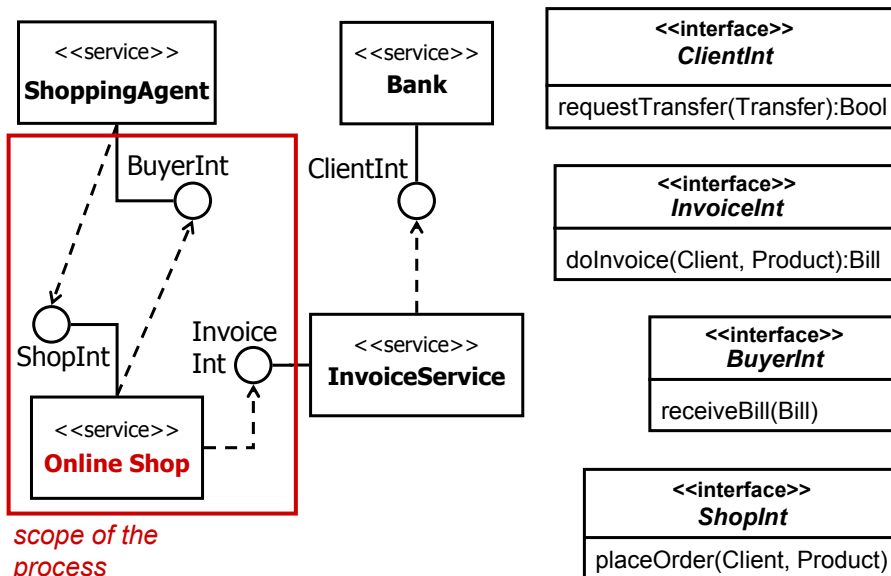


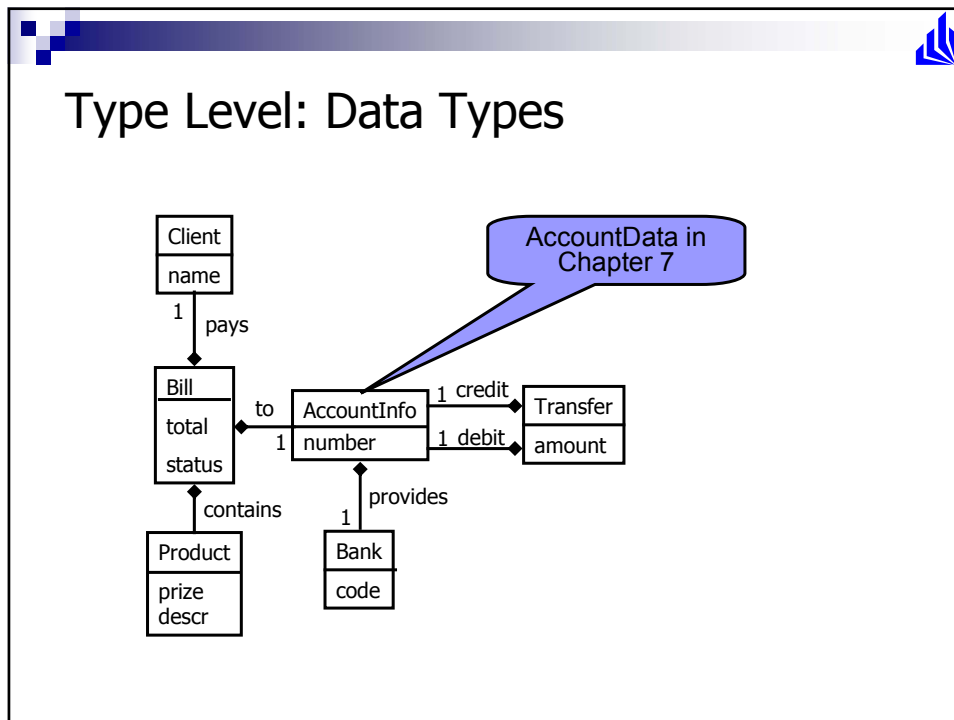


## What needs to be defined?

- architecture: type level
  - components and interfaces
  - data types
  - connectors
- architecture: prototypical instance level
  - component instances and connections
- process structure
  - activities and their dependencies

