

Η Γλώσσα WS-BPEL

Πρότυπο του OASIS

**Organization for the Advancement of Structured Information
Standards (OASIS)**

Εισαγωγή

- ▶ Η **WS-BPEL** (ή **BPEL**) είναι μια γλώσσα για την περιγραφή και εκτέλεση *επιχειρησιακών διαδικασιών* με χρήση Web Services, σε δύο επίπεδα
 - ▶ Αφαιρετικές Επιχειρησιακές Διαδικασίες
 - ▶ Εκτελέσιμες Επιχειρησιακές Διαδικασίες
- ▶ Οι διαδικασίες στη WS-BPEL εισάγουν και εξάγουν λειτουργικότητα χρησιμοποιώντας αποκλειστικά την τεχνολογία των **Web Services**

BPEL vs. Άλλες γλώσσες προγ/σμού

- ▶ Με τη BPEL μπορώ να γράψω προγράμματα που
 - χρησιμοποιούν web services (WS)
 - είναι WS
- ▶ Με άλλες γλώσσες μπορώ επίσης να καλέσω web services αλλά η BPEL είναι ειδικά φτιαγμένη γι αυτό. Όλο το I/O στη BPEL γίνεται μέσω WS endpoints
 - Π.χ. Σε μία standard γλώσσα προγραμματισμού εάν θες να τυπώσεις κάτι στην κονσόλα ή σ' ένα pop up window, χρησιμοποιείς μία I/O library. Στην BPEL, στέλνεις το string που θέλεις να τυπώσεις σε ένα WS (αυτό μπορεί μετά να χρησιμοποιεί μία I/O library για να το τυπώσει σ' ένα pop up window ή στην κονσόλα)

