### LAB 1– Value Model & BPMN Modeling

Ingegneria dei Processi Aziendali

Module 2 - Lab

Unit 1 – Process Modeling

**Fulvio Frati** 

Università di Milano

### Outline

### **Analyzed Process: Expense Reimbursement**

Value Model

**BPMN Model** 

## **Analyzed Process - 1**

The process is a sample expense reimbursement process:

- It provides for reimbursement of expenses incurred by employees for the company. E.g. buying a technical book, office supplies or software
- In a normal day there are several hundreds of instances of this process created

**Three Actors** 

- Employee
- Reimbursement Office
- Supervisor

## Value Model - 1



- Employees buying goods
- Employee considered as a *Market Segment*

### Value Model - 2



- E3value can model complex value flows
- AND/OR splits are supported by the model

## **Analyzed Process - 2**

- 1. The employee create a reimbursement request. After the Expense Report is received by the Office, a new account must be created if the employee does not already have one
- 2. The report is then reviewed by the Office for automatic approval
  - Amounts under \$200 are automatically approved
  - Amounts equal to or over \$200 require approval of the supervisor
  - In case of rejection, the employee must receive a rejection notice by email
- 3. The reimbursement goes to the employee's direct deposit bank account
- 4. If no action has happened in 7 days, then the employee must receive an approval in progress email
- 5. If the request is not finished within 30 days, then the process is stopped and the employee receives an email cancellation notice and must re-submit the expense report

## **BPMN Diagram: Create Account**

- Employee Create Reimb. Request **Expense Report Process** Create new Account **Reimbursement Office** no Receive Reimb Request yes Account exists?
- 1. The employee create a reimbursement request. After the Expense Report is received by the Office, a new account must be created if the employee does not already have one

### **BPMN Diagram: Auto-approve**



## 2. The report is then reviewed by the Office for automatic approval

• Amounts under \$200 are automatically approved

## **BPMN Diagram: Supervisor Approval**



2. The report is then reviewed by the Office for automatic approval

• Amounts equal to or over \$200 require approval of the supervisor

3. The reimbursement goes to the employee's direct deposit bank account

## **BPMN Diagram: Start Inspection**



### **BPMN Diagram: Approval in Progress**



4. If no action has happened in 7 days, then the employee must receive an approval in progress email

## **BPMN Diagram: Cancellation Notice**



5. If the request is not finished within 30 days, then the process is stopped and the employee receives an email cancellation notice and must re-submit the expense report <sup>12</sup>

## **BPEL MODELING**

## **BPEL Modeling**

# In the Lab we concentrate only in the visual modeling of the process

### **Different modeling steps**

- Define the partners list
- Define the type of data in input to the process
- Define (or import) the partner WSDL
- Build the process

## **BPEL: Create client's XSD - 1**

### **XSD** define types of the data exchanged

### **Define types for the client application**

# Important to design the type of operation to execute and data to exchange

• Data are defined as XML complex types

### **BPEL: Create Client's XSD - 2**

e name string

e result string

S Schema : http://ww	ww.example.org/expXSD			
📛 Directives				
Elements	Types			
e input : inputCplx	inputCplx			
e output : outputCplx	Se outputCplx			
	الــــــا			
Attributes	Groups			
	]]			

Define the type of data given as input to th	e
process	

## **BPEL: Create Employee WSDL - 1**

Define the WSDL for the employee management web service

### Generally WSDLs are directly imported from source applications

### The service provides the following operations:

- getAccount
- createAccount
- getBankDetails

### **BPEL: Create Employee WSDL - 2**

H	<ol> <li>employ</li> </ol>					
	🏶 getAccount					
	🕼] input	🕝 name	😑 string			
	🗘 output	parameters	e getAccountResponse			
	🏶 getBankDetails					
	🕼] input	🕝 name	😑 string			
	🗘 output	parameters	e getBankDetailsResponse			
	light createAccount					
	input		🖫 createCmplx			
	↓] output □ parameters		e createAccountResponse			

### Define the operation and the type of data exchanged

E getAccountResponse	
e getAccountResponse	
	getAccountResponseType)
	e name string
	e amount string
	e descript string
//www.example.org/employ/	Open In New Editor
ankDetailsResponse	~
E getBankDetailsResponse	(getBankDetailsResponseType)     (e) bankname string     (e) BAN string
/www.example.org/employ/	[Open In New Editor
ateCmplx	▼
e BAN string E IBAN string	
	Open In New Editor

18

## **BPEL: Create Supervisor WSDL**

### **Define the supervisor service**

### **Operation:**

- approve
  - Input: name amount description
  - Output: decision
- makePayment
  - Input: amount bank IBAN
  - Output: result

### **BPEL: Create a new process**

## Create an empty structure for the process

Each process starts with a *receiveInput*, to receive input from the client, and ends with *callbackClient*, to return results

Define the type of input and output variables importing the client XSD

₹ main
receiveInput
allbackClient

## **BPEL: call Partner Operation**

Import the partner WSDL to be called by the process

**Define the partner role** 

Add the *requestEmployeeProfile* and select the operation (*getAccount*)





## **BPEL: assign Value to Parameters**

🖹 Markers 🔲 Prop	perties 🛛 🖓 Servers 🙀 Data So	ource Explorer 📔 Snip	pets				2 - 1
😑 Assign							
Description	<u>V</u> alidate						
Details	Variable to Variable	From:	Variable	*	<u>T</u> o:	Variable	*
Join Behavior		+ employeeR	equest : getAccountRequest		employeeR	equest : getAccountRequest	
Namespaces			esponse : getAccountResponse		name :	string	
Documentation	□ ■ input : input			🗈 🖷 employeeResponse : getAccountResponse			
	e name : string			🕀 🐨 😑 input : input			
		⊞…	tput			itput	
			polypame.		0		
	<u>N</u> ew <u>D</u> elete	Query:	TIS 1.1 dille		Query:		
	Move Up Move Down			Ignore Missing Source Data			Keep Source Element Name

### BPEL uses the *assign* task to map value from client to operation parameters

Select the source and destination of parameter values



### **BPEL: insert IF statement**



#### Add the IF statement

Define the condition: use the XPath sintax

Call the createAccount operation if the account does not exist



## **BPEL: Wait 7 days**

#### Use the Pick element: wait until the first event applies:

- The getAccount Web Service returns
   → process continues
- The 7 days timer lasts
   → send mail



