

Master Software Technology Business Process Technology – [08] Workflows

Agenda

Workflow Management in BPM

Workflow Management key tasks

Workflow terms and definitions

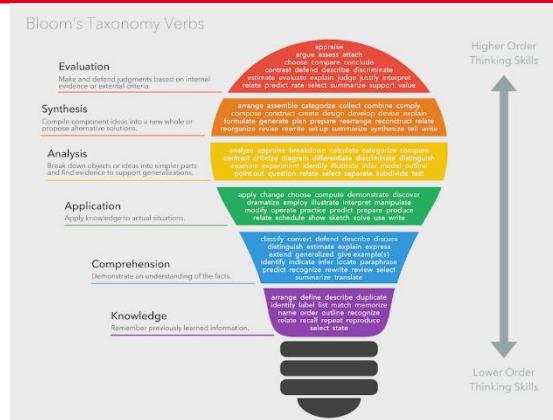
The WfMC reference model

Workflow modeling/execution/monitoring using SaaS: Fujitsu RunMyProcess

Workflow Modeling Standards overview

Business Process Execution Language (BPEL)

Learning Goals



- ✓ Understand the area of Workflow Management
- ✓ Understand its relation to other BPM areas
- ✓ Understand key tasks in Workflow Management
- ✓ Model and execute workflows using Web Services and BPEL

Workflow Management in the BPM cycle

Implement

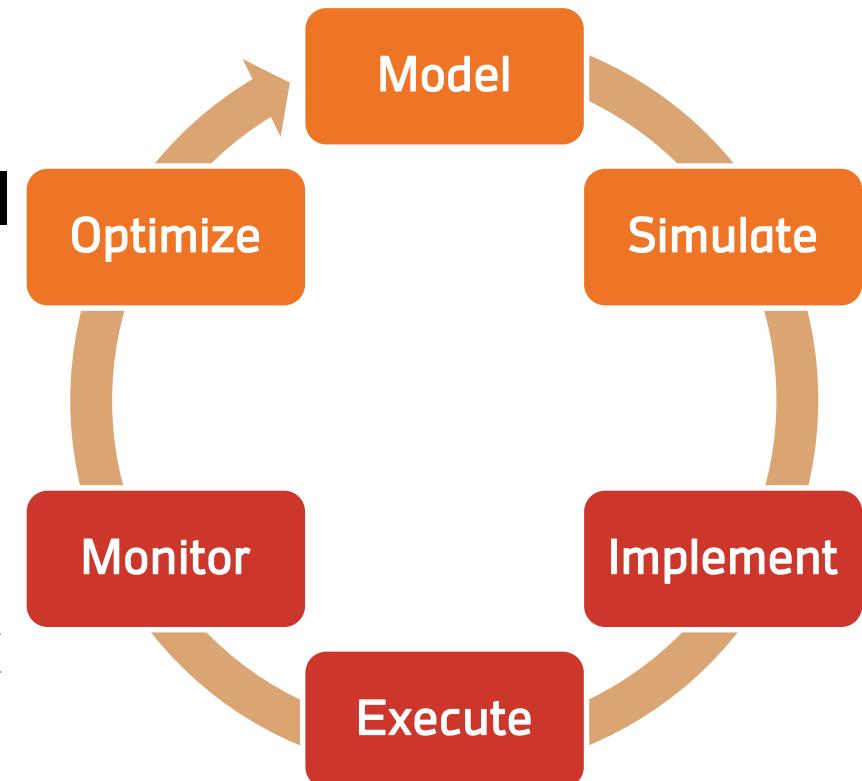
- Resolve all open questions, incl. technical

Execute

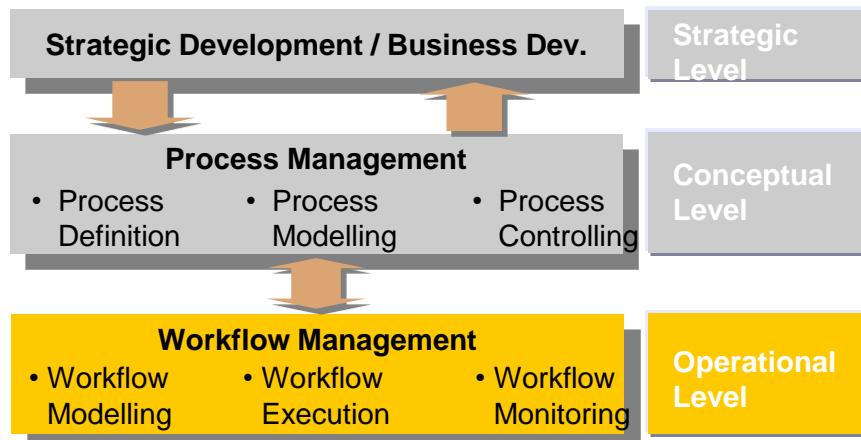
- Earn the merits of previous efforts

Monitor

- Gather data to feedback on performance



Workflow Management key tasks (Recap)



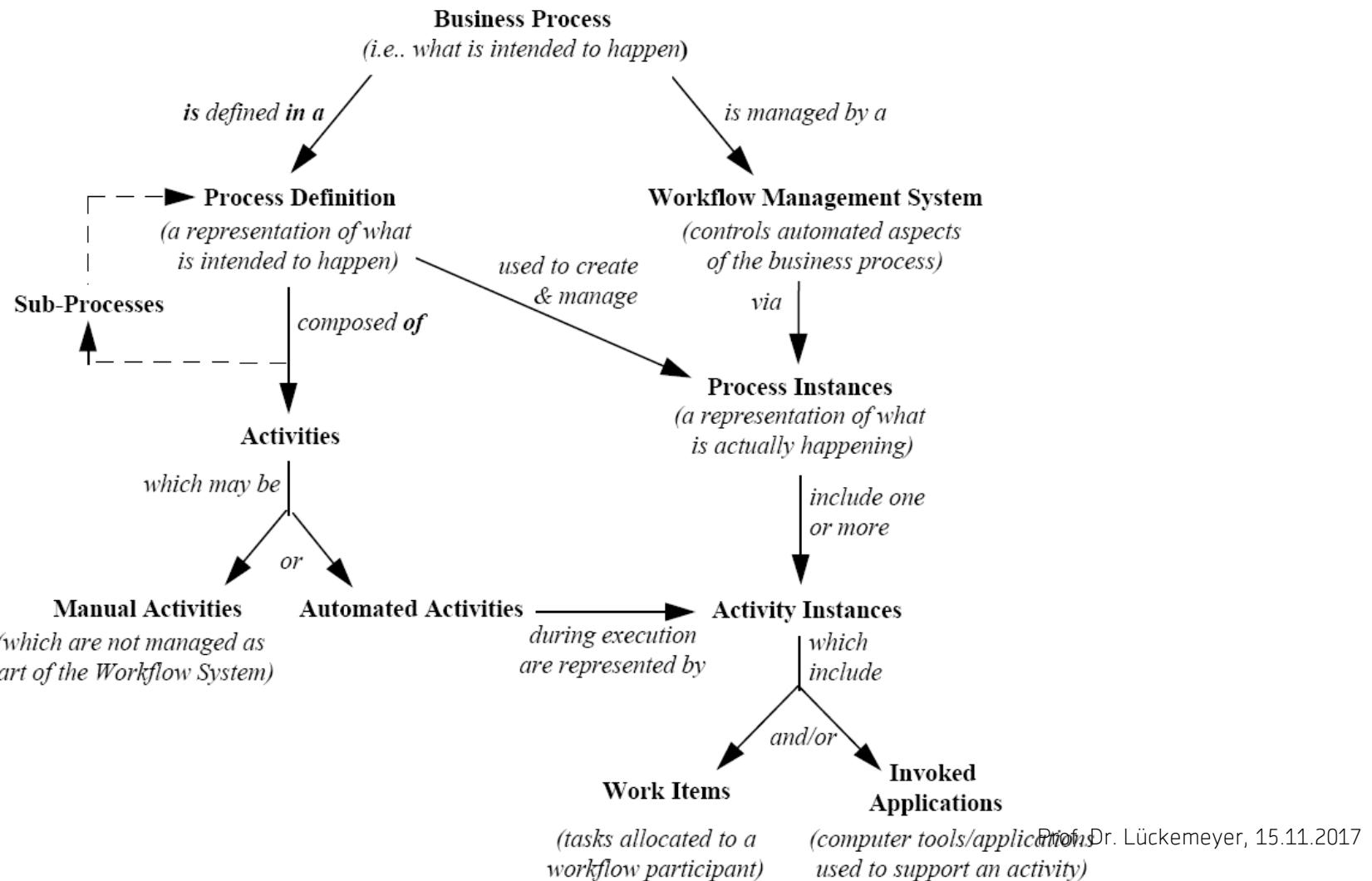
Key Tasks

- Workflow Modelling
 - Modelling executable processes as workflow templates
- Workflow Execution
 - Creating workflow instances
 - Running them in a **workflow engine** (part of a Workflow Mgmt. System)
- Workflow Monitoring
 - Monitoring process performance: comparing actual and target values of KPIs

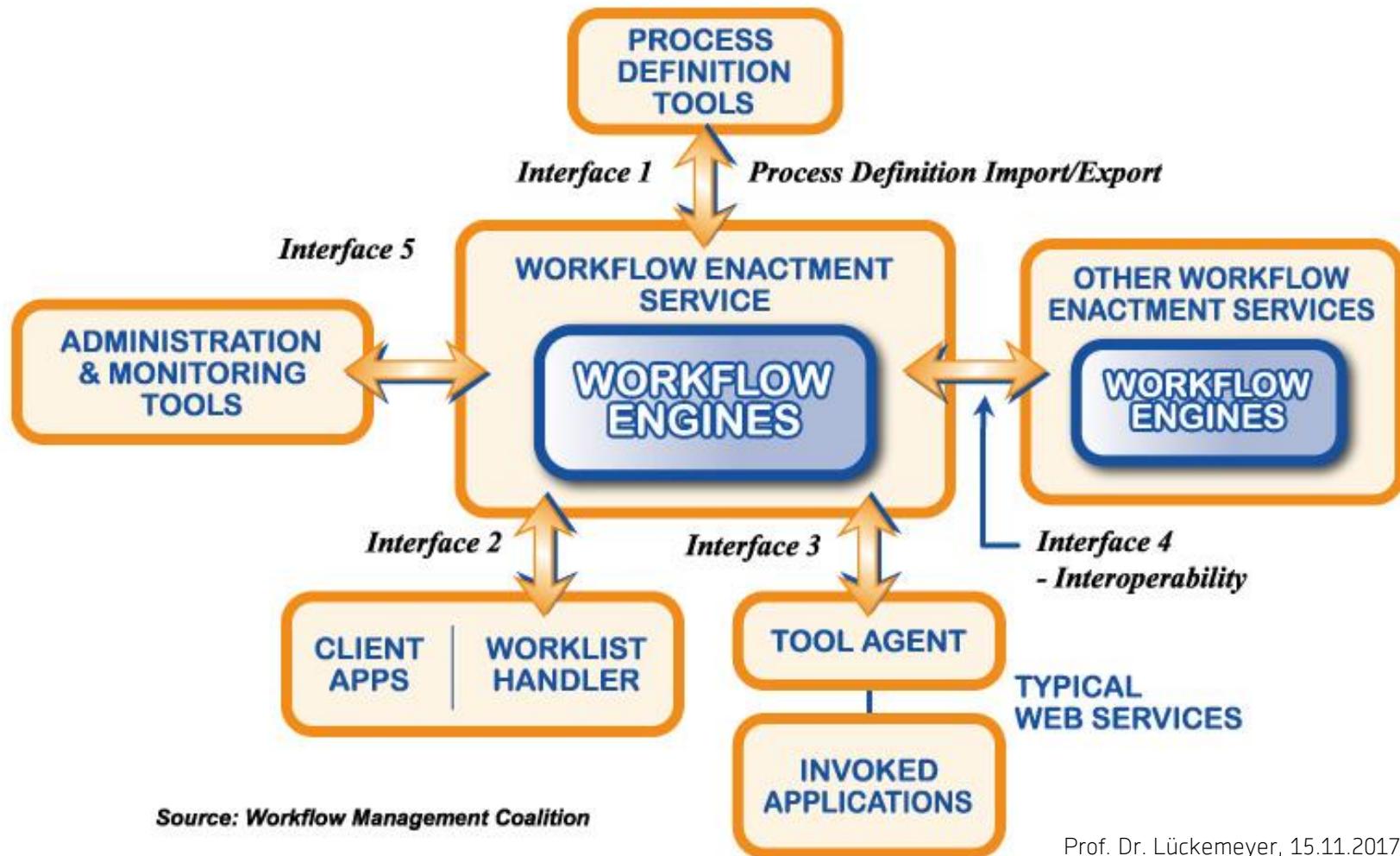
Terms and Definitions

- Workflow Management Coalition (WfMC)
 - Organization which covers Workflows since 1993
- Business Process
 - A set of one or more linked procedures or activities which collectively realise a business objective or policy goal, normally within the context of an organisational structure defining functional roles and relationships.
- Workflow
 - The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.
- Workflow Management System
 - A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications.

Terms and relations



The WfMC Reference Model



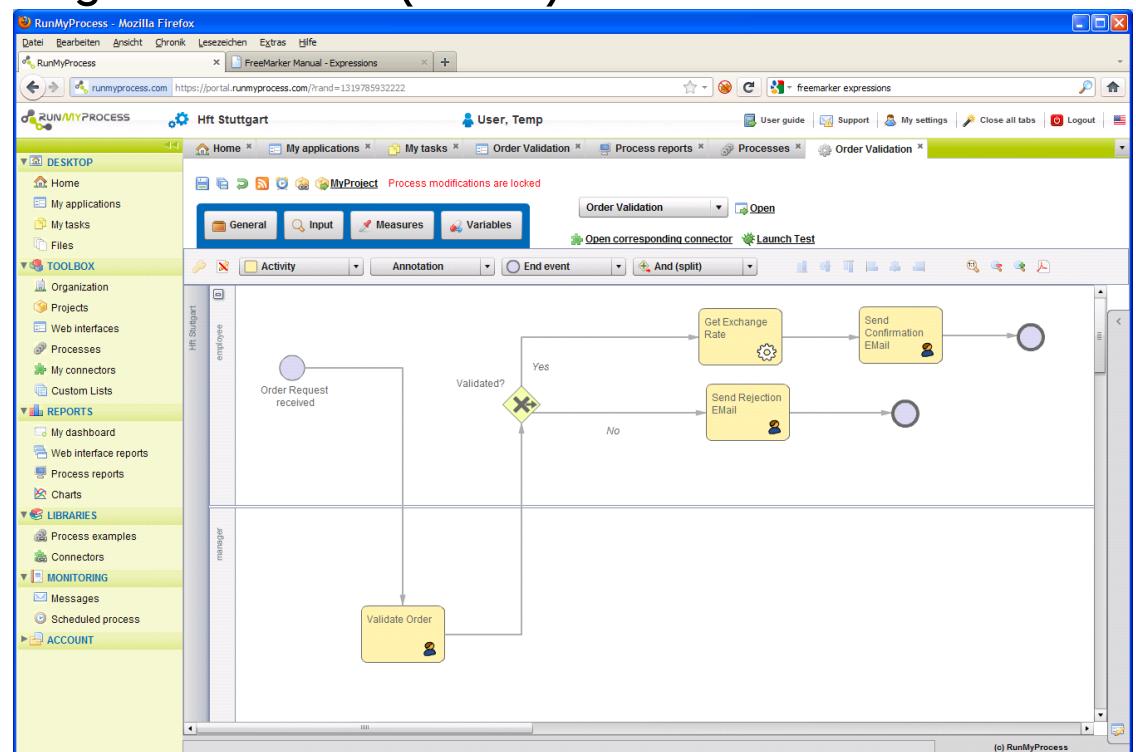
Fujitsu RunMyProcess

Web-based Workflow-Management-Tool (Flash)