WS-BPEL και WSDL

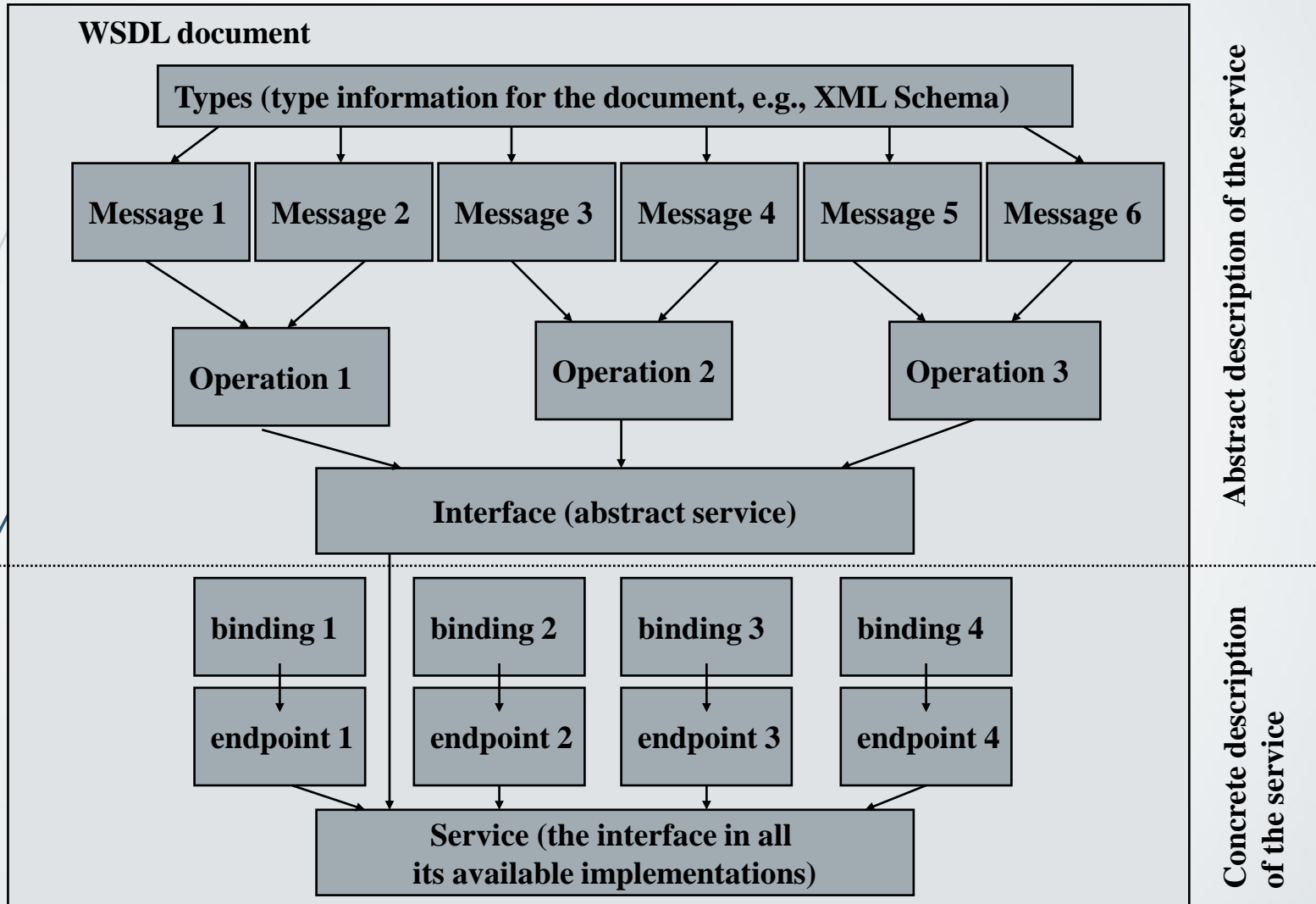
Τα WS-BPEL business processes

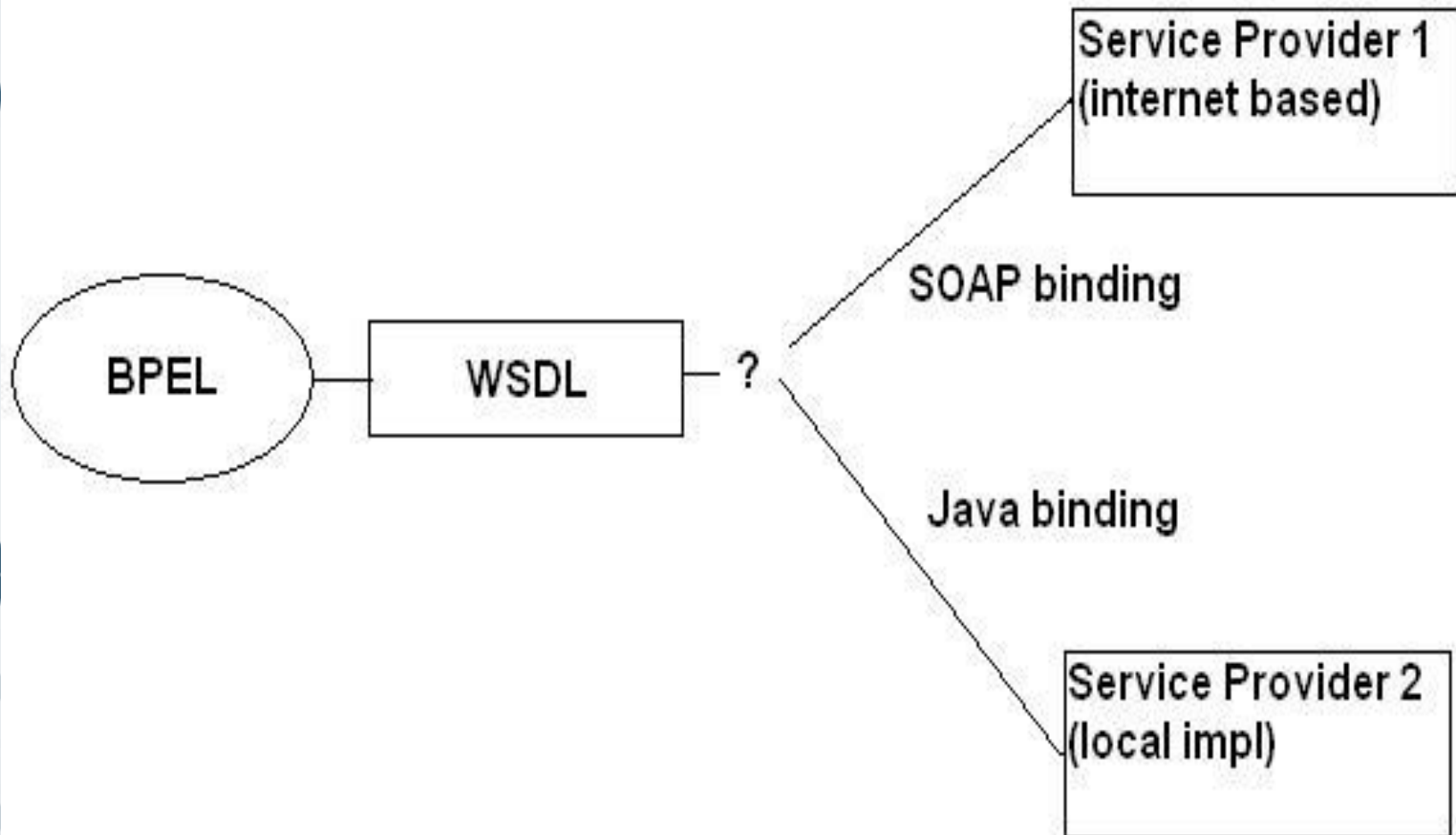
- ▶ **μοντελοποιούνται ως Web Services**
 - ▶ περιγράφονται με τη χρήση της WSDL
- ▶ **περιλαμβάνουν αλληλεπιδράσεις μεταξύ Web Services**
 - ▶ περιγράφονται με τη χρήση της WSDL

Θυμηθείτε ότι στη WSDL

- ▶ Το abstract μέρος είναι απλά περιγραφή του τι προσφέρει μια υπηρεσία – δεν δίνει πληροφορίες για το πως έχει υλοποιηθεί ή για το που βρίσκεται
- ▶ Για να είναι χρήσιμη μια περιγραφή, χρειάζεται να ξέρουμε και το binding (που παρέχεται στο concrete part).
 - ▶ Το binding ορίζει πως μπορείς να μιλήσεις με το WS
 - ▶ Μπορεί να υπάρχουν πολλές υλοποιήσεις για ένα WS και άρα πολλά bindings.
 - ▶ Το σύνηθες binding είναι SOAP/HTTP (access/transport protocol) για WS στο Internet.
 - ▶ Ένα άλλο binding είναι το Java binding

Στοιχεία της WSDL





Επικοινωνία WS client – BPEL process

- Ο Web Service client στέλνει request στο BPEL process
- Δημιουργείται ένα στιγμιότυπο του BPEL process
- Ο Client επιδρά με το στιγμιότυπο του BPEL process (αν χρειάζεται), μέχρι να ολοκληρωθεί η εκτέλεση

Σύντομη Επισκόπηση

WS-BPEL

WS-Addressing*

WSDL

XML Schema, XPath

- ▶ Η WS-BPEL είναι προϊόν της συγχώνευσης των
 - ▶ **XLANG** (Microsoft)
 - ▶ **WSFL** (IBM)
- ▶ Βασίζεται σε μια σειρά από *XML specifications*, τα οποία και χρησιμοποιεί

* WS-Addressing χρησιμοποιείται για τη διευθυνσιοδότηση των Web Services.

Η BPEL είναι μια XML γλώσσα προγραμματισμού

Έχει τρία βασικά συστατικά:

- ▶ Programming Logic (BPEL)
- ▶ Data types -> XSD (XML schema definition)
- ▶ Input/Output -> WSDL

Για παράδειγμα στο Hello world πρόγραμμα:

- ▶ Το XSD καθορίζει τα data types (το string type που θα αποθηκεύσει το Hello World)
- ▶ Το WSDL καθορίζει το service που θα τυπώσει το string
- ▶ Η BPEL τα βάζει όλα μαζί για να δημιουργηθεί και να τυπωθεί το string

The import statement

```
<import importType="http://schemas.xmlsoap.org/wsdl/"  
        location="../../test_bucket/service_libraries/tptp_EnginePrinterPort.wsdl"  
  
        namespace="http://www.eclipse.org/tptp/choreography/2004/engine/Print" />
```

- Εισάγει WSDL και XSD αρχεία
 - Δηλ. οι τύποι δεδομένων που χρησιμοποιούνται και τα endpoint descriptions ορίζονται σε ξεχωριστά αρχεία
- Πλεονεκτήματα: επαναχρησιμοποίηση σε πολλά BPEL αρχεία
 - Σημ: τα WSDL αρχεία επίσης κάνουν import XSD αρχεία

Απαιτήσεις εκτέλεσης

- ▶ Για το **deployment** και την **εκτέλεση ενός business process** σε κάποια **μηχανή εκτέλεσης WS-BPEL** χρειάζονται να υπάρχουν τα εξής:
 - ▶ Το **WS-BPEL document** που περιλαμβάνει τον ορισμό του business process
 - ▶ Το **WSDL document** που περιγράφει το WS interface του business process
 - ▶ Τα **WSDL documents** που περιγράφουν τα WS interfaces των Web services που χρησιμοποιούνται από το business process

Processes και Partners (1)

- ▶ Η BPEL παρέχει και καταναλώνει WSDL services.
- ▶ Στη BPEL ονομάζουμε **partners**
 - ▶ τους clients που καλούν το BPEL process ή
 - ▶ τα εξωτερικά web services που καλούνται από το process
- ▶ Η σύνδεση μεταξύ ενός partner και της BPEL process ονομάζεται **partner link**.
- ▶ Επομένως ένα BPEL process έχει τουλάχιστον έναν partner κι ένα partner link
 - ▶ Ποιό είναι αυτό;

Processes και Partners (2)

- ▶ Ένα BPEL *process* περιγράφει την **ενορχήστρωση** των αλληλεπιδράσεων με διάφορους **partners**
- ▶ Καθένας από τους συμμετέχοντες *partners* παρέχει ένα ή περισσότερα *port types* (service interfaces), που μοντελοποιούνται ως **partner links**
- ▶ Κάθε *partner link* χαρακτηρίζεται από ένα **partner link type**
- ▶ Κάθε *partner link type* ορίζει
 - ▶ Πως δύο WSDL-defined partners μπορούν να επικοινωνούν μεταξύ τους
 - ▶ Τι προσφέρει ο καθένας τους (δηλ. το ρόλο του καθένα)
- ▶ Κάθε **role** προσδιορίζει το **WSDL portType** εκείνο που το *partner link* ή το *process* παρέχουν για την επίτευξη της αλληλεπίδρασης

Επικοινωνία BPEL με Web services

- ▶ Ένα BPEL process επικοινωνεί με ένα WSDL-defined web service με 3 ξεχωριστούς τρόπους:
 - ▶ Το web service μπορεί να καλέσει τη BPEL process
 - ▶ Το web service μπορεί να κληθεί από τη BPEL process
 - ▶ Το web service μπορεί να έχει και τους δύο ρόλους, δηλαδή μπορεί να κληθεί από τη BPEL process και με τη σειρά μπορεί να καλέσει το BPEL process