## Type Level: Components and Interfaces



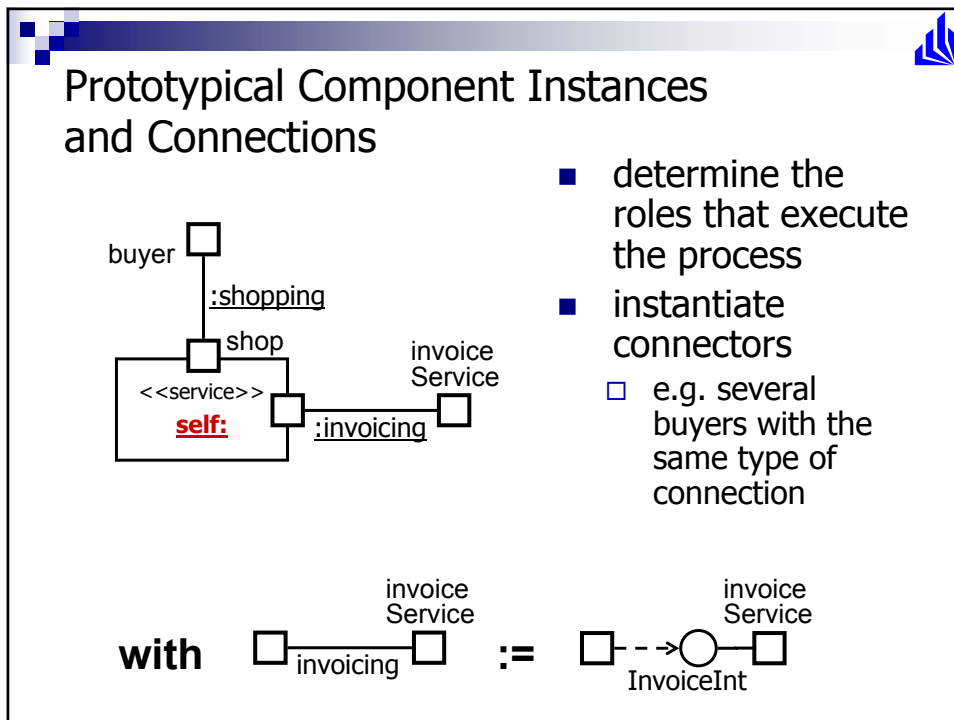
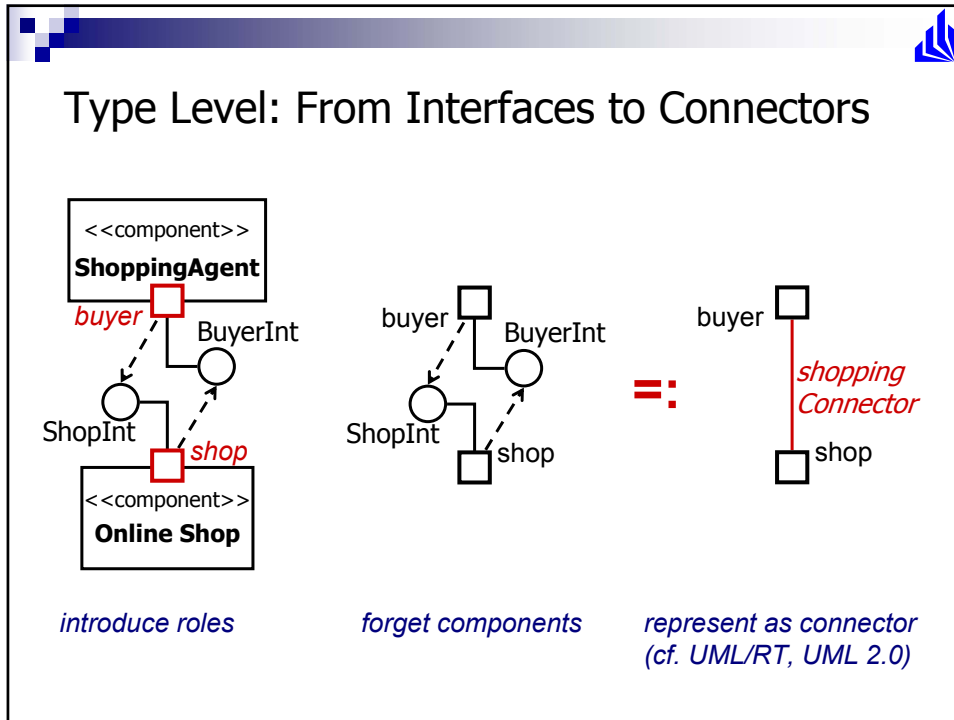