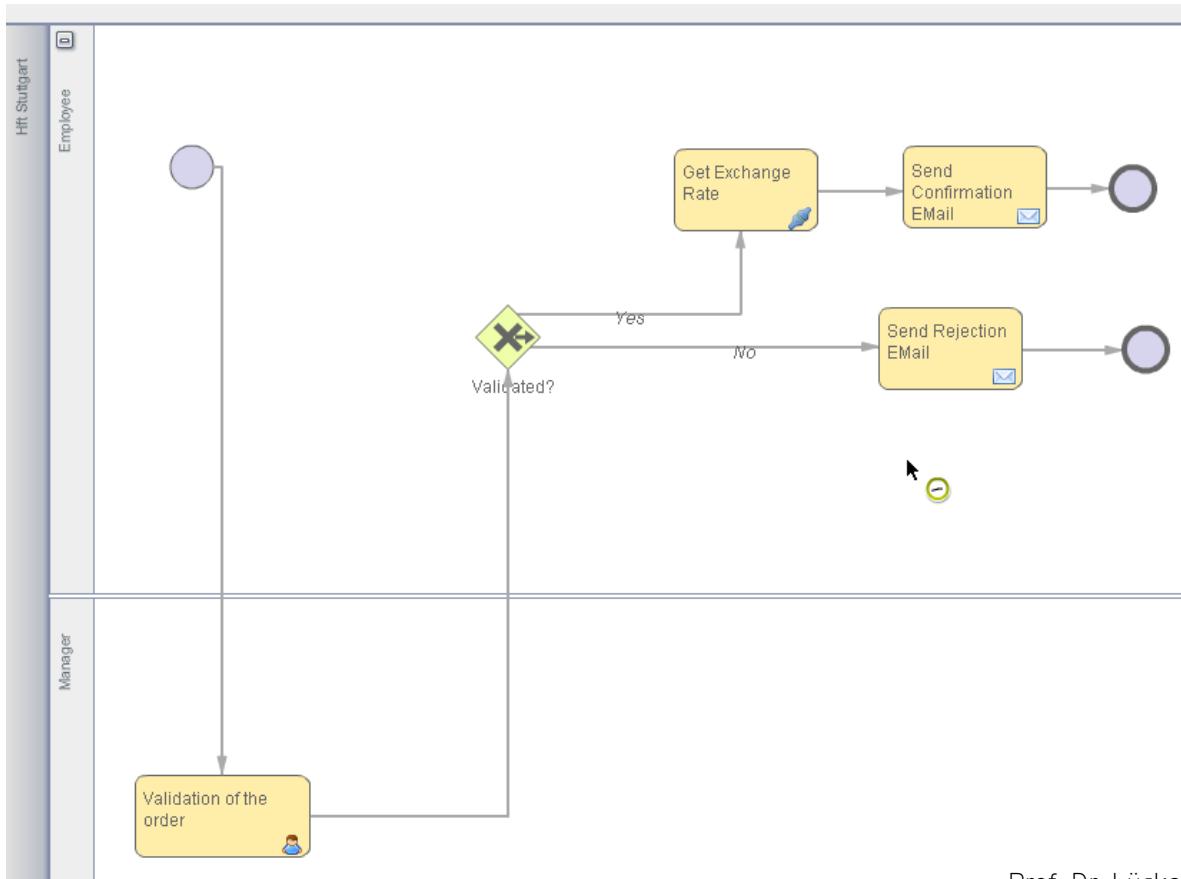
Executable BPMN

Web-Forms to enter data
in manual tasks

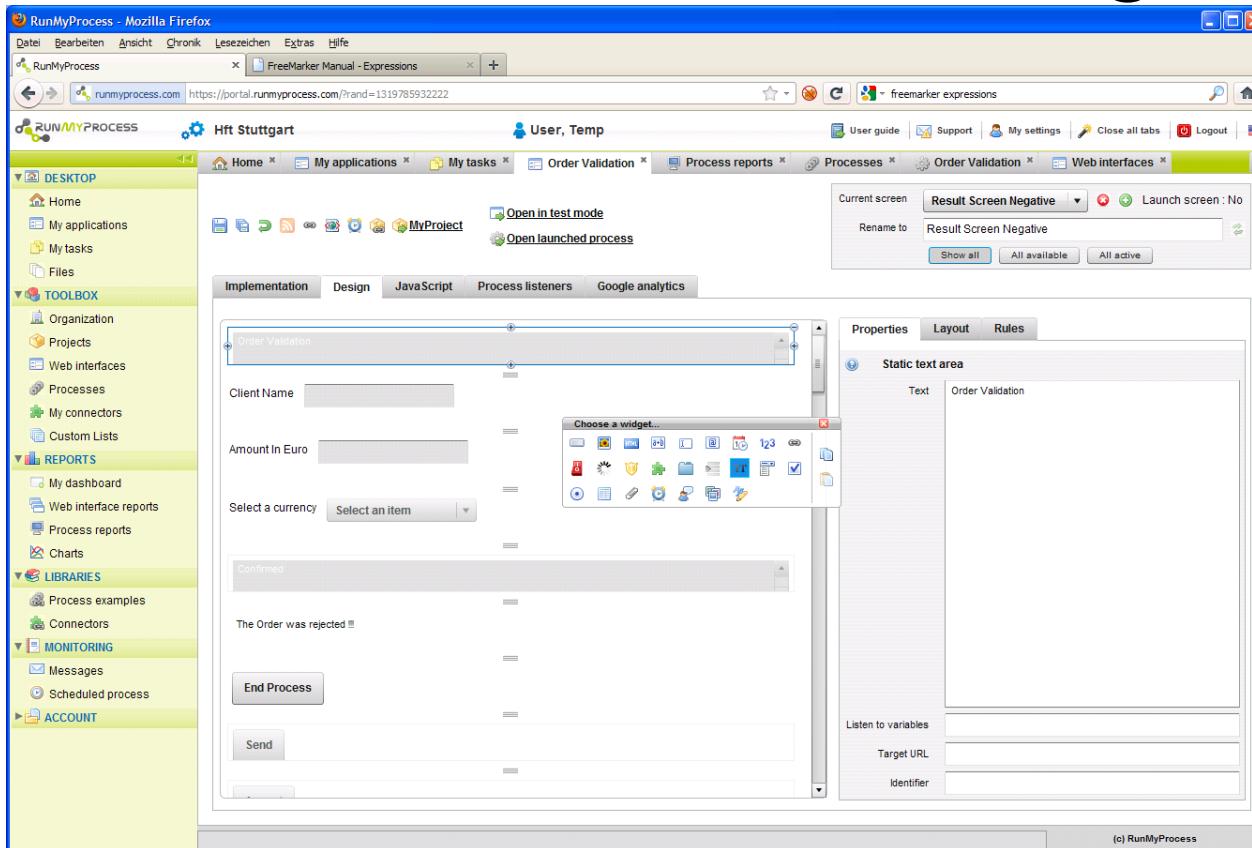
Support of Web-Service-
Tasks and E-Mail-Tasks



Fujitsu RunMyProcess example: order validation



Fujitsu RunMyProcess features: Forms / Web-Interface Designer

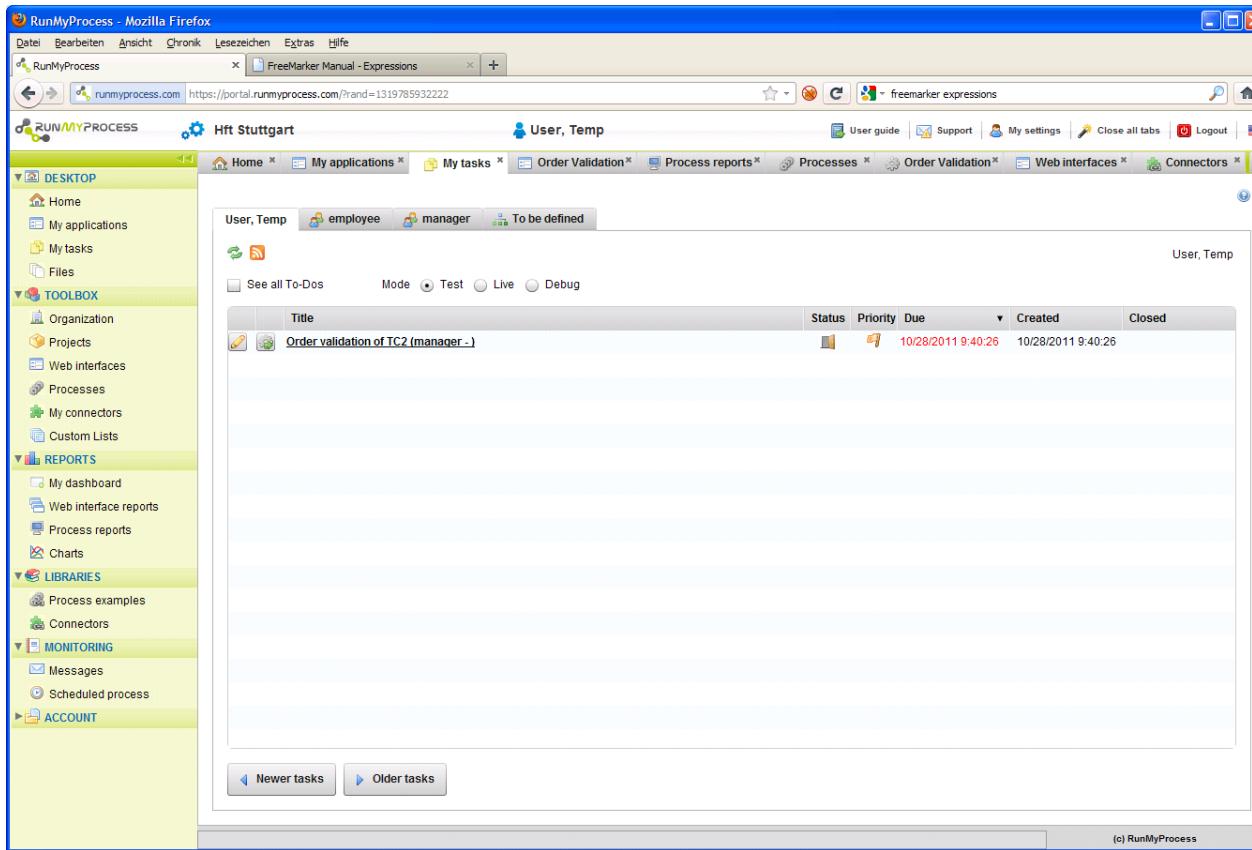


Fujitsu RunMyProcess features: Predefined Connectors & Services

The screenshot shows a Mozilla Firefox browser window displaying the RunMyProcess platform. The URL is <https://portal.rumyprocess.com/>. The page title is "RunMyProcess - Mozilla Firefox". The main content area shows a list of predefined connectors and services, categorized by provider. The left sidebar contains navigation links for Desktop, Toolbox, Reports, Libraries, Monitoring, and Account.

State	Name	Provider	Description
Green	Qualcomm HOS Driver Verify / QUALCOMM HOS Provider (All : RunMyProcess Library)	RunMyProcess Library	INPUT PARAMETERS :
Green	Add a row in getting started sr Google Spreadsheets	RunMyProcess Library	Add a row in Vacation Request spreadsheet
Green	Add a row in Vacation Reques Google Spreadsheets	RunMyProcess Library	Create a new user
Green	Delete a draft of a form	Runmyprocess Internal service	In the draft url below, the task_id is in blue:
Green	Get details of a process instar	RunMyProcess - Secured Conn	Get details of a process instance [ADVANCED FEATURE]
Green	Upload file to a bucket	Amazon S3	INPUT PARAMETERS:
Green	Create a new user	Runmyprocess Internal service	Create a new user.
Green	Prepay - Sub-account balance	Simwood API v3	Returns the account number, currency and current balance of s
Green	File - get report from 'hash'	Simwood API v3	Returns a file for a report generated by another command usin
Green	Prepay - Summary of account	Simwood API v3	Generates a report showing movements on a prepay account.
Green	Prepay - Balance	Simwood API v3	Returns the currency and current balance of specified Simwoo
Green	Credit - all invoices on accoun	Simwood API v3	Returns invoices raised on the account.
Green	Credit - all unpaid invoices on	Simwood API v3	Returns unpaid invoices on the account.
Green	Credit - all paid invoices on ac	Simwood API v3	Returns paid invoices on the account.
Green	Get orders	Bigcommerce	1
Green	Daily Euro Exchange Rate	European Central Bank	Daily rate exchange for Euro
Green	List meeting	WebEx	Returns the list of your meetings.
Green	Add a row in webinar PO spre	Google Spreadsheets	Leave Auth as blank

Fujitsu RunMyProcess execution: Tasks/Todos



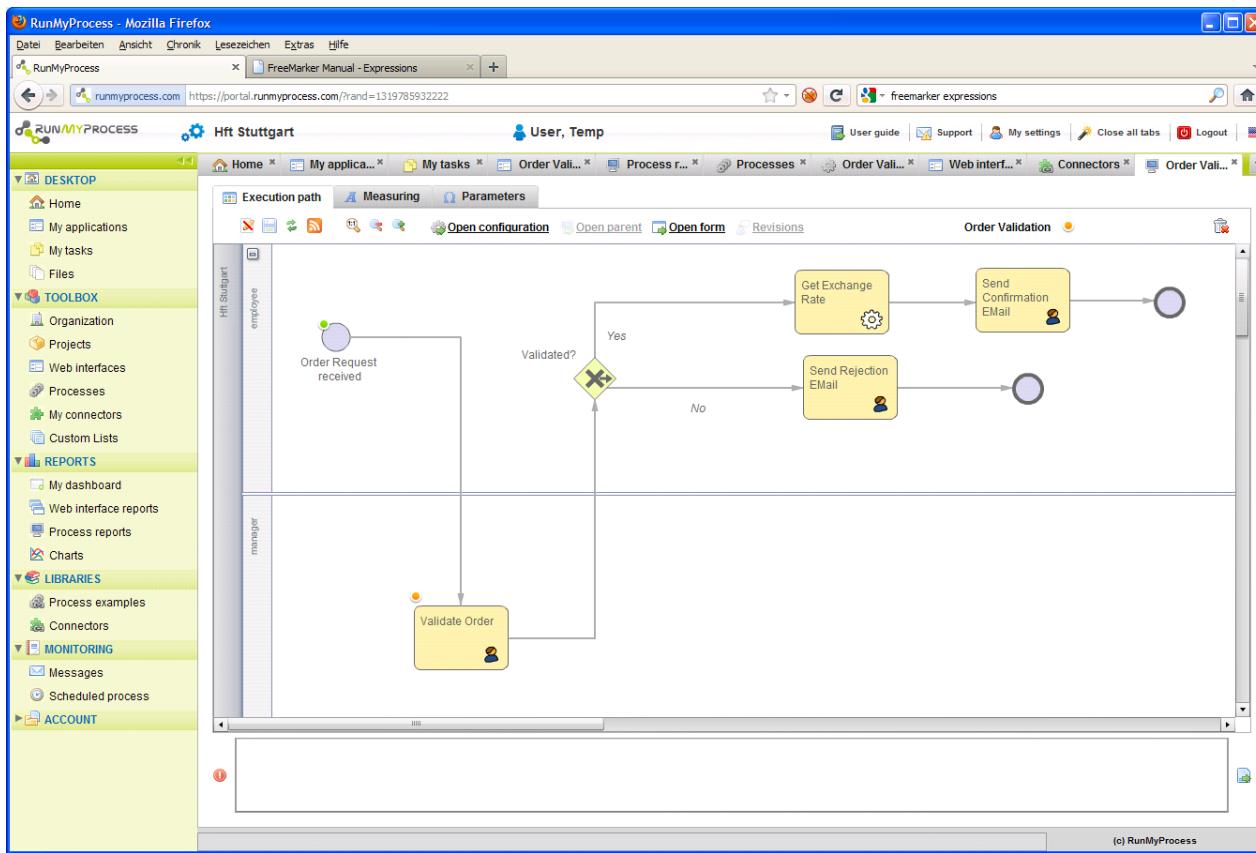
Fujitsu RunMyProcess monitoring: BAM overview