Δομή ενός Business Process

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

```
  <partnerLinks>
```

```
    <!-- The declaration of partner links -->
```

```
  </partnerLinks>
```

```
  <variables>
```

```
    <!-- The declaration of variables -->
```

```
  </variables>
```

```
  <sequence>
```

```
    <!-- The definition of the BPEL business process main body -->
```

```
  </sequence>
```

```
</process>
```


Δομή ενός Business Process

```
<process name="ncname" targetNamespace="uri"  
  queryLanguage="anyURI"?  
  expressionLanguage="anyURI"?  
  suppressJoinFailure="yes|no"?  
  enableInstanceCompensation="yes|no"?  
  abstractProcess="yes|no"?  
  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/">
```

```
<partnerLinks>?  
  <partnerLink name="ncname" partnerLinkType="qname"  
    myRole="ncname"? partnerRole="ncname"?>+  
  </partnerLink>  
</partnerLinks>
```

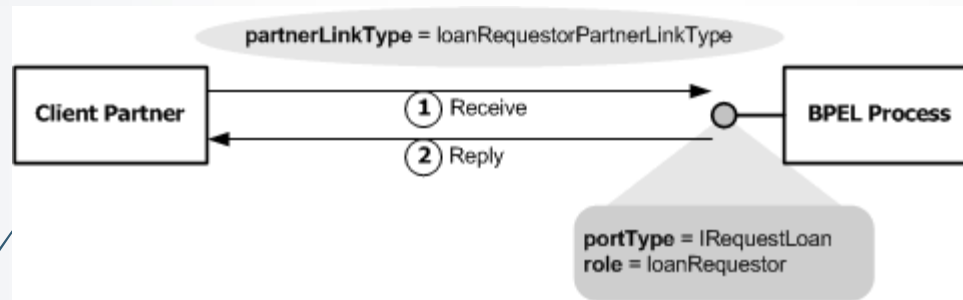
```
<partners>?  
  <partner name="ncname">+  
    <partnerLink name="ncname"/>+  
  </partner>  
</partners>
```

```
<variables>?  
  <variable name="ncname" messageType="qname"?
```

Παράδειγμα Επικοινωνίας BPEL με Web Services

Scenario 1: WSDL-defined web services can invoke a BPEL process

Το BPEL process παρέχει το *IRequestLoan* portType (interface) μέσω του οποίου ένα WSDL-defined Client Partner web service καλεί το operation *requestLoan*.



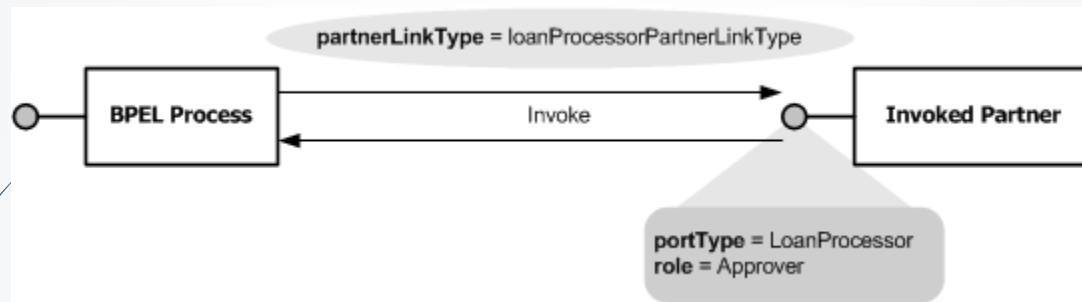
Ο Client Partner χρειάζεται ένα BPEL Process που να παρέχει ένα *IRequestLoan* portType (interface) με μια λειτουργία *requestLoan* την οποία θα καλέσει

Το BPEL Process προσφέρει ένα *IRequestLoan* portType (interface) με μια operation *requestLoan* που μπορεί να κληθεί από οποιοδήποτε web service.

Παράδειγμα Επικοινωνίας BPEL με Web Services

Scenario 2: WSDL-defined web services can be invoked by a BPEL process

Σ' αυτό το σενάριο, ο Invoked partner παρέχει ένα *ILoanProcessor* portType (interface) μέσω του οποίου, το WSDL-defined BPEL Process καλεί τη λειτουργία *processLoanRequest*.



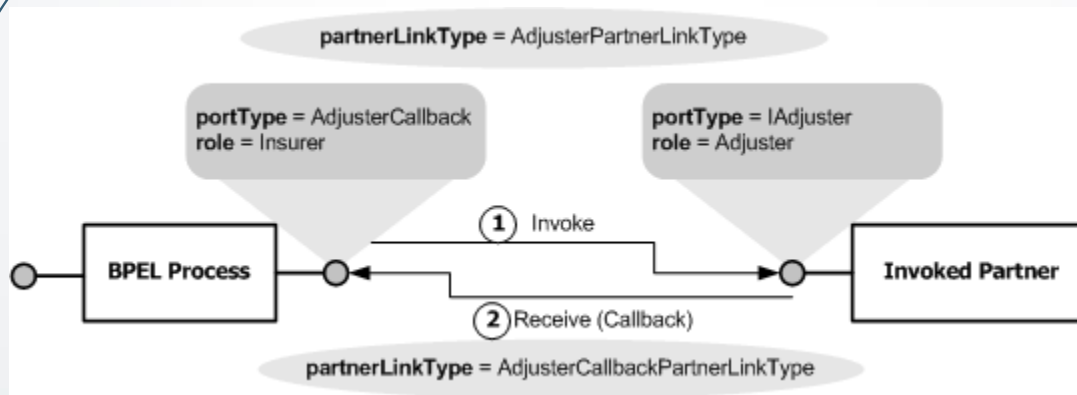
- Το BPEL Process απαιτεί να υπάρχει ένας WSDL-defined Invoked Partner για να παρέχει το *LoanProcessor* portType με τη λειτουργία *processLoanRequest* για να την καλέσει
- Ο Invoked Partner προσφέρει ένα *LoanProcessor* portType με μια λειτουργία *processLoanRequest* η οποία μπορεί να κληθεί από οποιαδήποτε web service.

Παραδείγματα Επικοινωνίας BPEL με Web Services

Scenario 3: WSDL-defined web services that have either characteristics

Σ' αυτό το σενάριο, ένας WSDL-defined Invoked Partner προσφέρει ένα *IAdjuster* portType (interface) που έχει μια λειτουργία *addNewAdjustmentJob*.