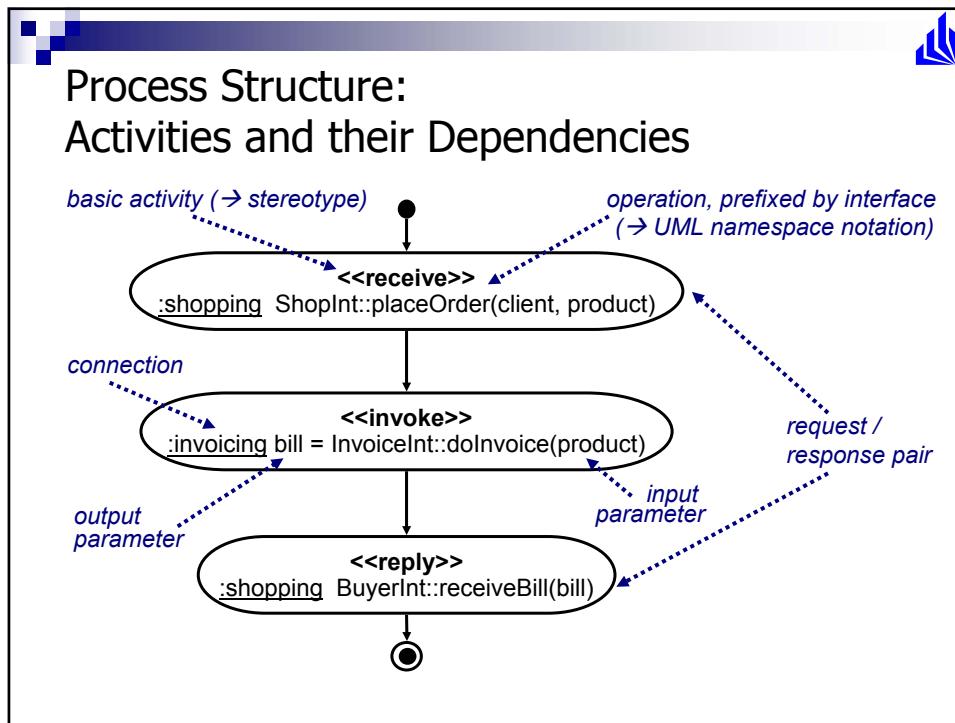
### Type Level: From Interfaces to Connectors

**Aim:** define processes independently of services to support reuse and flexible deployment

**Therefore:**

- components are replaced by *partner roles*
- complementary roles are combined to peer-to-peer *connections*
- connections are typed over *connectors* which define for each role the *interfaces* a partner playing this role must implement





- ### What needs to be defined?
- architecture: type level
    - components and interfaces
    - data types
    - connectors

WSDL
  - architecture: prototypical instance level
    - component instances and connections

BPEL4WS
  - process structure
    - activities and their dependencies

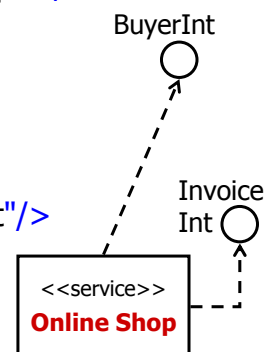
## Components and Interfaces: PortType provided by the Process

```
<portType name="ShopPortType">  
  <operation name="placeOrder">  
    <input message="ns:placeOrderInput"/>  
  </operation>  
</portType>
```