The screenshot shows a Mozilla Firefox browser window displaying the Fujitsu RunMyProcess monitoring interface. The URL is https://portal.rumyprocess.com/. The interface has a top navigation bar with links like Home, My applications, My tasks, Order Validation, Process reports, Processes, Order Validation, Web interfaces, and Connectors. On the left, there's a sidebar with categories: DESKTOP (Home, My applications, My tasks, Files), TOOLBOX (Organization, Projects, Web interfaces, Processes, My connectors, Custom Lists), REPORTS (My dashboard, Web interface reports, Process reports, Charts), LIBRARIES (Process examples, Connectors), MONITORING (Messages, Scheduled process), and ACCOUNT. The main content area is titled 'Reports' and shows a table for 'Order Validation'. The table has columns: Name, Status, Events, Created, Updated, and Delete. There are three entries in the table:

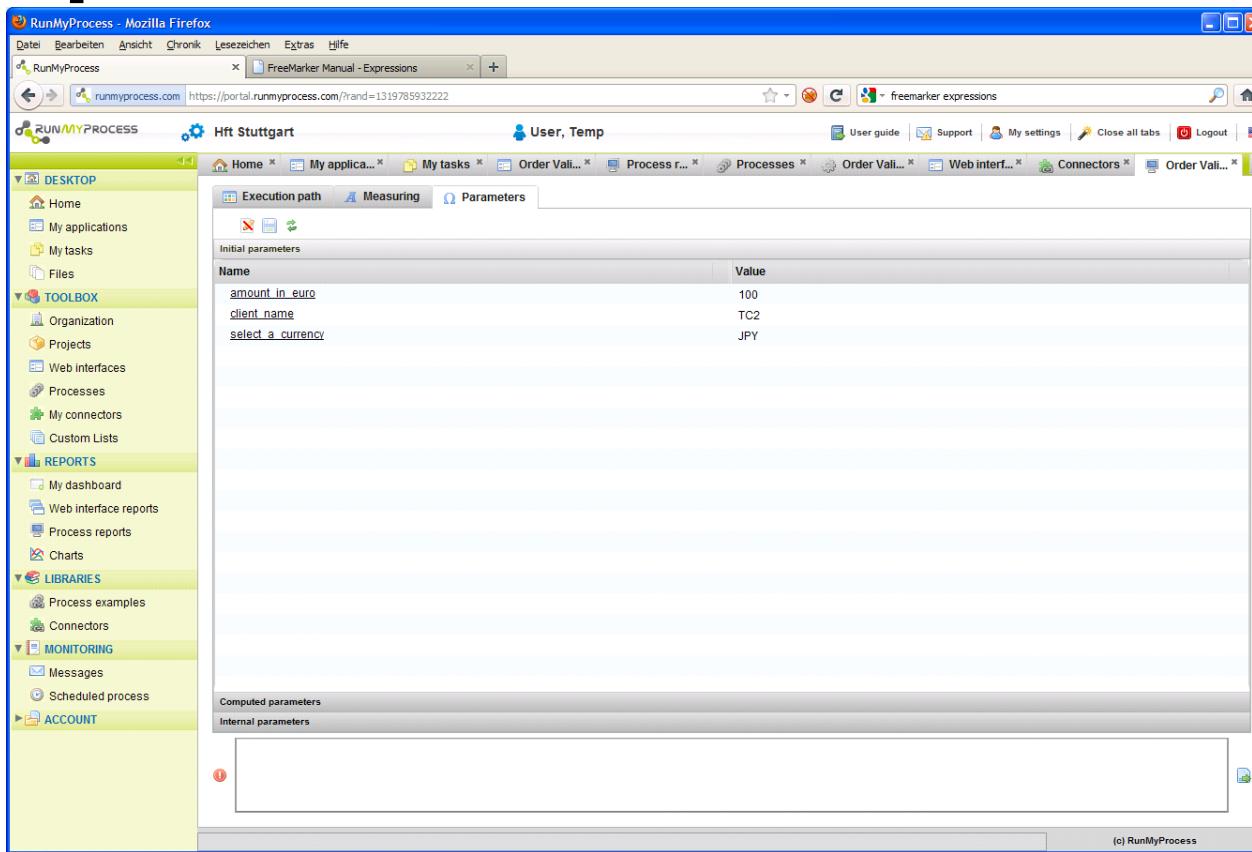
Name	Status	Events	Created	Updated	Delete
Order Validation	●		10/28/2011 9:40:20	10/28/2011 9:40:31	
Order Validation	●		10/28/2011 9:34:26	10/28/2011 9:34:52	
Order Validation	●		10/28/2011 9:33:32	10/28/2011 9:34:11	

At the bottom, there are buttons for 'Previous instances' and 'Next instances'.

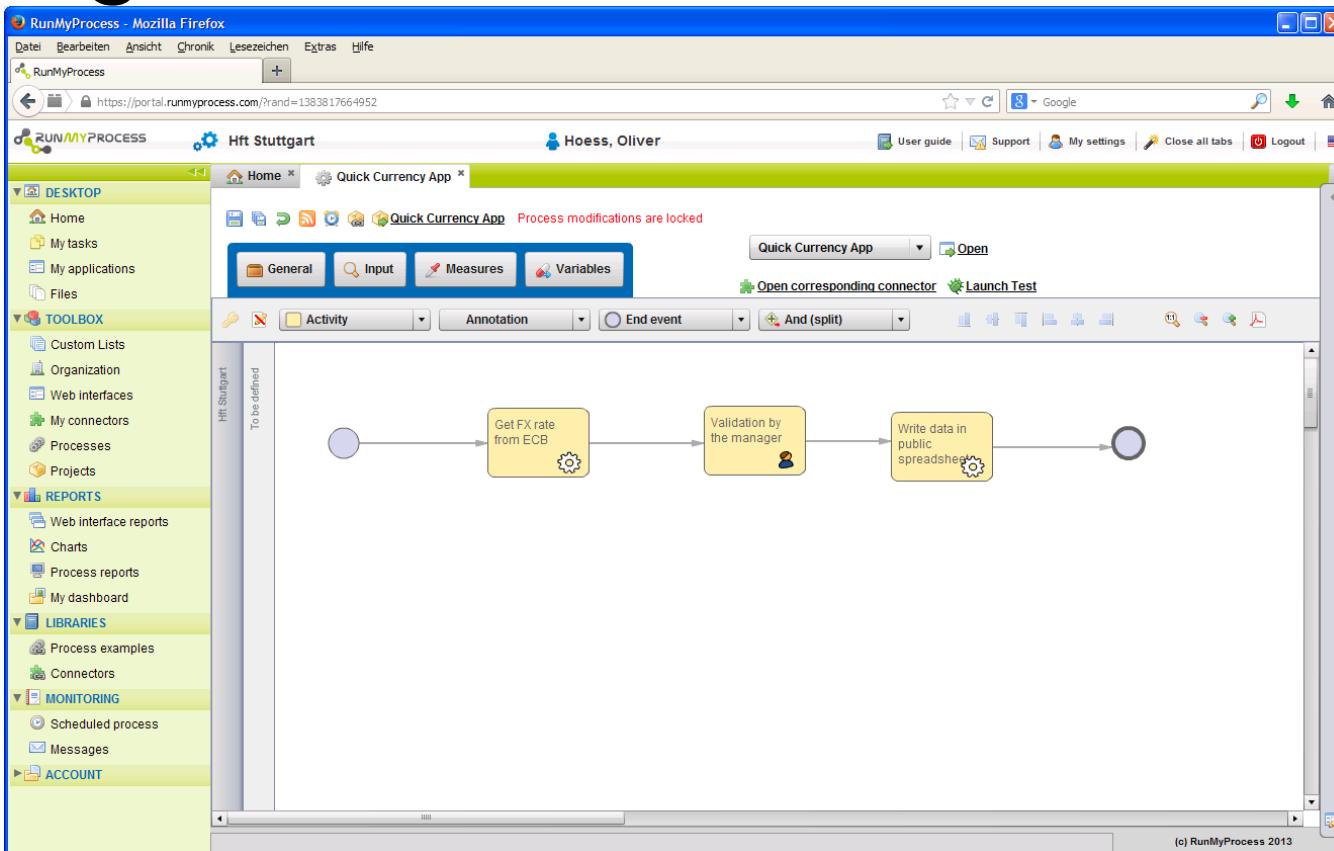
Fujitsu RunMyProcess monitoring: BAM detail view



Fujitsu RunMyProcess monitoring: BAM quantitative view



Fujitsu RunMyProcess example: getting started tutorial



Fujitsu RunMyProcess example: getting started tutorial Spreadsheet

The screenshot shows a Google Sheets spreadsheet titled "Getting started with RunMyProcess. My FX rate". The spreadsheet contains a single sheet named "Sheet1" with data starting from row 1465. The columns are labeled "Timestamp", "1 EUR =", "Currency", and "Name". The data includes various entries such as 10/28/2013 2:28:00, 1.447 USD, jehende4, Rothwell, Tom, and 11/1/2013 0:49:04, 1.447 USD, Martin, Kenneth. The "Name" column contains names like Rothwell, Tom, tom, hcchon, Gonzalez, Antonio, Payan, Ricardo, Seror, Laurent, Pezzo, Cyro, Hai Cao, and Halard, Marc-Yervant.

	Timestamp	1 EUR =	Currency	Name
1465	10/28/2013 2:28:00	1.447	USD	jehende4
1466	10/28/2013 13:49:10	1.3777	USD	Rothwell, Tom
1467	10/28/2013 13:52:56	0.8519	GBP	Rothwell, Tom
1468	10/28/2013 14:37:51	0.99	USD	tom
1469	10/28/2013 15:42:20	2	EUR	hcchon
1470	10/29/2013 21:38:29	1.447	USD	Gonzalez, Antonio
1471	10/30/2013 4:30:41	2	EUR	Payan, Ricardo
1472	10/30/2013 4:34:34	1.3768	USD	Payan, Ricardo
1473	10/30/2013 12:11:54	1	EUR	Seror, Laurent
1474	10/30/2013 15:21:23	1.3755	USD	Seror, Laurent
1475	10/30/2013 22:15:11	1.3755	USD	Pezzo, Cyro
1476	10/30/2013 22:28:01	0.8559	GBP	Pezzo, Cyro
1477	11/1/2013 0:32:02	1	EUR	Hai Cao
1478	11/1/2013 0:49:04	1.447	USD	Martin, Kenneth
1479	11/1/2013 1:00:14	0.8502	GBP	Hai Cao
1480	11/2/2013 23:37:52	1.3505	USD	bitownse
1481	11/4/2013 12:22:45	1.01	EUR	Halard, Marc-Yervant
1482	11/4/2013 14:08:24	1.357	USD	Testeur, Tst
1483	11/5/2013 10:11:30	1.3506	USD	Halard, Marc-Yervant
1484	11/6/2013 6:58:19	132.59	JPY	Hiroaki Nishimura
1485	11/6/2013 7:04:15	132.59	JPY	Hiroaki Nishimura
1486	11/7/2013 10:00:26	1.3517	USD	Shiota Nobuyuki
1487	11/7/2013 11:18:06	1.3517	USD	毛利 信浩
1488	11/7/2013 11:21:56	1	EUR	毛利 信浩
1489	11/7/2013 11:27:20	1.3517	USD	毛利 信浩
1490	11/7/2013 11:37:25	1.6	USD	Hoess, Oliver
1491				
1492				
1493				

Standards for the definition of executable process models

WSFL

Web-Services
Flow Language
(IBM)

XPDL

XML Process
Definition Language
(WfMC)

BPML

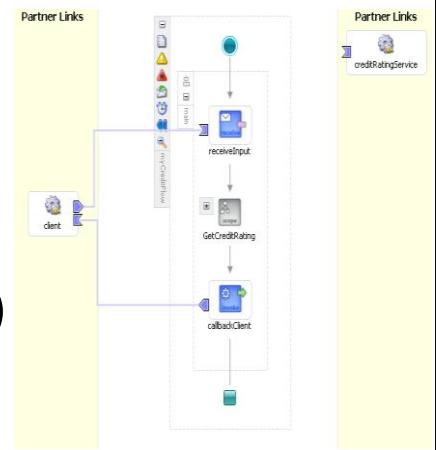
Business Process
Modeling Language
(BPMI)

XLANG

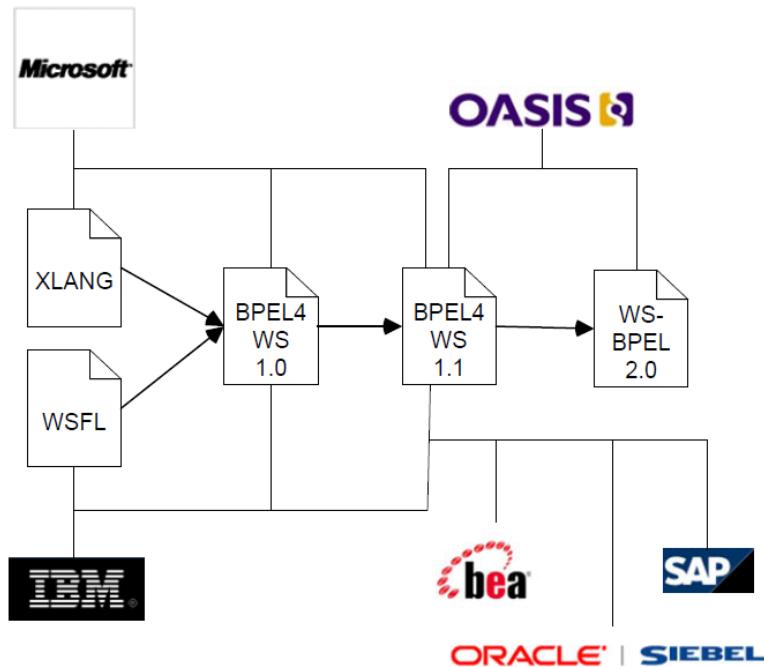
(Microsoft)

WS-BPEL – Web-Services Business Process Execution Language

- XML-based Standard for the description of executable process models (a.k.a. BPEL4WS)
- Standardized by OASIS (Organization for the Advancement of Structured Information Standards)
- Current version V2.0



BPEL: History



Development since 2002
based on existing approaches
Workflow languages existed
for about 20 years
Advantages of BPEL

- Support by important market players
- Use of Web-Service standards
 - WSDL
 - XPATH