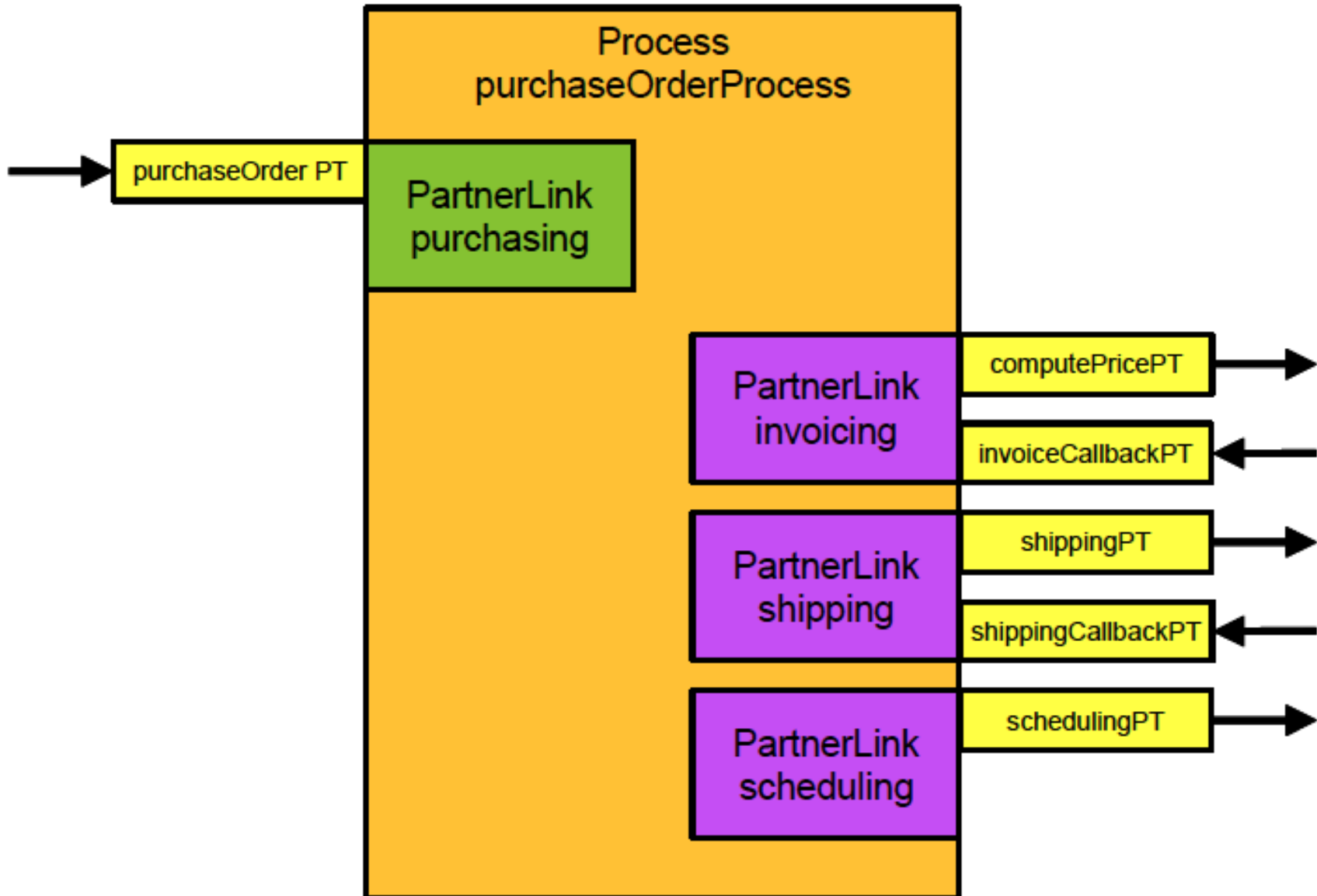
Το BPEL process παρέχει ένα *AdjusterCallback* portType (interface) που μπορεί να χρησιμοποιηθεί από τον Partner που έχει γίνει invoked για να κάνει callback τη λειτουργία *submitAdjustmentJob* της BPEL Process.



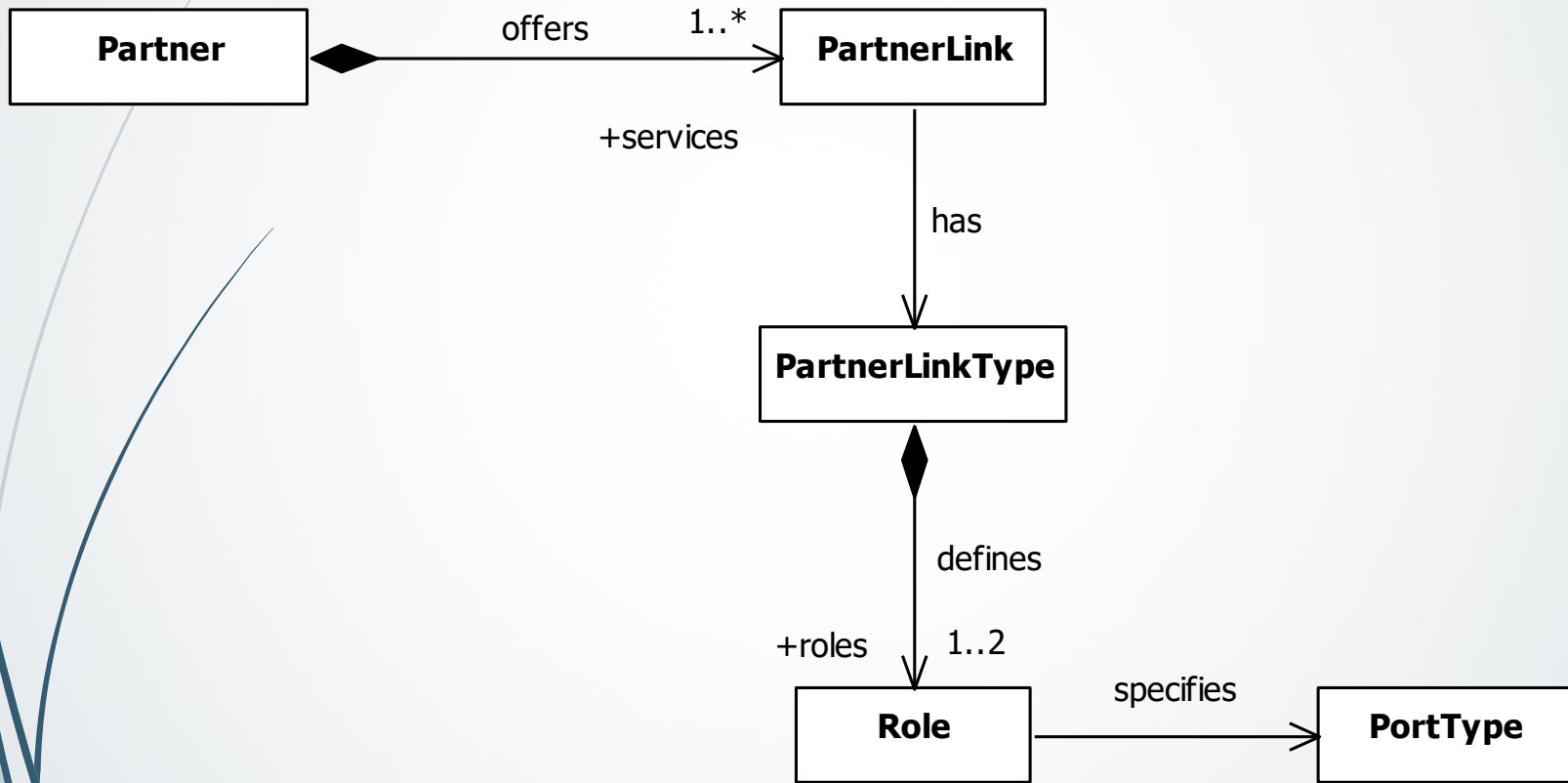
Που ορίζονται οι partners, κλπ

- Στο WSDL των services που καλούνται από τη BPEL process
- Στο WSDL των services που καλούν τη BPEL process
- Στο WSDL της BPEL process
- Στο BPEL process definition

Processes και Partners



Processes και Partners



Partner Links, Partner Link Types, Roles

Μπορεί να υπάρξει σύγχυση μεταξύ του partner links και του partner link type:

- ➔ **Partner link types** and **roles** are special WSDL extensions defined by the BPEL specification. As such, they are **defined in WSDL files**, not in the process BPEL file.
- ➔ **Partner Link** is a BPEL 2.0 element. It **is defined in the process BPEL file**.

Partner link types are prerequisites to the Partner Link element definition.

A Partner Link element can only be defined by referring to a particular partner link type and role which, as mentioned, must be defined in a WSDL file.