## Components and Interfaces: PortTypes required by the Process

```
<portType name="InvoicingPortType">  
  <operation name="doInvoice">  
    <input message="ns:doInvoiceInput"/>  
    <output message="ns:doInvoiceOutput"/>  
  </operation>  
</portType>  
  
<portType name="BuyerPortType">  
  <operation name="receiveBill">  
    <input message="ns:receiveBillInput"/>  
  </operation>  
</portType>
```



### Interfaces: I/O Message

```

<message name="placeOrderInput">
  <part name="client" type="ns:Client"/>
  <part name="product" type="ns:Product"/>
</message>

<message name="doInvoiceInput">
  <part name="client" type="ns:Client"/>
  <part name="product" type="ns:Product"/>
</message>
<message name="doInvoiceOutput">
  <part name="bill" type="ns:Bill"/>
</message>

<message name="receiveBillInput">
  <part name="bill" type="ns:Bill"/>
</message>

```

**<<interface>>  
ShopInt**

---

placeOrder(Client, Product)

**<<interface>>  
InvoiceInt**

---

doInvoice(Client, Product):Bill

**<<interface>>  
BuyerInt**

---

receiveBill(Bill)

### Data Types: XML Schema

```

<types>
  <xs:schema targetNamespace="http://example.com/namespace"
    xmlns:ns="http://example.com/namespace"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:complexType name="Transfer"> ... </xs:complexType>
    <xs:complexType name="AccountInfo"> ... </xs:complexType>
    <xs:complexType name="Bill">
      <xs:sequence>
        <xs:element name="pays" type="ns:Client"/>
        <xs:element name="contains" type="ns:Product"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="to" type="ns:AccountInfo"/>
        <xs:element name="Bill.status" type="xs:string"/>
        <xs:element name="Bill.total" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:schema>

```

## Data Types: XML Schema

```

...
<xs:complexType name="Client">
  <xs:sequence>
    <xs:element name="Client.name"
      type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Product">
  <xs:sequence>
    <xs:element name="Product.descr"
      type="xs:string"/>
    <xs:element name="Product.prize"
      type="xs:string"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
</types>
  
```

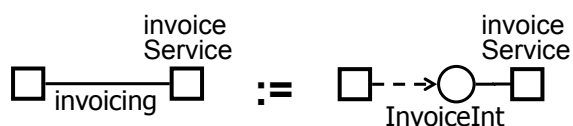
## Connectors: Partner Link Types

```

<partnerLinkType name="shoppingLinkType">
  <role name="buyer">
    <portType name="BuyerPortType"/>
  </role>
  <role name="shop">
    <portType name="ShopPortType"/>
  </role>
</partnerLinkType>
  
```

## Connectors: Partner Link Types

```
<partnerLinkType name="invoicingLinkType">  
  <role name="invoiceService">  
    <portType name="InvoicePortType"/>  
  </role>  
</partnerLinkType>
```



## What needs to be defined?

- architecture: type level
  - components and interfaces
  - data types
  - connectors
- architecture: prototypical instance level
  - component instances and connections
- process structure
  - activities and their dependencies

WSDL ✓

BPEL4WS

### Connections: Partner Links

```

    <partnerLinks>
      <partnerLink
        name="invoicingLink"
        partnerLinkType="invoicingLinkType"
        partnerRole="invoiceService"/>
      <partnerLink
        name="shoppingLink"
        partnerLinkType="shoppingLinkType"
        myRole="shop"
        partnerRole="buyer"/>
    </partnerLinks>
  