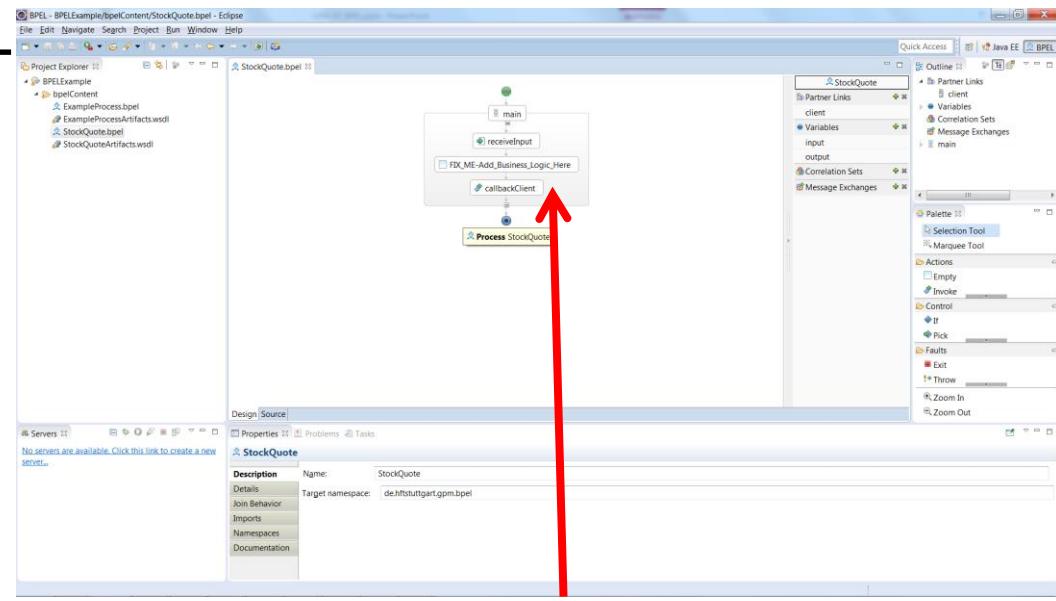
Standardization by OASIS

BPEL Elements: <process>

Root element of a BPEL process
<process name=„Stock Quote“>

— ...
</process>

Consists of
– Variables
– Partnerlinks
– Further elements



Process

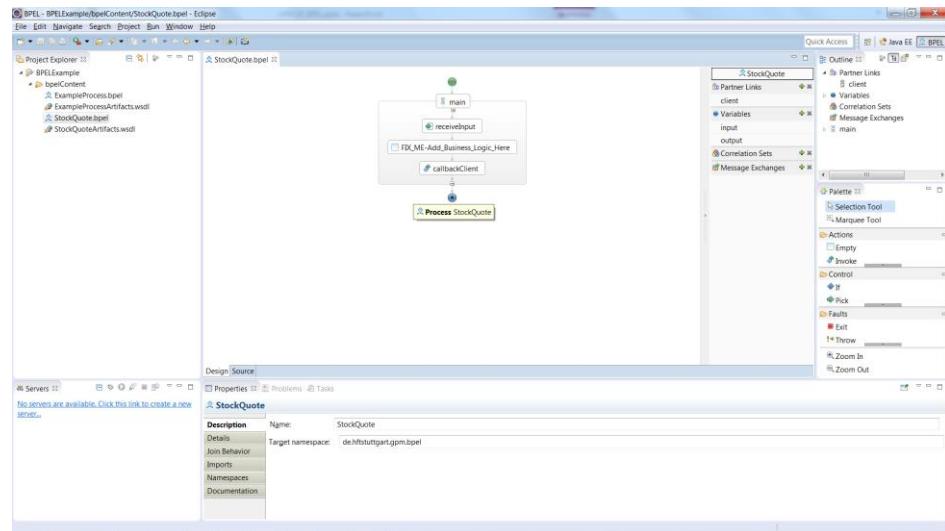
BPEL Elements: Containers

<sequence/>

- Sequential execution of all subnodes
- Normally, a process begins with a sequence

<flow/>

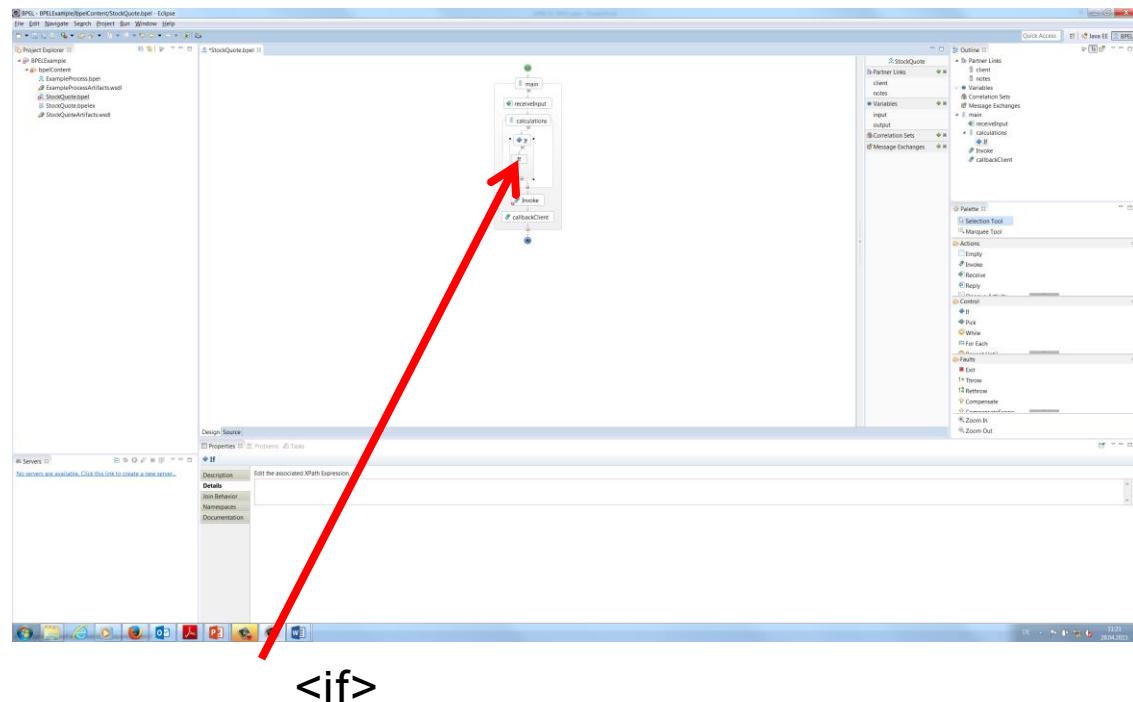
- Parallel execution of all subnodes
- <link/> for synchronization of the execution



BPEL Elements: Control Structures

- if

- <if>: Testing a condition
 - If-then-else semantics
 - Condition in Eclipse BPEL specified as an Xpath expression



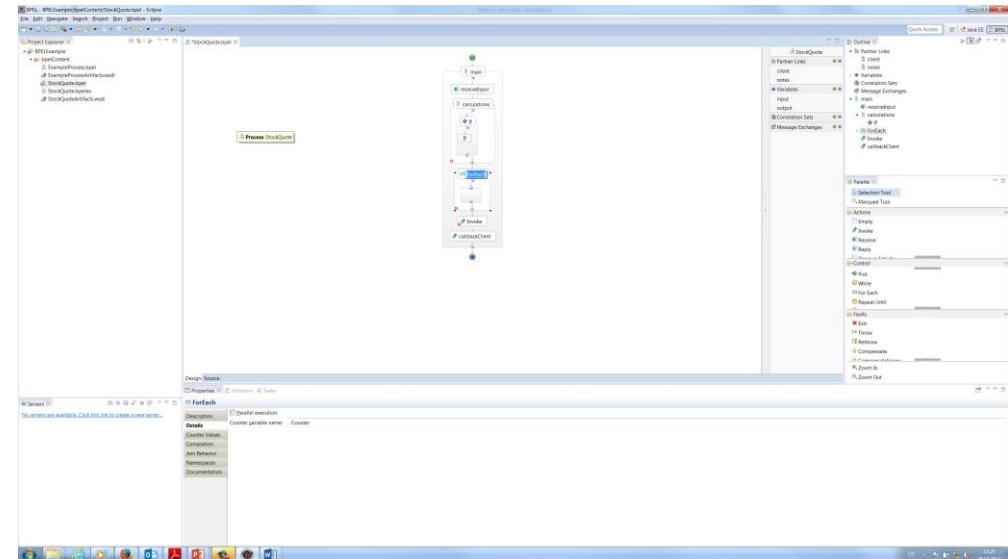
BPEL Elements: Control Structures (2) - loops

Derived from classical
programming
languages

<while>

<RepeatUntil>

<ForEach>



BPEL Elements: <partnerlink>

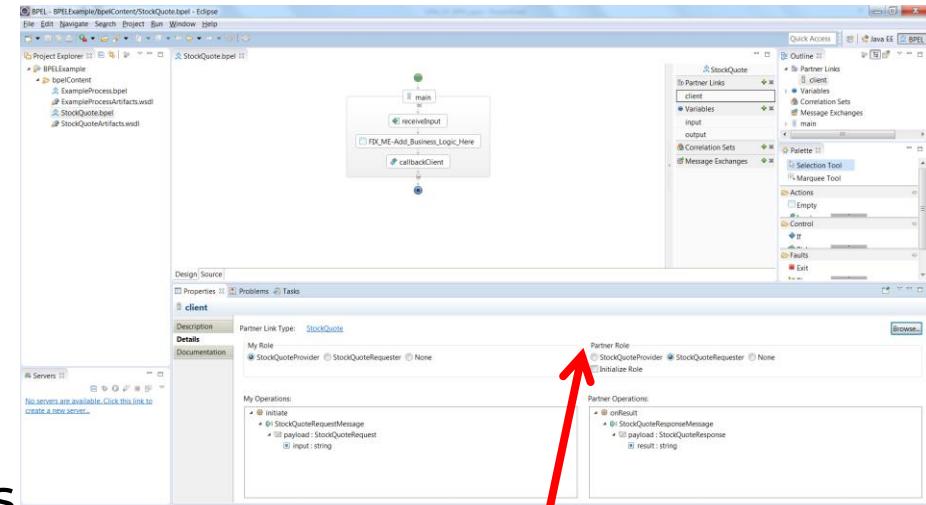
Basic Idea

- Web services are composed to higher level services by BPEL processes
- BPEL processes in turn are web services

<partnerlink>

- Denotes a web service which is used or provided by a BPEL process

Description of the interface via WSDL

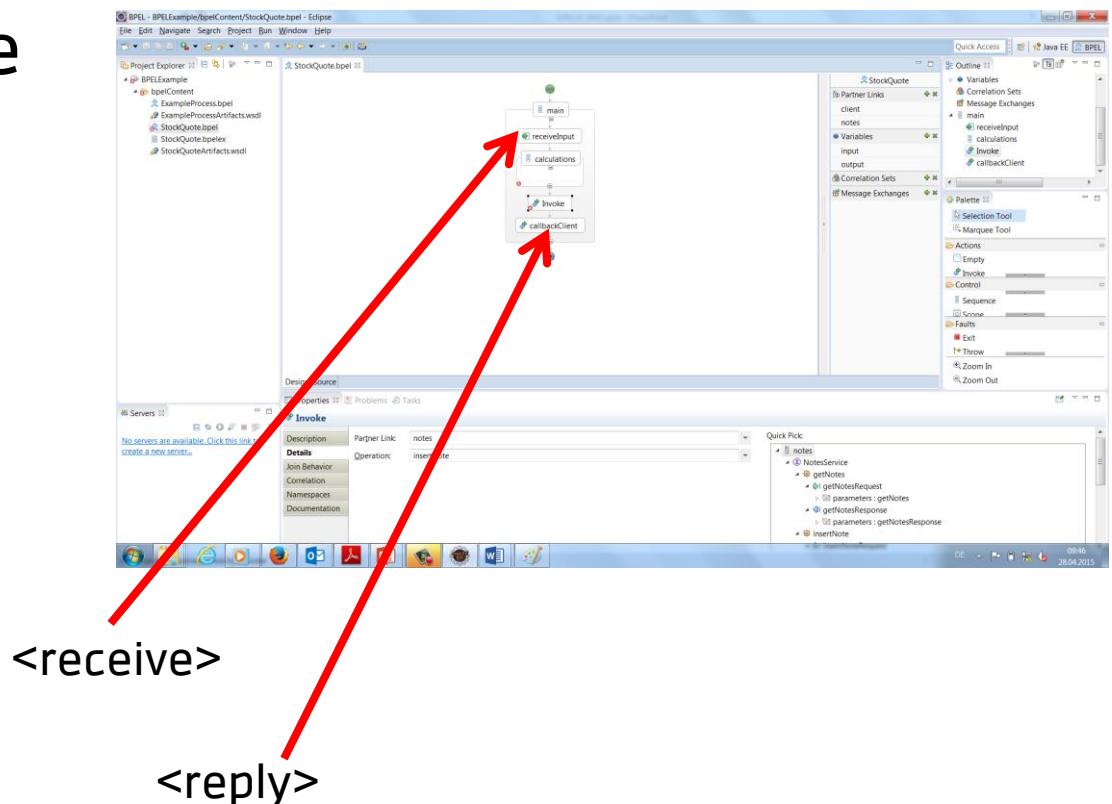


Partnerlink

BPEL Elements: Communication

<receive>: Receive request from the calling party

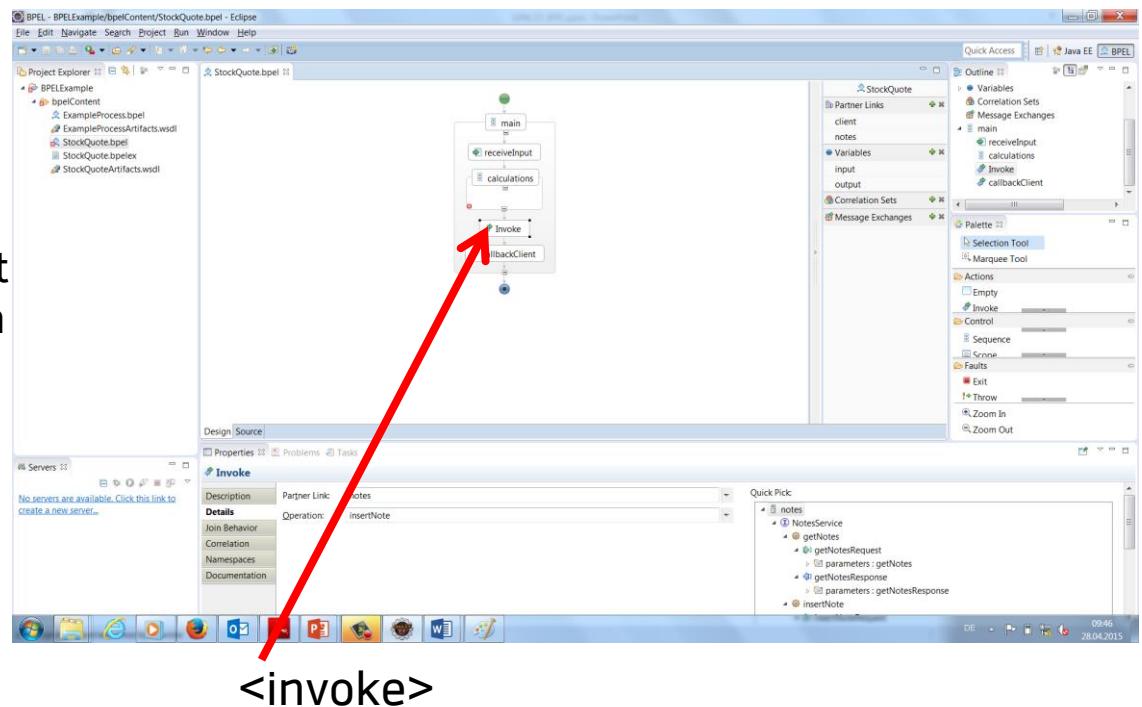
<reply>: Answer request



BPEL Elements: Communication (2)

<invoke>: Invocation of a web service

- partnerLink: Adressee, as described by the WSDL
- portType: WSDL target port
- operation: WSDL operation
- inputVariable: storage of the input parameters
- outputVariable: storage of the result values