Partner Links, Partner Link Types, Roles

- ▶ The Partner Link **type** καθορίζει τη σχέση μεταξύ της διαδικασίας και των partners με το να ορίζει τους ρόλους που έχει η κάθε υπηρεσία σε ένα conversation.
 - ▶ Η σχέση αυτή ορίζεται περαιτέρω με τον ορισμό του port type που προσφέρεται από κάθε service για να λαμβάνει μηνύματα
 - ▶ Ένας ρόλος ορίζει ένα port type σ' ένα WSDL file.
- ▶ Οι ρόλοι καθορίζουν το conversational aspect της διαδικασίας ή του partner.
 - ▶ Χρησιμοποιείται ένας μονός ρόλος
 - ▶ Για κάθε σύγχρονη λειτουργία χρειάζεται ένας μόνο ρόλος γιατί τα αποτελέσματα επιστρέφονται χρησιμοποιώντας την ίδια λειτουργία.
 - ▶ Για κάθε ασύγχρονη λειτουργία χρειάζονται δύο ρόλοι, ένας για την κλήση της λειτουργίας και ένας για το callback (δηλαδή αναστρέφεται ο ρόλος του partner)

The partnerLinkType and role Definitions in WSDL Files - 1

- ▶ Τα *Partner link types* ορίζονται στα WSDL αρχεία μέσω του μηχανισμού επέκτασης της WSDL (extensibility element mechanism)
- ▶ Το `<partnerLinkType>` element ενσωματώνεται απ ευθείας στο αρχείο WSDL για κάθε partner process και για κάθε service process που συμμετέχουν σ' ένα BPEL orchestration
- ▶ Τα `<partnerLinkType>` elements που ορίζονται στο αρχείο WSDL των partners, προσδιορίζουν το portType (interface) element στο οποίο αναφέρεται το partnerLink που υπάρχει στον ορισμό του BPEL process.

The `partnerLinkType` and role Definitions in WSDL Files - 2

- ▶ To `<partnerLinkType>` element contains
 - ▶ one `<role>` element for each role (service provider or service consumer) the service process can play, as defined by its `partnerLink` element
 - ▶ `myRole` (indicating a service provider) and
 - ▶ `partnerRole` (indicating a service consumer) attributes—
in the service process's BPEL process definition.
- ▶ Therefore, a `partnerLinkType` can have either one or two `<role>` child elements.
- ▶ The following example shows one `<role>` child element.

The partnerLinkType and role Definitions in WSDL Files

```
<plink:partnerLinkType name="loanProcessorEjbPartner">  
  
    <plink:role name="loanProcessorEJB"  
        portType="ns:LoanProcessorEJBSEI">  
    </plink:role>  
  
</plink:partnerLinkType>
```

Ορισμός partner links στο BPEL file (1)

- ▶ **Partner Links** ορίζονται σ' ένα αρχείο BPEL.
- ▶ Ο ορισμός ενός **<partnerLink>** καθορίζει το portType (interface) του partner process που θα συμμετάσχει στα BPEL orchestrations της επιχειρησιακής διαδικασίας που ορίζεται
 - ▶ Αυτές τα partner processes μπορούν να είναι πελάτες της orchestrations της επιχειρησιακής διαδικασίας που ορίζεται στο BPEL ή μπορούν να κληθούν από την επιχειρησιακή διαδικασία
- ▶ Το στοιχείο **<partnerLink>** καταγράφει πληροφορία επικοινωνίας μεταξύ της BPEL process και των partners. Ο ρόλος της BPEL process εξαρτάται από τη φύση της επικοινωνίας της BPEL process με τους partners της.
- ▶ Στο παράδειγμά μας, όταν η BPEL process *LoanRequestor* καλεί (invokes) τον partner *LoanProcessorEJB*, έχει το ρόλο του *LoanRequestor* ενώ ο partner *LoanProcessorEJB* έχει το ρόλο του *LoanProcessorEJB*

Ορισμός partner links στο BPEL file (2)

```
<partnerLinks>  
  
  <partnerLink name="bpelImplemented"  
    partnerLinkType="ns0:loanRequestorPartnerLinkType" ***  
    myRole="loanRequestor"/>  
  
  <partnerLink name="ejbInvoked"  
    partnerLinkType="ns1:loanProcessorEjbPartner" +++  
    partnerRole="loanProcessorEJB"/>  
  
</partnerLinks>
```

*** Αυτό το partnerLinkType ορίζεται στο WSDL της business process

+++ Αυτό το partnerLinkType ορίζεται στο WSDL του web service που καλείται από το business process

Μεταβλητές (Variables)

- ▶ Χρησιμοποιούνται ως **buffers** κατά την εκτέλεση του *process*
 - ▶ Προσωρινή αποθήκευση μηνυμάτων που στέλνονται στο *process* από τους *partners* ή/και αντίστροφα
 - ▶ Αποθήκευση ενδιάμεσων δεδομένων που επηρεάζουν τη λογική εκτέλεσης του *process*
- ▶ Έχουν ως τύπο ένα από τα ακόλουθα:
 - ▶ WSDL message type
 - ▶ XML Schema simple type
 - ▶ XML Schema element

Variables: Παράδειγμα

```
<!-- Ορισμός ενός message στο WSDL document. -->  
<message name="customer" xmlns:x="http://example">  
  <part name="firstName" type="xsd:string"/>  
  <part name="lastName" type="xsd:string"/>  
  <part name="creditCard" type="xsd:string"/>  
</message>  
  
<!-- Ορισμός ενός variable στο BPEL document. -->  
<variable name="myCustomer" messageType="x:customer"/>
```


Message Properties

- ▶ Αποτελούν ένα γενικό τρόπο αναπαράστασης συγκεκριμένων *"κομματιών"* ενός μηνύματος
 - ▶ Ο ορισμός ενός property δημιουργεί ένα global όνομα και το συσχετίζει με ένα απλό XML Schema type
- ▶ **Τυπικές περιπτώσεις χρήσης:**
 - ▶ Εξαγωγή συγκεκριμένων τιμών από μια μεταβλητή
 - ▶ `getVariableProperty` ('variableName', 'propertyName')
 - ▶ Δημιουργία tokens που χρησιμοποιούνται για correlation μηνυμάτων και services

propertyAlias defined in WSDL files

- ➔ Map properties to messages.
- ➔ Message parts contain business-application-specific properties.
- ➔ A property alias maps a specific property to a specific element or attribute in a message part.
 - ➔ Once defined, you can use the property name as an alias for the message part and the location of the property within the message.
 - ➔ Property Aliases are defined in the WSDL file using the WSDL extensibility element mechanism.
 - ➔ The query expression used to access the relevant element or attribute is written in the selected query language, the default being XPath 1.0.