```

### Structure of the Process: Receive

```

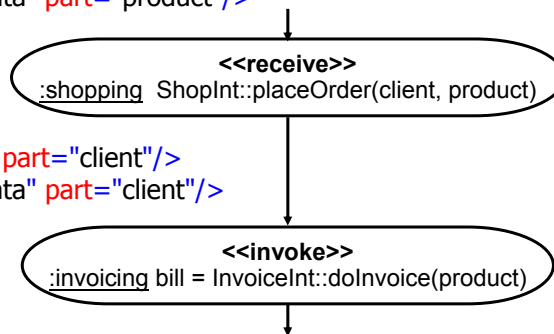
    <sequence>
      <receive name="receiveOrder"
        partnerLink="shoppingLink"
        portType="ShopPortType"
        operation="placeOrder"
        variable="order" createInstance="yes"/>
      ...
    
```

## Structure of the Process: Variable Assignments

```

...
<assign>
  <copy>
    <from variable="order" part="product"/>
    <to variable="invoiceData" part="product"/>
  </copy>
</assign>
<assign>
  <copy>
    <from variable="order" part="client"/>
    <to variable="invoiceData" part="client"/>
  </copy>
</assign>
...

```



## Variable Declarations

```

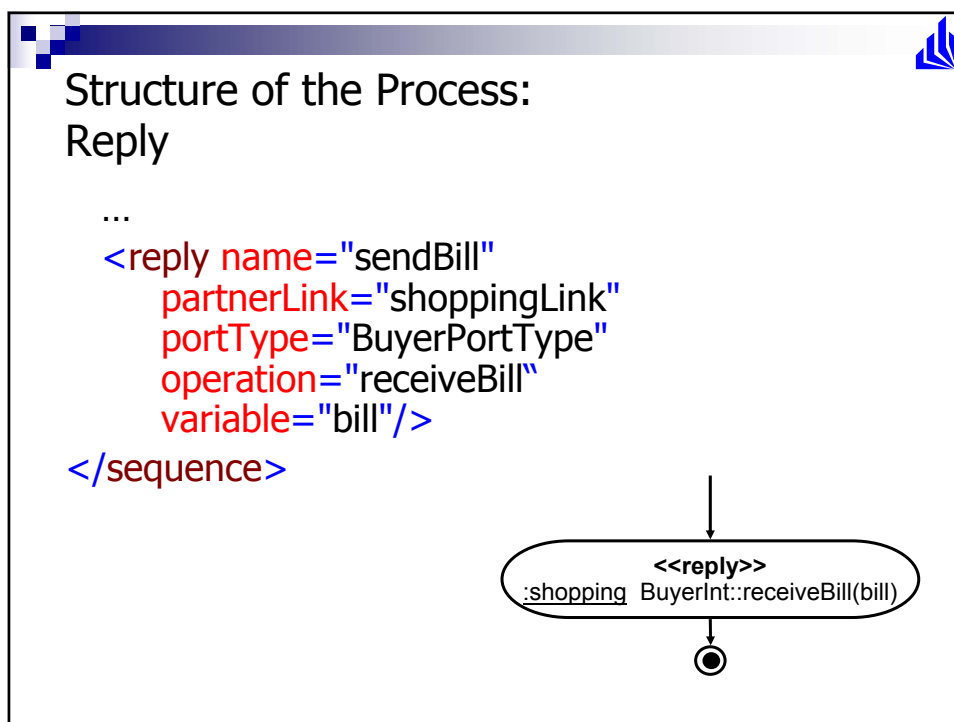
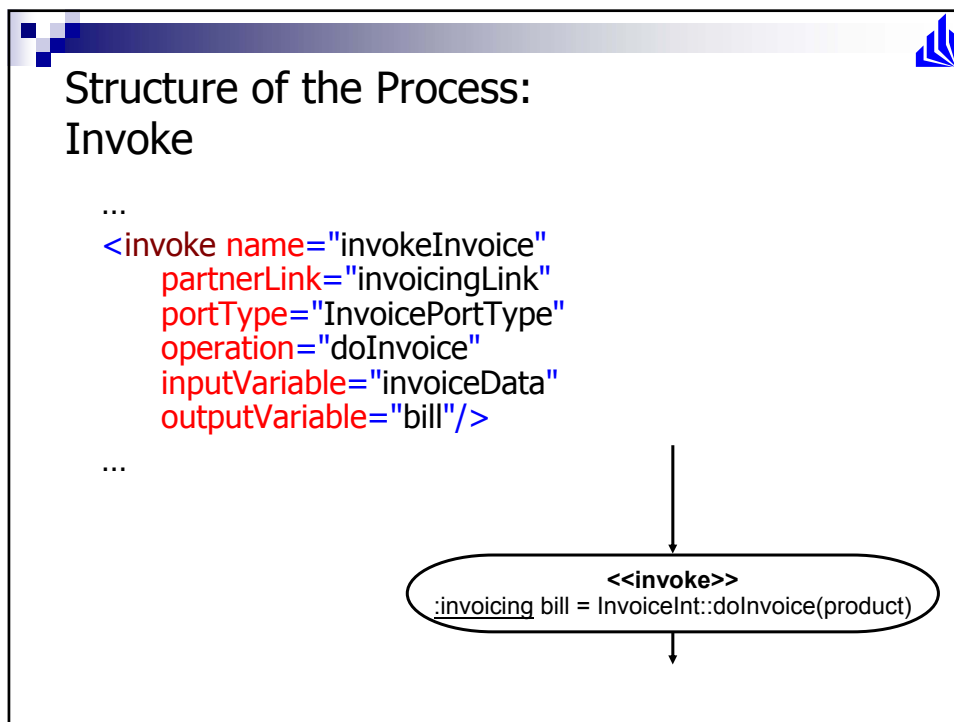
<variables>
  <variable name="order"
    messageType="ns:placeOrderInput"/>

  <variable name="invoiceData"
    messageType="ns:doInvoiceInput"/>

  <variable name="bill"
    messageType="ns:receiveBillInput"/>
</variables>