BPEL Elements: Assignments

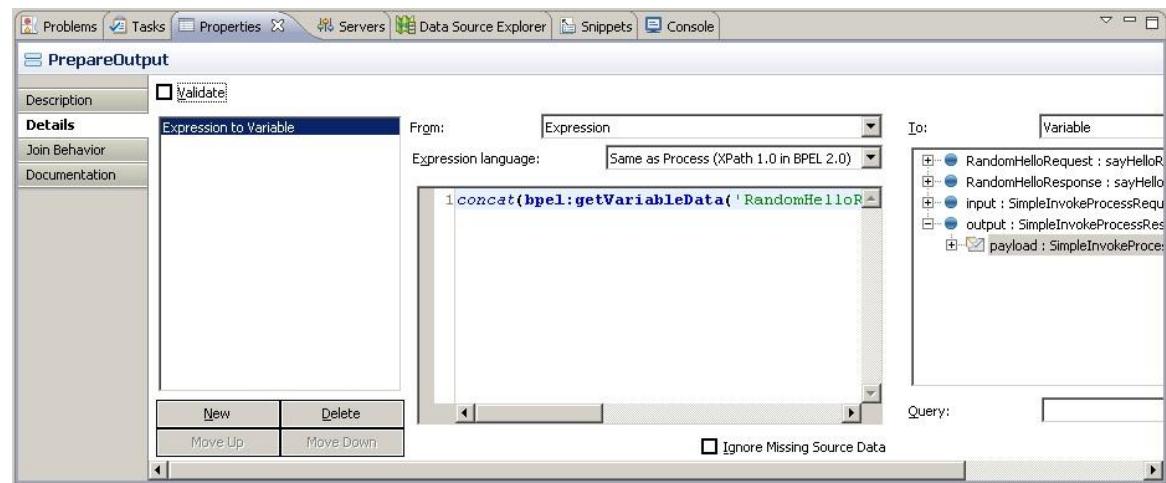
<assign>: Node for assignments

<copy>

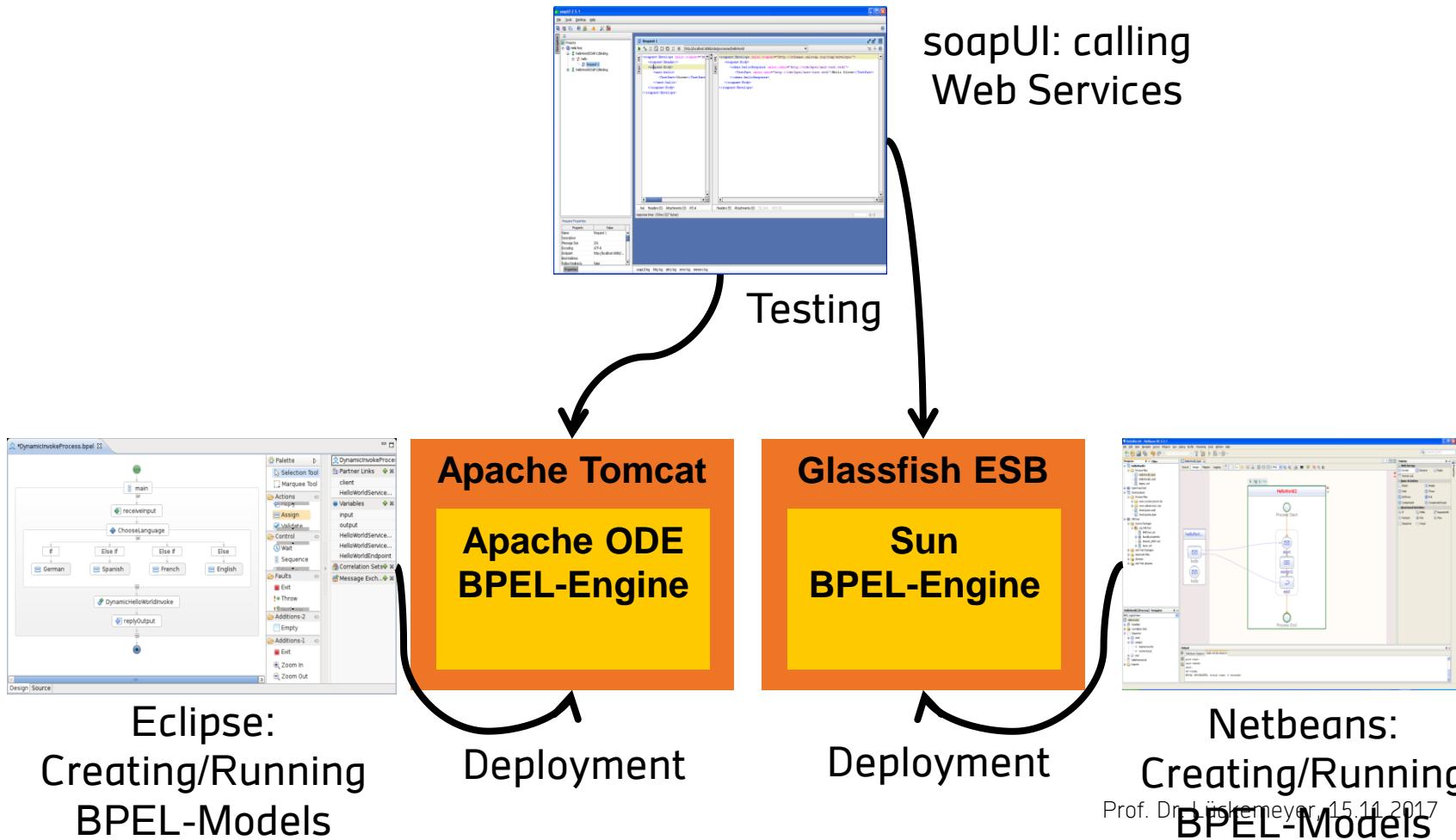
- Copy operation
- One or more <copy> operations per <assign/>

<from><to>: from ... to

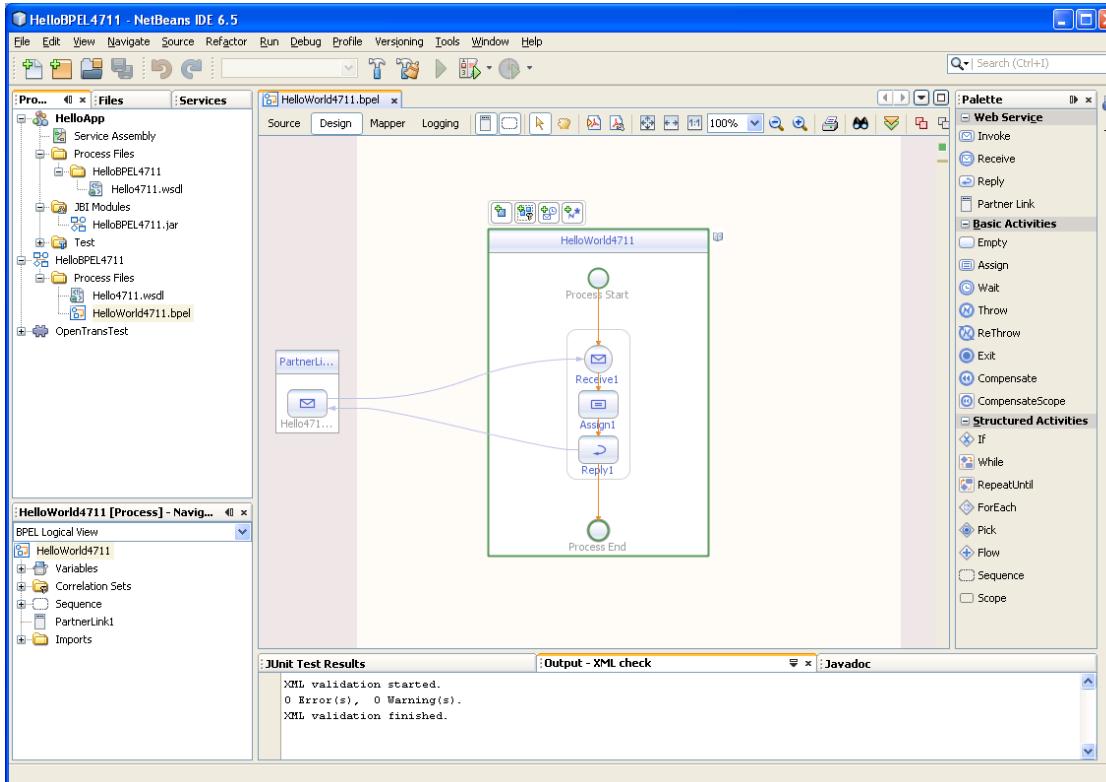
- variables: Source and destination can be variables
- Expressions / constants: Source can be an expression or a constant



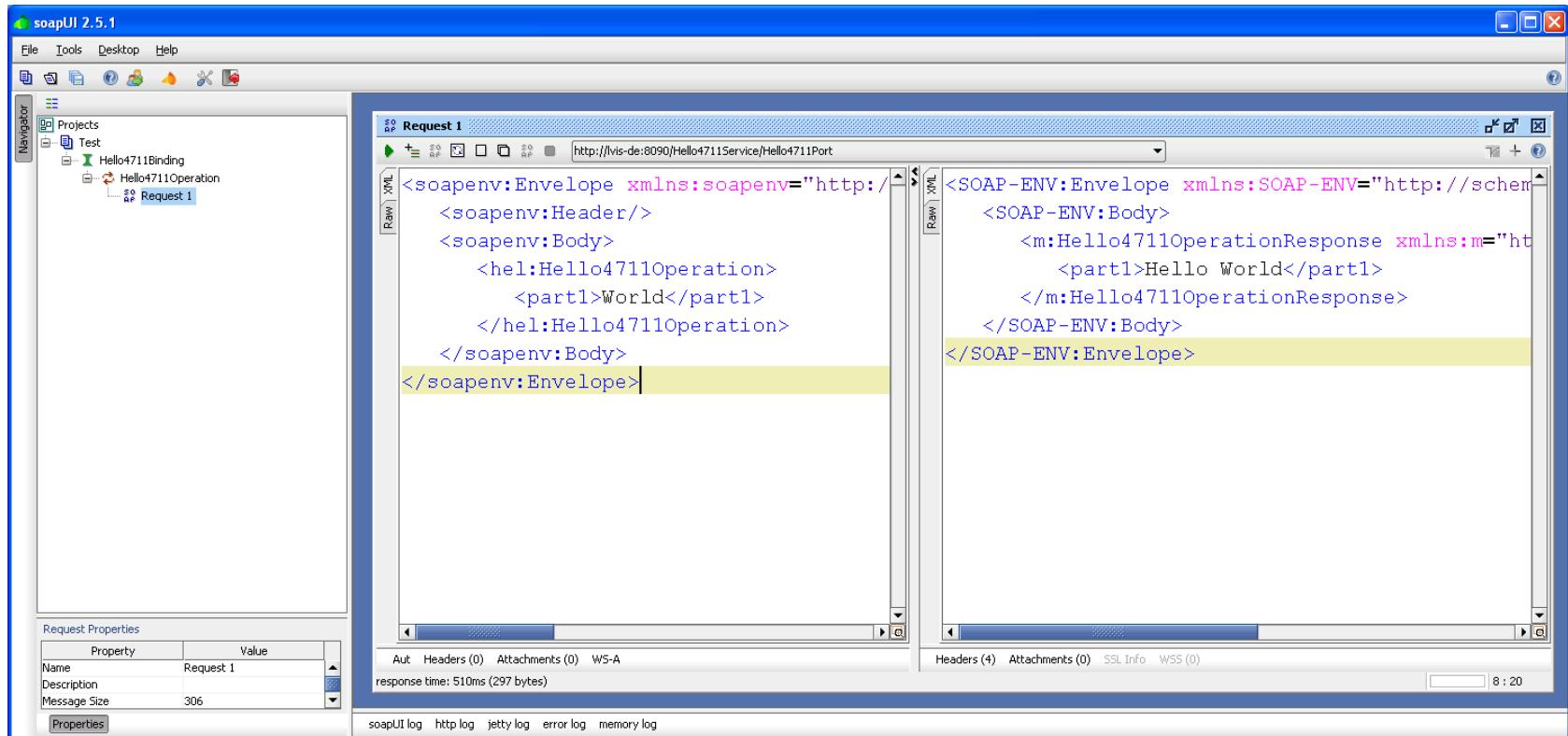
BPEL: Example technology stack



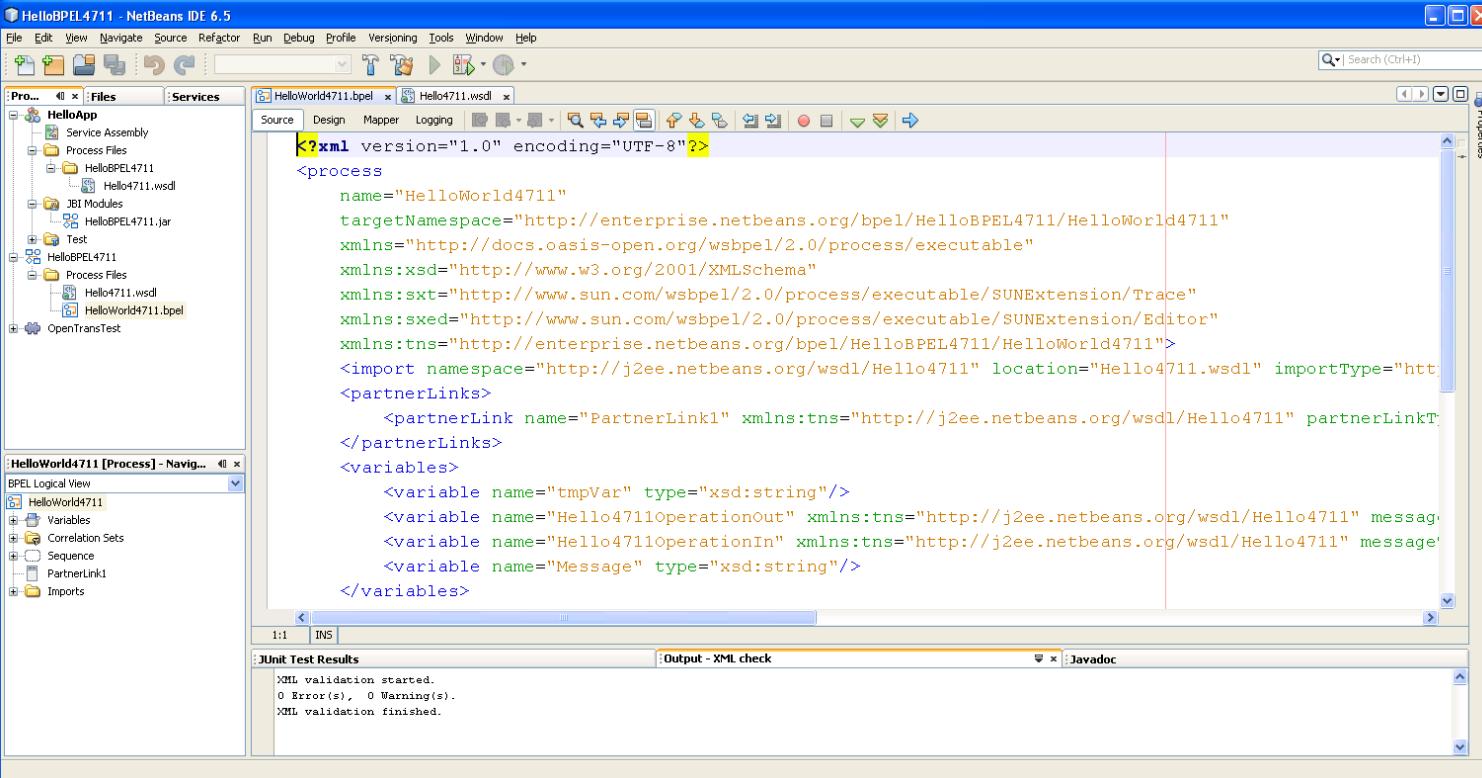
BPEL: Simple process example



BPEL: example test using soapUI



BPEL example: XML source code (1)