Δραστηριότητες (Activities)

► Βασικές Δραστηριότητες

- receive, reply, invoke, assign, throw, terminate, wait, empty

► Δομημένες Δραστηριότητες

- sequence, switch, while, pick, flow, scope, compensate

Providing WS operations

➤ receive

- χρησιμοποιείται για τη **λήψη μηνυμάτων** από Web services
- αποτελεί τον μοναδικό τρόπο δημιουργίας στιγμιοτύπων ενός *business process*

➤ reply

- χρησιμοποιείται για την **αποστολή απόκρισης** σε ένα μήνυμα που ελήφθη με τη χρήση της *receive*

Ο συνδυασμός των δραστηριοτήτων **receive-reply** χρησιμοποιείται για την **υλοποίηση σύγχρονων request-response αλληλεπιδράσεων** μεταξύ του **business process** και των συμμετεχόντων **partners**

receive-reply: Παράδειγμα

```
<process name="PurchaseOrderProcess" targetNamespace="uri"
  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/">
  <!-- Ορισμός partnerLinks, variables, κτλ. -->

  <sequence>
    <!-- Λήψη ενός μηνύματος από τον πελάτη, σχετικά με μια παραγγελία. -->
    <receive partnerLink="purchasing" portType="purchaseOrderPT"
      operation="sendPurchaseOrder" variable="PO">
    </receive>

    <!-- Επεξεργασία της παραγγελίας. -->

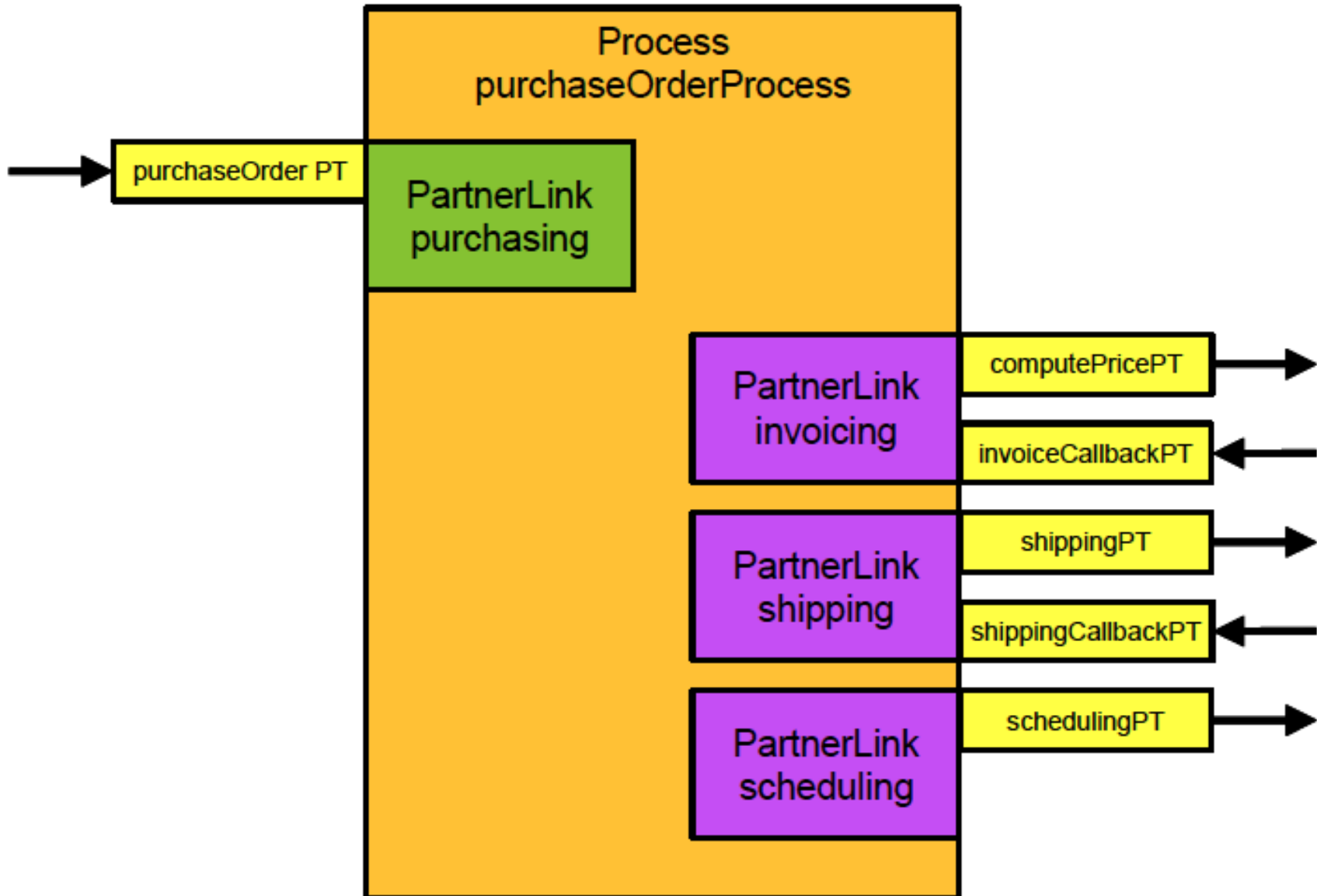
    <!-- Αποστολή του τιμολογίου ως απόκριση στο αρχικό μήνυμα του πελάτη. -->
    <reply partnerLink="purchasing" portType="purchaseOrderPT"
      operation="sendPurchaseOrder" variable="invoice"/>
  </sequence>
</process>
```

Invoking WS operations

➤ invoke

- κλήση προς ένα **web service operation** ΠΟΥ προσφέρεται από κάποιον *partner*
- **σύγχρονη (request/response)** ή **ασύγχρονη (one-way)** ΕΠΙΚΟΙΝΩΝΙΑ
- Δυνατότητα stateful επικοινωνίας με τη χρήση **correlations**
- **Fault** και **compensation handling**

Processes και Partners



invoke: Παράδειγμα

```
<process name="PurchaseOrderProcess" targetNamespace="uri"
  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/">
  <!-- Ορισμός partnerLinks, variables, κτλ. -->

  <sequence>
    <!-- Λήψη ενός μηνύματος από τον πελάτη, σχετικά με μια παραγγελία. -->

    <!-- Υπολογισμός της τιμής για την παραγγελία. -->
    <invoke partnerLink="invoicing" portType="computePricePT"
      operation="priceCalculation"
      inputVariable="PO"
      outputVariable="POPrice">
    </invoke>

    <!-- Αποστολή τιμολογίου στον πελάτη. -->
  </sequence>
</process>
```


Assignment

➤ assign

- χρησιμοποιείται για την **ανάθεση τιμών** και **ενημέρωση** των *variables* σε ένα *business process*
- **αντιγράφει** δεδομένα από μια μεταβλητή σε κάποια άλλη
- τα *from* και *to* πρέπει να έχουν **συμβατούς τύπους**
- μπορεί να περιλαμβάνει μία ή περισσότερες **στοιχειώδεις αναθέσεις**

throw

- ▶ Χρησιμοποιείται για τη **σηματοδότηση εσωτερικών σφαλμάτων** ενός *business process*
- ▶ Ένα σφάλμα ταυτοποιείται από το **όνομά του**, που είναι μοναδικό στο scope του business process
 - ▶ **faultName** attribute
- ▶ Προαιρετικά, η δραστηριότητα μπορεί να περιλαμβάνει μια **μεταβλητή** η οποία **παρέχει πληροφορία** για το συγκεκριμένο σφάλμα
 - ▶ **faultVariable** attribute

throw: Παράδειγμα

Χωρίς χρήση του `faultVariable`

```
<throw xmlns:FLT="http://example.com/faults"  
      faultName="FLT:OutOfStock"/>
```