```

*Variables are not explicitly declared in UML,  
but can be collected from activity diagram.*



### What needs to be defined?

- architecture: type level
  - components and interfaces
  - data types
  - connectors
- architecture: prototypical instance level
  - component instances and connections
- process structure
  - activities and their dependencies

WSDL ✓

BPEL4WS ✓

### What else is there?

#### BPEL Syntax Overview

<i>&lt;process&gt;</i>	activity ::=
<i>&lt;partnerLinks&gt;?</i>	<i>&lt;receive&gt;</i>
<i>&lt;partners&gt;?</i>	<i>&lt;reply&gt;</i>
<i>&lt;variables&gt;?</i>	<throw>
<correlationSets>?	<i>&lt;invoke&gt;</i>
<faultHandlers>?	<i>&lt;assign&gt;</i>
<compensationHandler>?	<terminate>
<eventHandlers>?	<wait>
<i>activity</i>	<i>&lt;sequence&gt;</i>
<i>&lt;/process&gt;</i>	<switch>
	<while>
	<pick>
	<flow>
	<scope>
	<compensate>

## Support for Editing and Executing BPEL4WS

### IBM WebSphere and WebSphere Studio Application Developer:

- Web application server with BPEL4WS engine
- CASE tool with graphical process editor and plugin for BPEL4WS import and export

### BPWS4J Project: Business Processes for Web Services with Java

- validator and execution engine for Tomcat
- tree-based editor in Eclipse

## Reading

- Business Process Execution Language for Web Services, Version 1.1: Specification and other Resources  
<http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>
- Business Process with BPEL4WS: Learning BPEL4WS  
<http://www-106.ibm.com/developerworks/library/ws-bpelcol.html>  
Part 4: Creating processes with the BPWS4J editor  
<http://www-106.ibm.com/developerworks/webservices/library/ws-bpelcol4/>
- BPEL4WS import / export plugin  
<http://www7b.software.ibm.com/wsdd/downloads/bpel/bpel.html>  
for WebSphere Studio Application Developer  
[http://www7b.software.ibm.com/wsdd/library/techarticles/0302\\_flurry/flurry3.html?origin=wsss](http://www7b.software.ibm.com/wsdd/library/techarticles/0302_flurry/flurry3.html?origin=wsss)