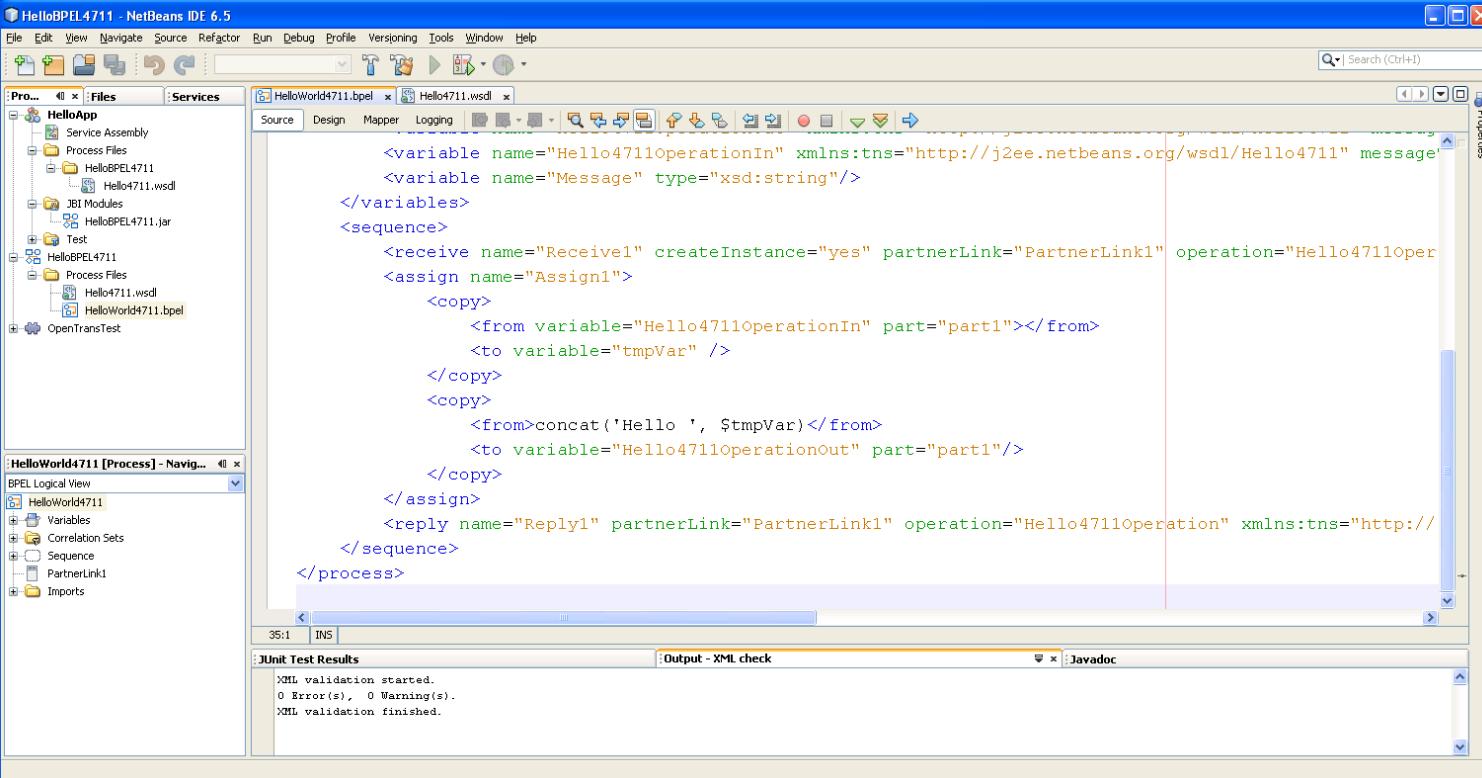


The screenshot shows the NetBeans IDE 6.5 interface with the title bar "HelloBPEL4711 - NetBeans IDE 6.5". The main window displays the BPEL source code for the process "HelloWorld4711". The code is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<process
    name="HelloWorld4711"
    targetNamespace="http://enterprise.netbeans.org/bpel/HelloBPEL4711/HelloWorld4711"
    xmlns="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:sxt="http://www.sun.com/wsbpel/2.0/process/executable/SUNExtension/Trace"
    xmlns:sxed="http://www.sun.com/wsbpel/2.0/process/executable/SUNExtension/Editor"
    xmlns:tns="http://enterprise.netbeans.org/bpel/HelloBPEL4711/HelloWorld4711">
    <import namespace="http://j2ee.netbeans.org/wsdl/Hello4711" location="Hello4711.wsdl" importType="http://www.w3.org/2001/XMLSchema-import"/>
    <partnerLinks>
        <partnerLink name="PartnerLink1" xmlns:tns="http://j2ee.netbeans.org/wsdl/Hello4711" partnerLinkType="tns:PortType"></partnerLink>
    </partnerLinks>
    <variables>
        <variable name="tmpVar" type="xsd:string"/>
        <variable name="Hello4711OperationOut" xmlns:tns="http://j2ee.netbeans.org/wsdl/Hello4711" message="tns:Hello4711OperationOut"/>
        <variable name="Hello4711OperationIn" xmlns:tns="http://j2ee.netbeans.org/wsdl/Hello4711" message="tns:Hello4711OperationIn"/>
        <variable name="Message" type="xsd:string"/>
    </variables>
</process>
```

The left sidebar shows the project structure with a node "HelloWorld4711 [Process] - Navigator" expanded, displaying "BPEL Logical View", "Variables", "Correlation Sets", "Sequence", "PartnerLink", and "Imports". The bottom status bar shows "XML validation started. 0 Error(s), 0 Warning(s). XML validation finished." The bottom right corner contains tabs for "JUnit Test Results", "Output - XML check", and "Javadoc".

BPEL example: XML source code (2)

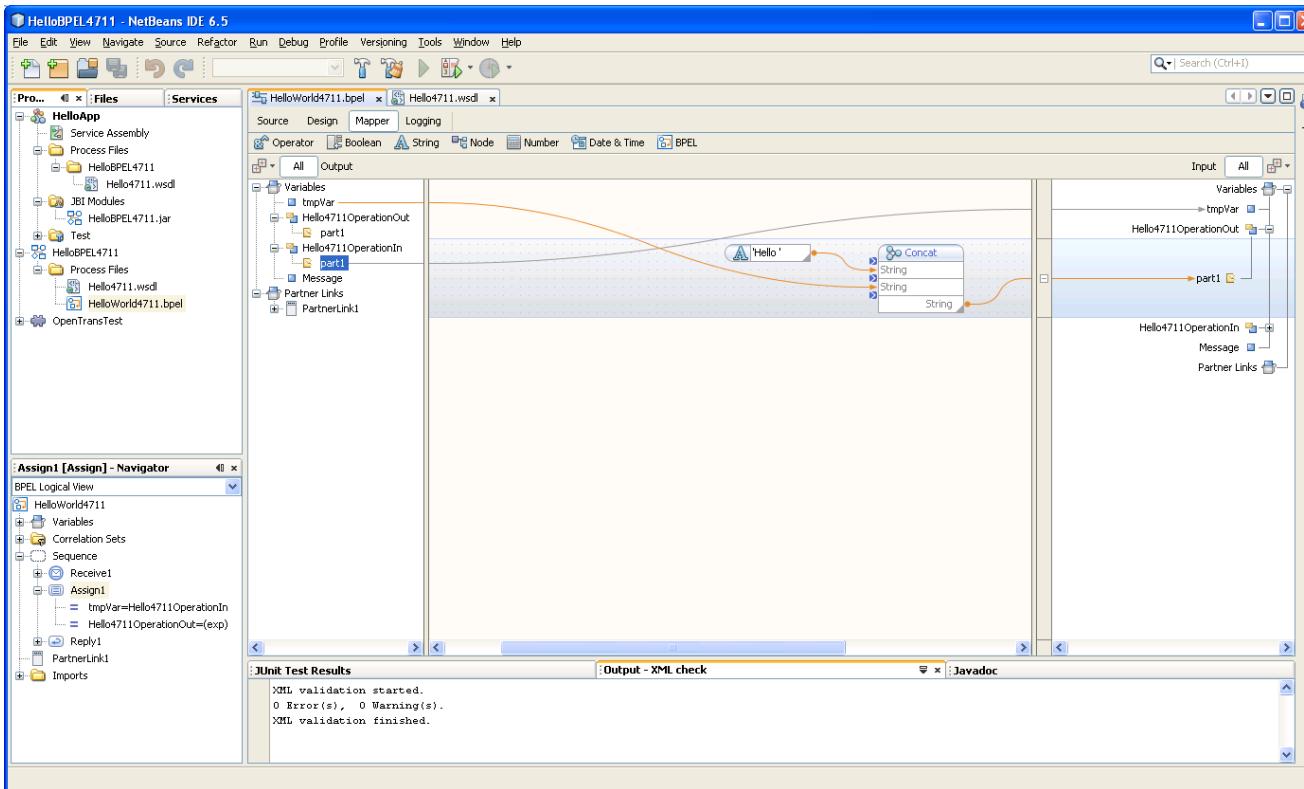


The screenshot shows the NetBeans IDE 6.5 interface with the title bar "HelloBPEL4711 - NetBeans IDE 6.5". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Versioning, Tools, Window, and Help. The toolbar has icons for New Project, Open Project, Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and others. The left sidebar shows a project tree for "HelloApp" with "HelloBPEL4711" selected, containing "Process Files" (HelloBPEL4711, Hello4711.wsdl), "JBIG Modules" (HelloBPEL4711.jar), "Test" (HelloBPEL4711, Process Files, Hello4711.wsdl, HelloWorld4711.bpel), and "OpenTransTest". Below the project tree is a "HelloWorld4711 [Process] - Navig..." node with "BPEL Logical View" expanded, showing "HelloWorld4711", "Variables", "Correlation Sets", "Sequence", "PartnerLink", and "Imports". The main editor area displays the BPEL XML source code:

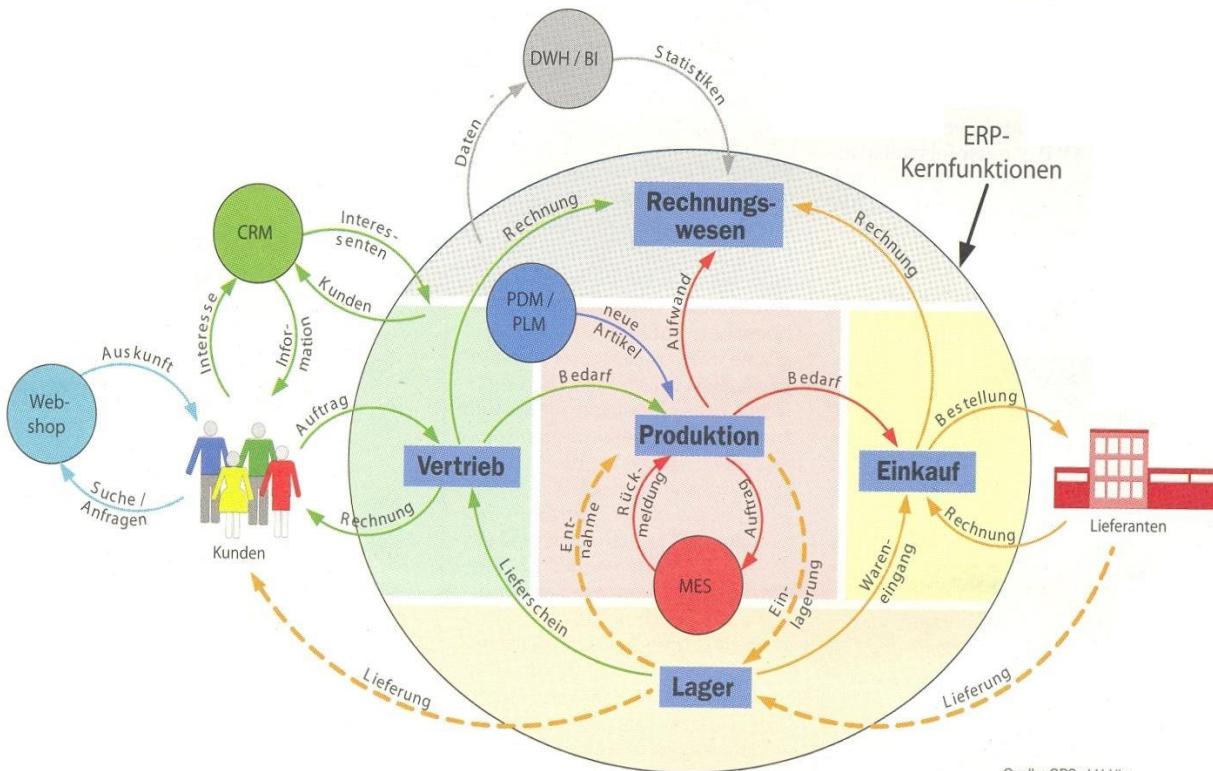
```
<variable name="Hello4711OperationIn" xmlns:tns="http://j2ee.netbeans.org/wsdl>Hello4711" message="Hello4711OperationInMessage" type="xsd:string"/>
<variable name="Message" type="xsd:string"/>
</variables>
<sequence>
    <receive name="Receive1" createInstance="yes" partnerLink="PartnerLink1" operation="Hello4711OperationIn">
        <assign name="Assign1">
            <copy>
                <from variable="Hello4711OperationIn" part="part1"></from>
                <to variable="tmpVar" />
            </copy>
            <copy>
                <from>concat('Hello ', $tmpVar)</from>
                <to variable="Hello4711OperationOut" part="part1"/>
            </copy>
        </assign>
        <reply name="Reply1" partnerLink="PartnerLink1" operation="Hello4711Operation" xmlns:tns="http://j2ee.netbeans.org/wsdl>Hello4711" type="xsd:string"/>
    </sequence>
</process>
```

The bottom status bar shows "35:1 IN5". The bottom tabs include "JUnit Test Results", "Output - XML check", and "Javadoc". The "Output - XML check" tab shows the validation results: "XML validation started.", "0 Error(s), 0 Warning(s).", and "XML validation finished."

BPEL example: Netbeans' graphical mapper

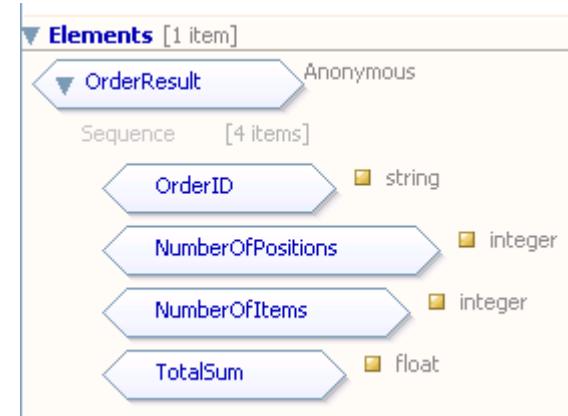
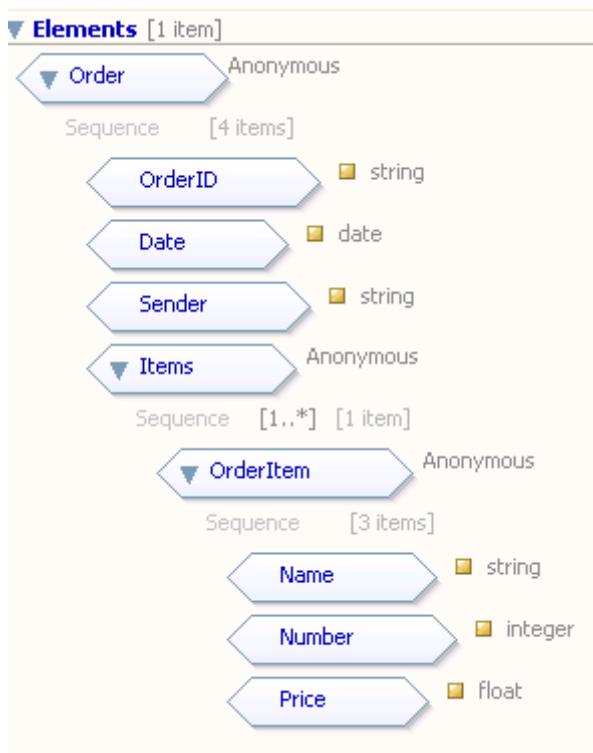


Aggregated services: reference company recap

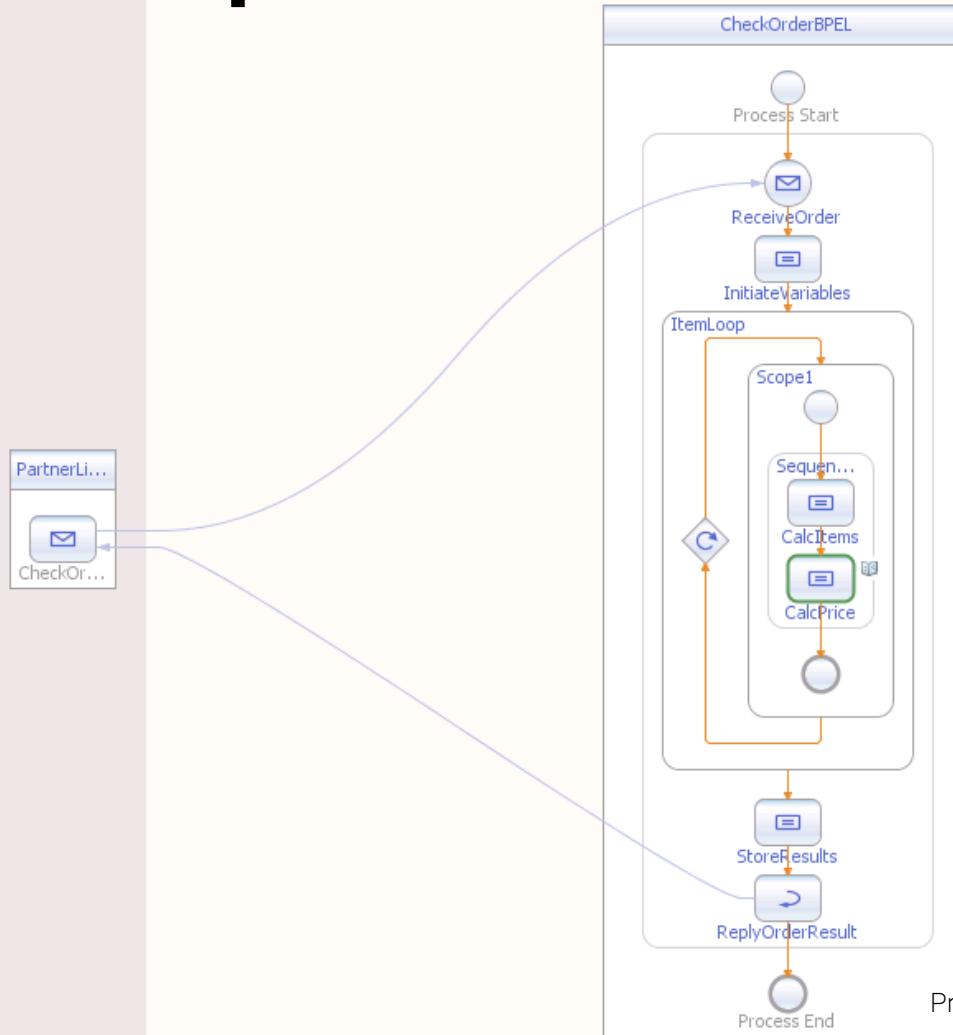


Quelle: GPSmbH-Ulm

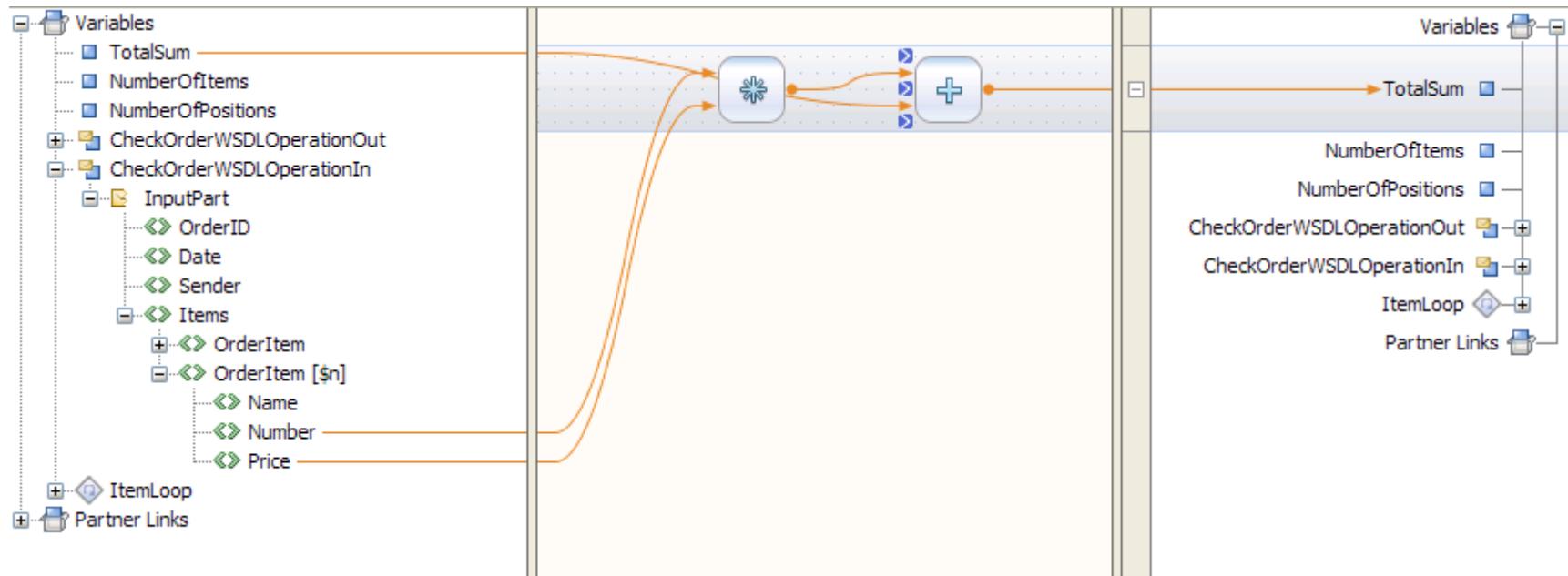
Aggregated services: order summation recap



BPEL example: order summation



BPEL example: main assignment



WSDL recap: document and rpc style

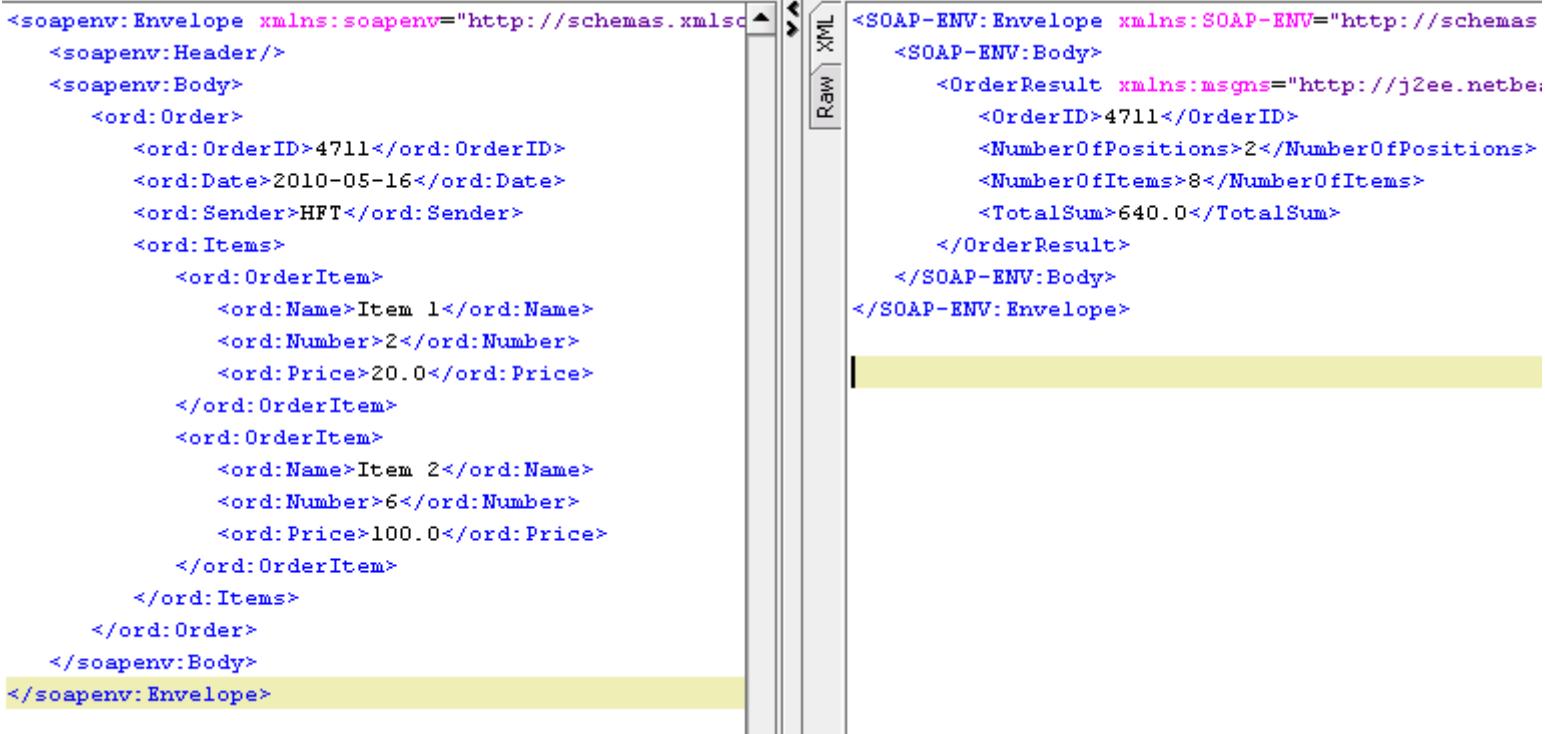
Document style (std.): Only data structure is passed