Με χρήση του `faultVariable`

```
<throw xmlns:FLT="http://example.com/faults"  
      faultName="FLT:OutOfStock"  
      faultVariable="item"/>
```

terminate

- ▶ Χρησιμοποιείται για τον **άμεσο τερματισμό** της εκτέλεσης ενός *business process*
- ▶ Προκαλεί τον τερματισμό όλων των εκτελούμενων δραστηριοτήτων του *business process*
- ▶ Η διαδικασία τερματισμού είναι άμεση, **χωρίς χειρισμό σφαλμάτων ή υποστήριξη ανάκαμψης**

wait

- ▶ Επιτρέπει σε ένα *business process* να προσδιορίσει μια **επιθυμητή χρονική καθυστέρηση**
- ▶ Η **χρονική καθυστέρηση** προσδιορίζεται με τη χρήση κάποιου *expression* και μπορεί να είναι είτε **σταθερή** είτε **μεταβλητή**
- ▶ **Τυπικές περιπτώσεις χρήσης**
 - ▶ Κλήση προς ένα *web service operation* κατά μια συγκεκριμένη χρονική στιγμή

wait: Παράδειγμα

```
<process ...>
...
<sequence>
  <wait until=«'2021-12-25T00:00+01:00'"/>

  <invoke partnerLink="SantaClaus" portType="xmasPT"
    operation="deliverPresents"
    inputVariable="addresses">
  </invoke>
</sequence>
...
</process>
```

empty

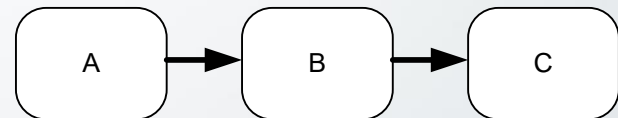
- ▶ Αποτελεί τον **απλούστερο τύπο δραστηριότητας** στην WS-BPEL
- ▶ Υποδηλώνει μια **default** δραστηριότητα που **δεν κάνει τίποτα**
- ▶ **Περιπτώσεις χρήσης**
 - ▶ Καταστολή ενός συλληφθέντος σφάλματος
 - ▶ Default επιλογή σε ένα βρόχο exclusive choice
 - ▶ Κατά τη διάρκεια μοντελοποίησης και ανάπτυξης του business process

Δομημένες Δραστηριότητες

sequence

- ▶ Περιέχει μία ή περισσότερες δραστηριότητες οι οποίες **εκτελούνται ακολουθιακά**, με τη σειρά εμφάνισής τους
- ▶ Τερματίζει όταν και η τελευταία περιεχόμενη δραστηριότητα τερματίσει

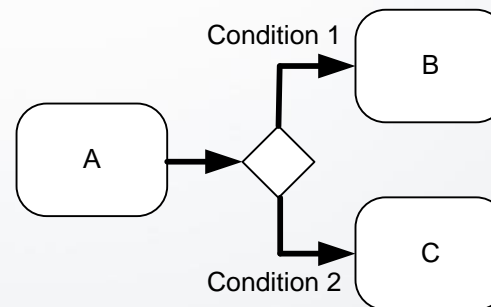
Αντιστοιχία με την *BPMN*:



switch

- ▶ Υποστηρίζει την **υπό συνθήκη εκτέλεση δραστηριοτήτων**
- ▶ Εξετάζει μια σειρά από συνθήκες και **εκτελεί την αντίστοιχη δραστηριότητα για την πρώτη αληθή συνθήκη**
- ▶ Αν δεν έχει προσδιοριστεί η default δραστηριότητα προς εκτέλεση, τότε υπονοείται η εκτέλεση της δραστηριότητας `empty`
- ▶ Ολοκληρώνει τη λειτουργία της, όταν η επιλεγμένη προς εκτέλεση δραστηριότητα τερματίσει

Αντιστοιχία με την *BPMN*:



switch: Παράδειγμα

```
<switch xmlns:inventory="http://supply-chain.org/inventory"
        xmlns:FLT="http://example.com/faults">

  <case condition="bpws:getVariableProperty(stockResult,level) > 100">
    <flow>
      <!-- perform fulfillment work -->
    </flow>
  </case>

  <case condition="bpws:getVariableProperty(stockResult,level) <= 0">
    <throw faultName="FLT:OutOfStock" variable="RestockEstimate"/>
  </case>

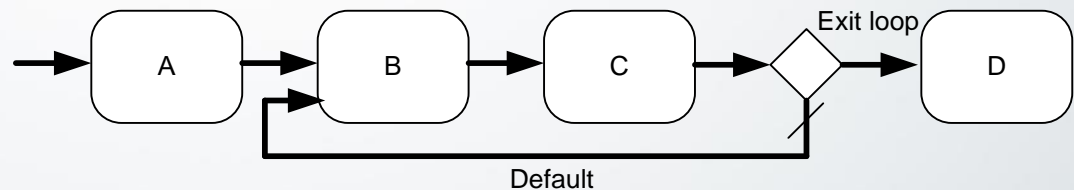
  <otherwise>
    <throw faultName="FLT:ItemDiscontinued"/>
  </otherwise>

</switch>
```

while

- ▶ Έχει την ίδια σημασιολογία με αυτή αντίστοιχων δομών άλλων "κλασικών" γλωσσών προγραμματισμού
- ▶ Υποστηρίζει την **επαναληπτική εκτέλεση μια προσδιορισμένης δραστηριότητας**, με βάση κάποια συνθήκη
- ▶ Η επαναληπτική εκτέλεση **συνεχίζεται όσο η συνθήκη έχει την τιμή *true***

Αντιστοιχία με **BPMN**:



while: Παράδειγμα

...

<!-- Επανάλαβε όσο υπάρχουν αντικείμενα προς επεξεργασία... -->

```
<while condition="bpws:getVariableData('itemsShipped') &lt;  
    bpws:getVariableProperty('shipRequest','props:itemsTotal')">
```

```
<sequence>
```

...

```
</sequence>
```

```
</while>
```

...

pick

- ▶ Χρησιμοποιείται για την αναμονή κάποιου εκ των καθορισμένων συμβάντων
- ▶ Ενεργοποιείται με την πραγματοποίηση του πρώτου χρονικά συμβάντος από το προκαθορισμένο σύνολο
- ▶ Εκτελεί τη δραστηριότητα που συσχετίζεται με το συμβάν και παύει να αναμένει τα υπόλοιπα συμβάντα του συνόλου
- ▶ Ειδική χρήση: Δημιουργία στιγμιοτύπου του business process ως απόκριση σε κάποιο συμβάν (λήψη μηνύματος, **onMessage**)
 - ▶ `createInstance="yes"`

pick: Παράδειγμα

```
<pick>
  <onMessage partnerLink="buyer" portType="orderEntry"
    operation="inputLineItem" variable="lineItem">
    <!-- activity to add line item to order -->
  </onMessage>

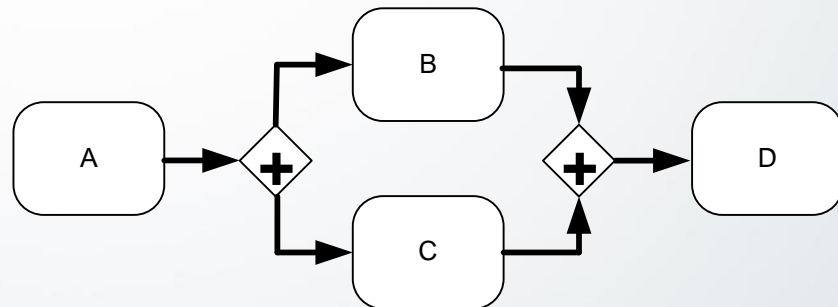
  <onMessage partnerLink="buyer" portType="orderEntry"
    operation="orderComplete" variable="completionDetail">
    <!-- activity to perform order completion -->
  </onMessage>

  <!-- set an alarm to go after 3 days and 10 hours -->
  <onAlarm for=" 'P3DT10H' ">
    <!-- handle timeout for order completion -->
  </onAlarm>
</pick>
```

flow

- ▶ Παρέχει δυνατότητες **συγχρονισμού** και **παράλληλης εκτέλεσης** δύο ή περισσότερων δραστηριοτήτων
- ▶ Τερματίζει όταν ολοκληρώσουν την εκτέλεσή τους όλες οι περιεχόμενες δραστηριότητες