```
<env:Body>
    <m:purchaseOrder
xmlns:m="someURI">
    ...
</m:purchaseOrder>
</env:Body>
```

RPC style: Method name and parameters are passed

```
<env:Body>
    <m:placeOrder
xmlns:m="someURI">
    ...
<m:purchaseOrder>
    ...
</m:purchaseOrder>
</m:placeOrder>
</env:Body>
```

BPEL example: order summation – SOAP messages

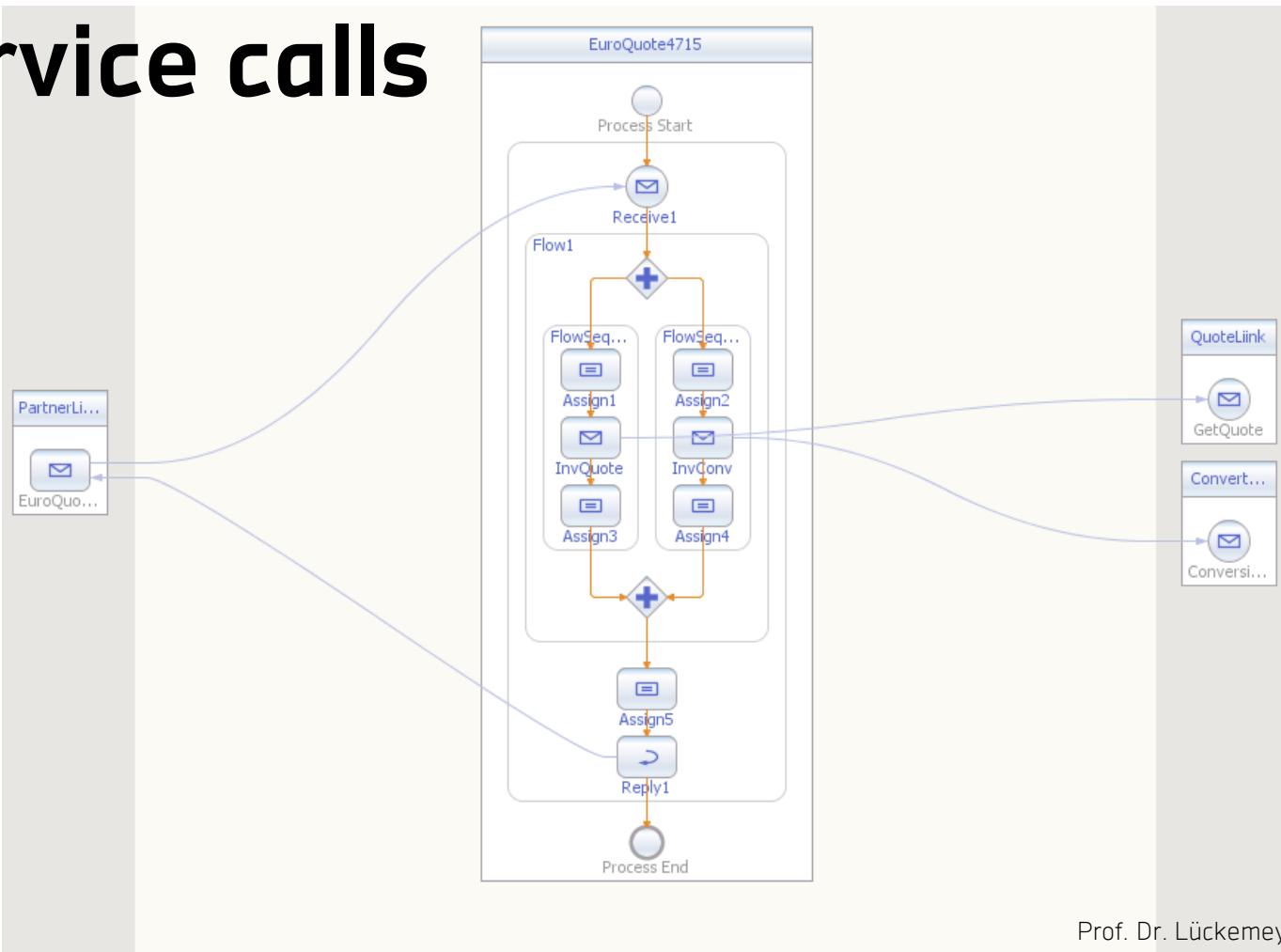


The screenshot shows a software interface for editing SOAP messages. On the left, there is a code editor window containing XML code for a request message. On the right, there is another code editor window containing XML code for a response message. Both windows have tabs for 'Raw' and 'XML'. The XML code is color-coded to highlight different elements and namespaces.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsc...
```