Αντιστοιχία με *BPMN*:



flow: Παράδειγμα

```
<sequence>
```

```
<!-- The 2 web service invocations will occur concurrently. -->
```

```
<flow>
```

```
  <invoke partnerLink="Seller" ... />
```

```
  <invoke partnerLink="Shipper" ... />
```

```
</flow>
```

```
<!-- The 3rd web service invocation will occur after both Seller and  
      Shipper respond. -->
```

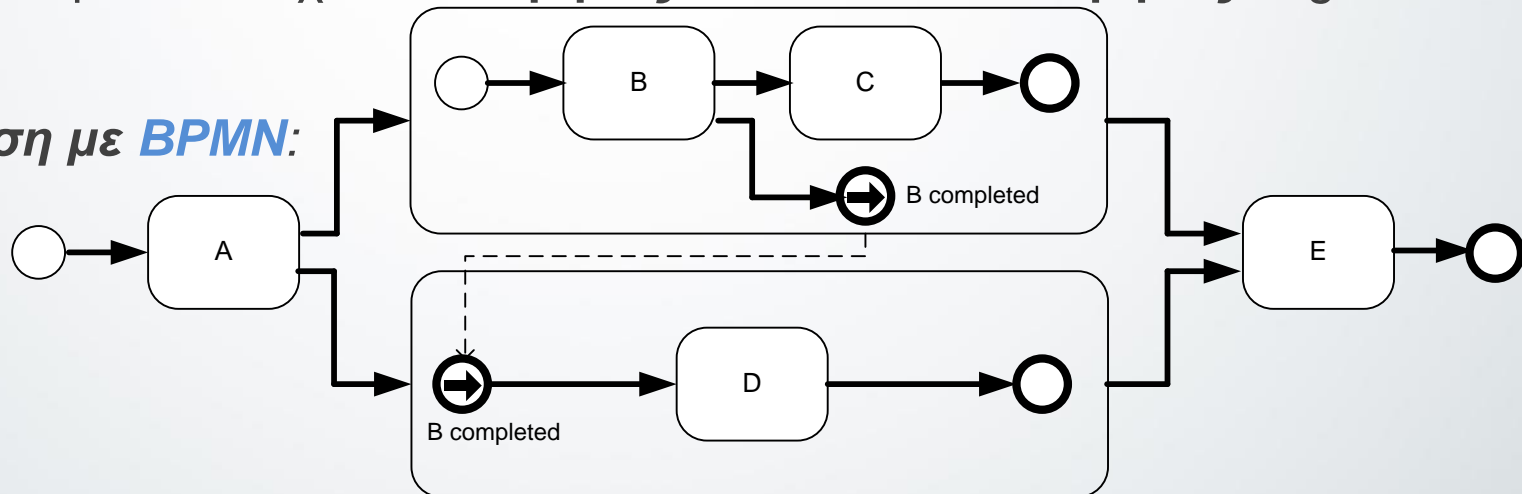
```
<invoke partnerLink="Bank" ... />
```

```
</sequence>
```

flow links

- ▶ Χρησιμοποιούνται για την **επίτευξη συγχρονισμού** μεταξύ των δραστηριοτήτων που εκτελούνται σε ένα flow
- ▶ Ορίζονται εντός της δραστηριότητας flow, στην οποία εφαρμόζονται
- ▶ Η σύνδεση 2 δραστηριοτήτων που πρέπει να συγχρονιστούν πραγματοποιείται με τη χρήση των στοιχείων **source** και **target**
- ▶ Κάθε link πρέπει να έχει **ένα ακριβώς source** και **ένα ακριβώς target**

Αντιστοίχιση με **BPMN**:



flow links: Παράδειγμα

```
<flow>
```

```
<!-- Οι ακολουθίες X, Y και η κλήση D θα εκτελεστούν παράλληλα,  
έχουν ωστόσο κάποιους περιορισμούς συγχρονισμού. -->
```

```
<!-- Ορισμός των links. -->
```

```
<links>
```

```
<link name="XtoY"/>
```

```
<link name="CtoD"/>
```

```
</links>
```

```
<!-- Η ακολουθία X θα εκτελεστεί πριν από την ακολουθία Y. -->
```

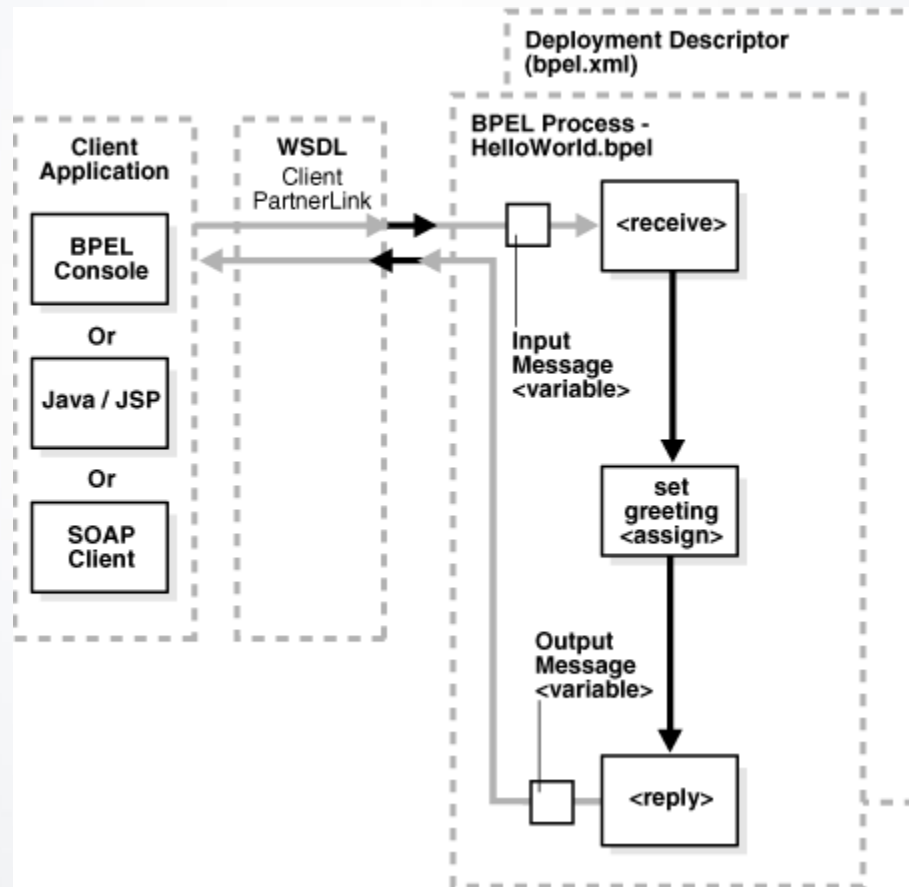
```
<sequence name="X">
```

```
<source linkName="XtoY"/>
```

```
<invoke partnerLink="A" ... />
```

```
<invoke partnerLink="B" ... />
```

```
</sequence>
```



Συμπεράσματα

- ▶ Η WS-BPEL είναι μια XML γλώσσα και χρησιμοποιείται για τον ορισμό υπηρεσιοστρεφών επιχειρησιακών διαδικασιών
- ▶ Κάθε task σε μια WS-BPEL διεργασία εκτελείται από ένα web service
- ▶ Οι WS-BPEL διεργασίες προσφέρουν την λειτουργικότητά τους ως ένα Web service και περιγράφουν τη διεπαφή τους με τη χρήση της γλώσσας WSDL

Για περισσότερες πληροφορίες...

- ▶ Επίσημη προδιαγραφή του προτύπου
 - ▶ <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>

Μπορείτε επίσης να δείτε

- ▶ <http://www.oracle.com/technetwork/articles/matjaz-bpel1-090575.html>