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas...
```

BPEL example (2): parallel web service calls



BPEL: fault handling

In distributed systems new error situations can occur

- Unavailability of a partner-service, e.g. due to system or network failure
- Wrong parameters

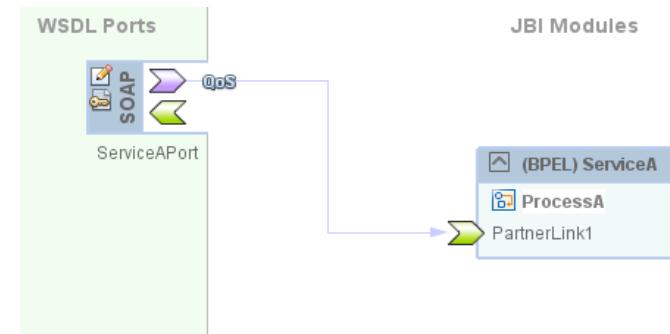
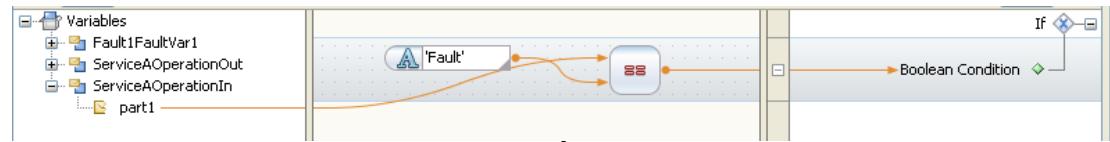
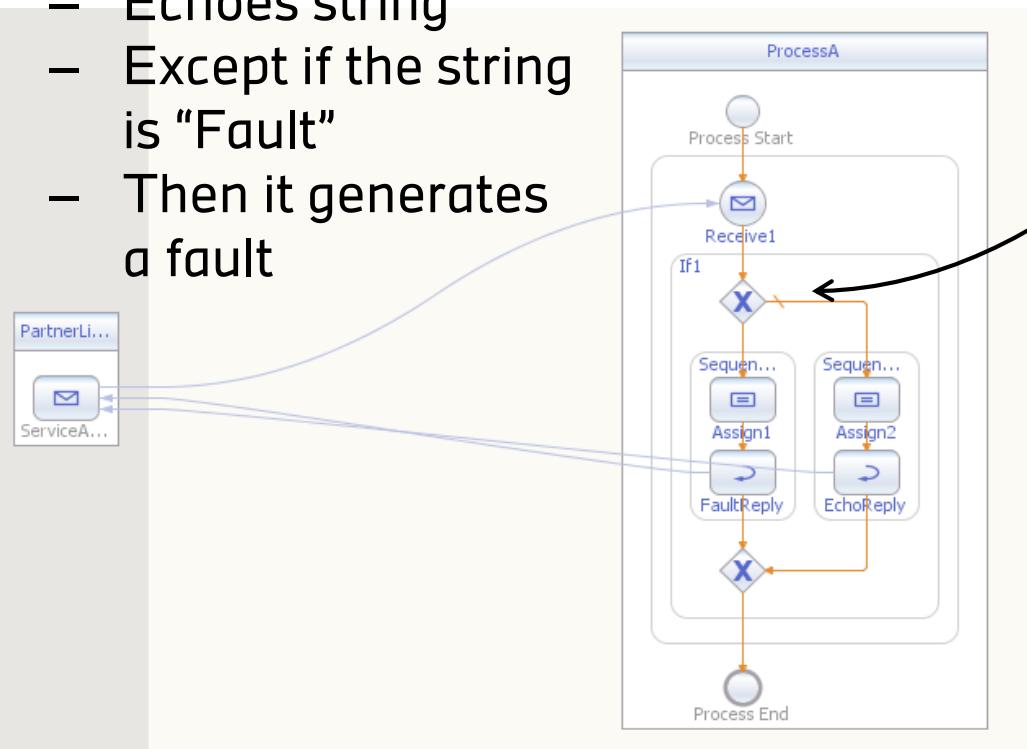
BPEL offers some constructs to deal with that fact:

- A fault type system
 - Pre-defined faults
 - User-defined faults
- Fault messages in addition to input/output parameters
- “Throw” activity to throw faults
- Fault handlers to catch faults
 - On a process level
 - On a scope level

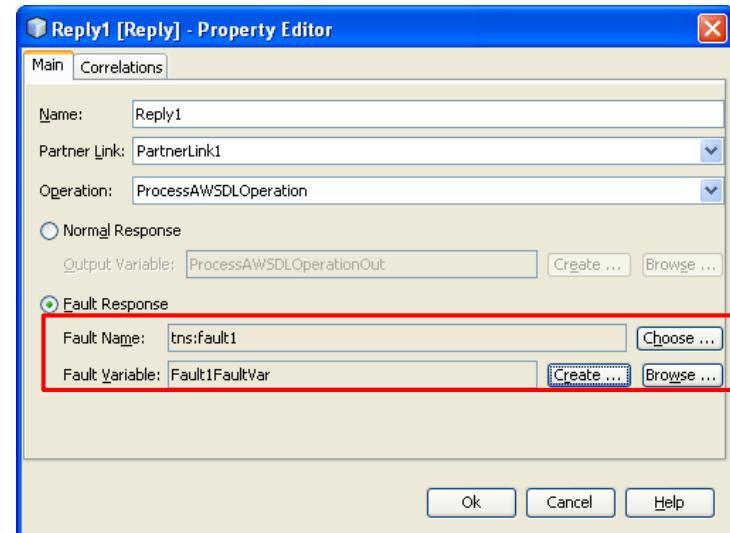
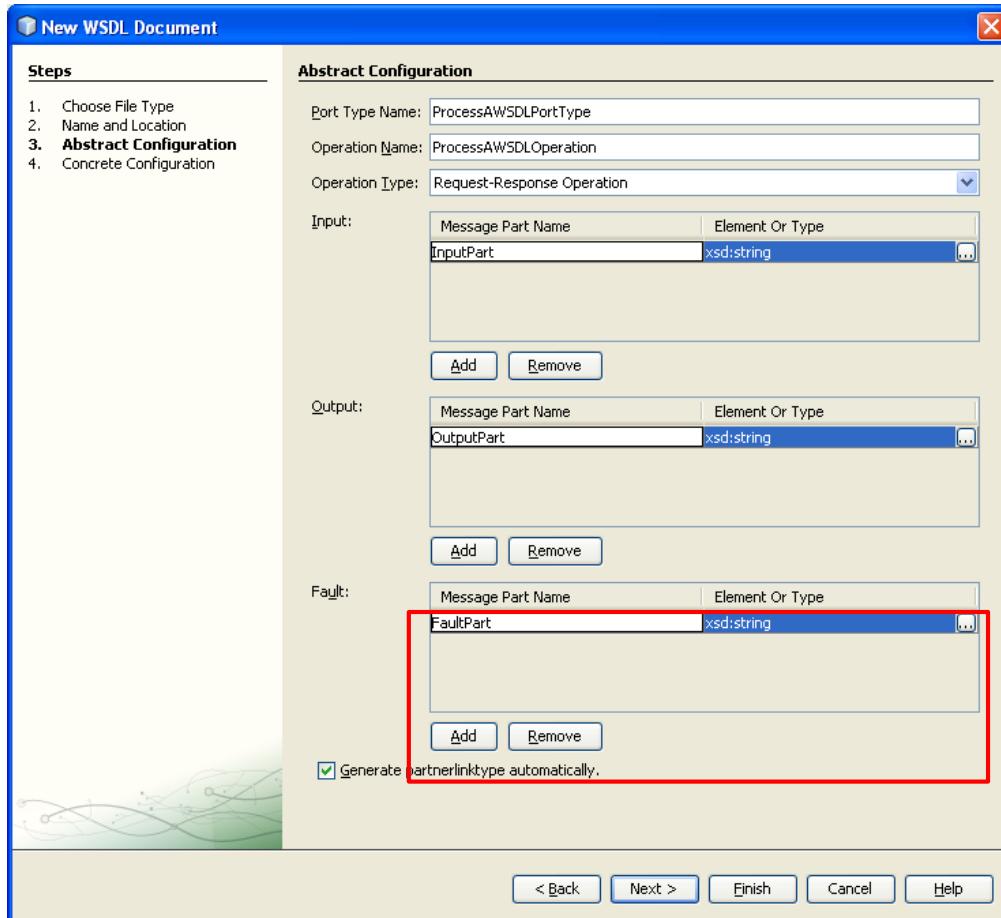
BPEL fault handling example: string echo

Process A

- Echoes string
- Except if the string is “Fault”
- Then it generates a fault



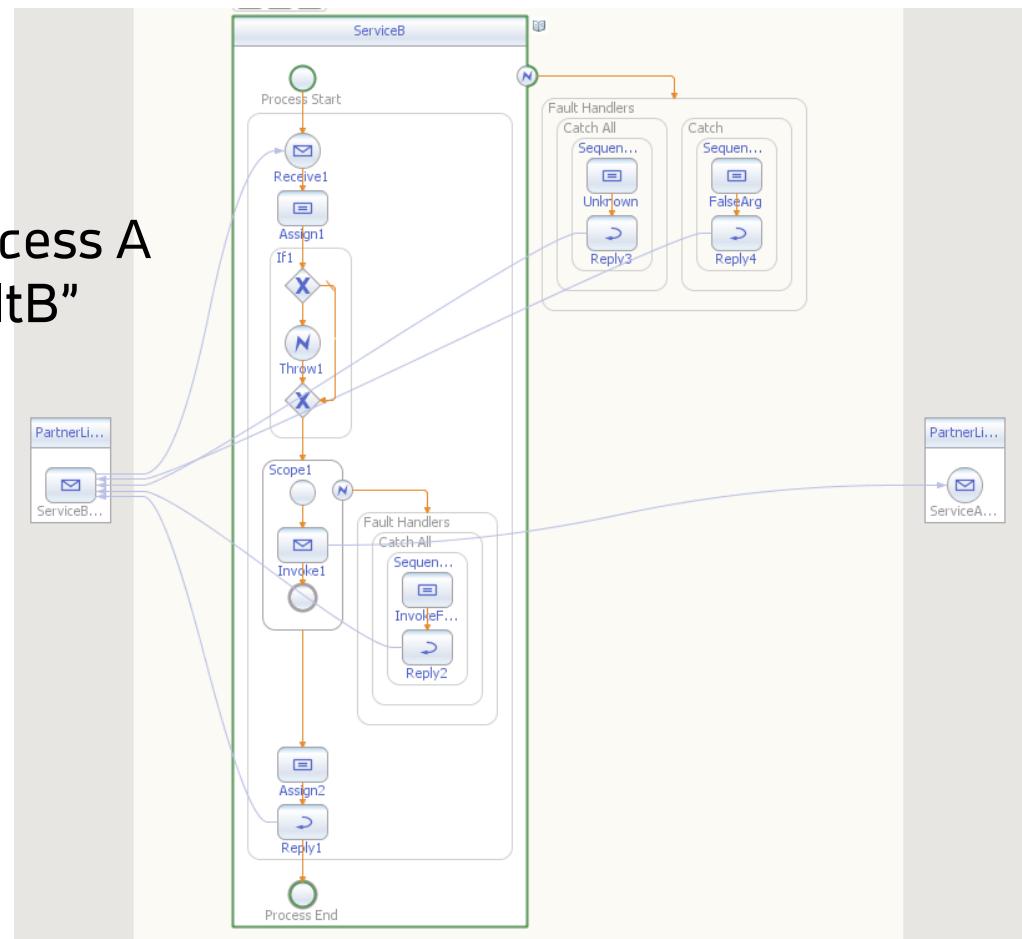
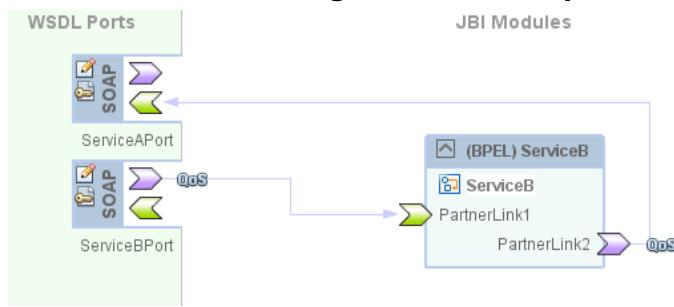
BPEL example: fault creation in the WSDL



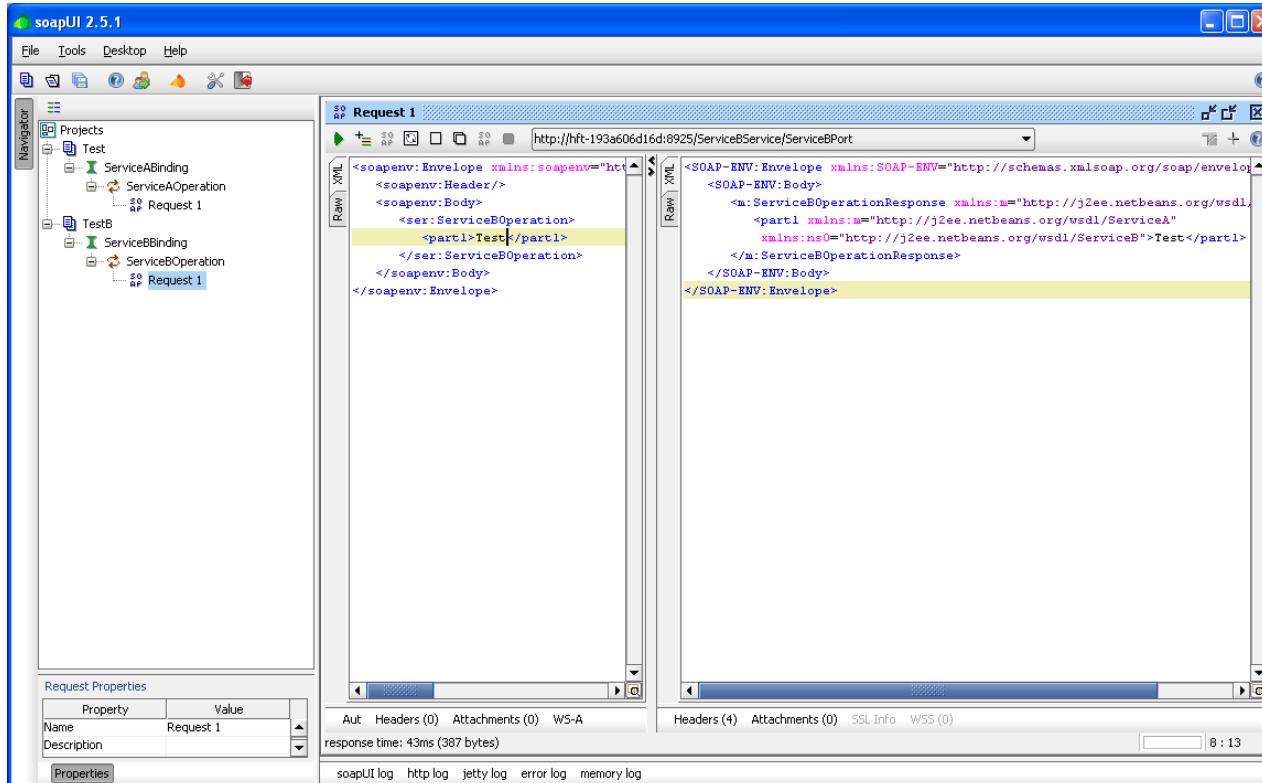
BPEL example: fault handling

Process B

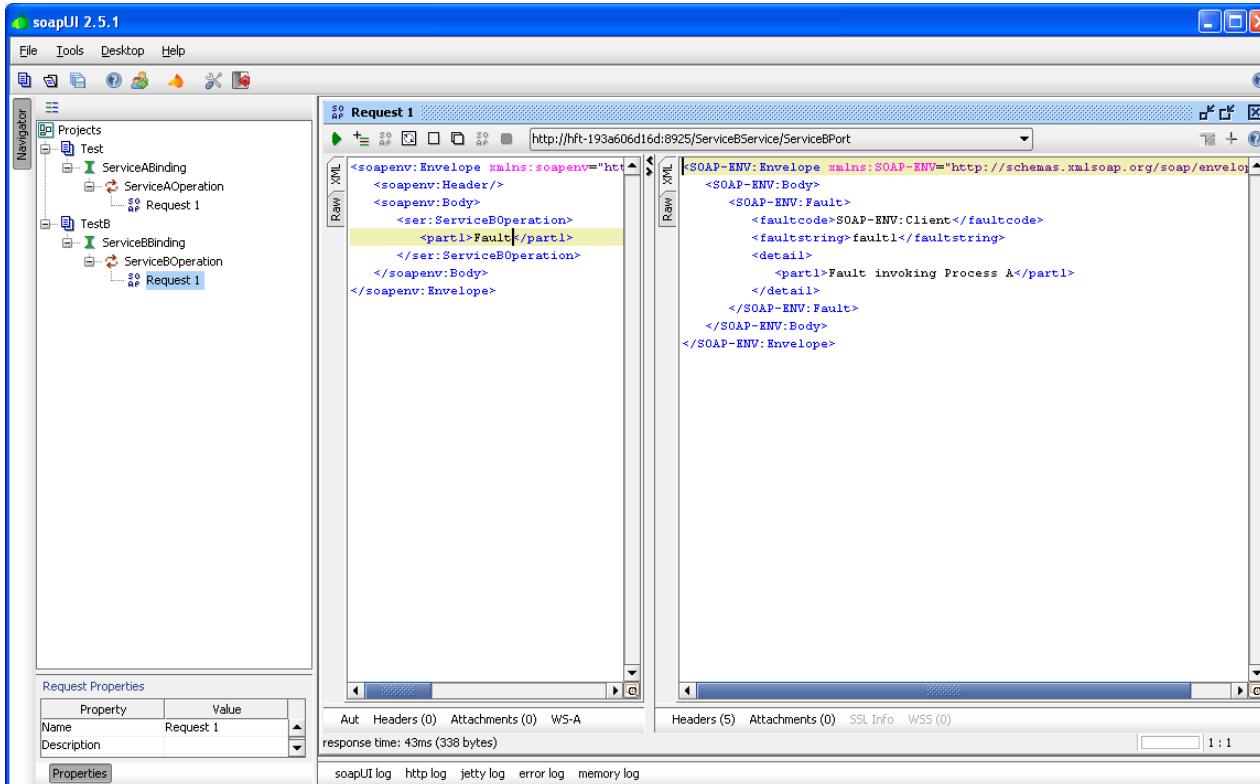
- Echoes string by using process A
- Except if the string is “FaultB” or A generates a fault
- Throw: Throwing Faults
- Catch: Catching Faults on a local or global scope



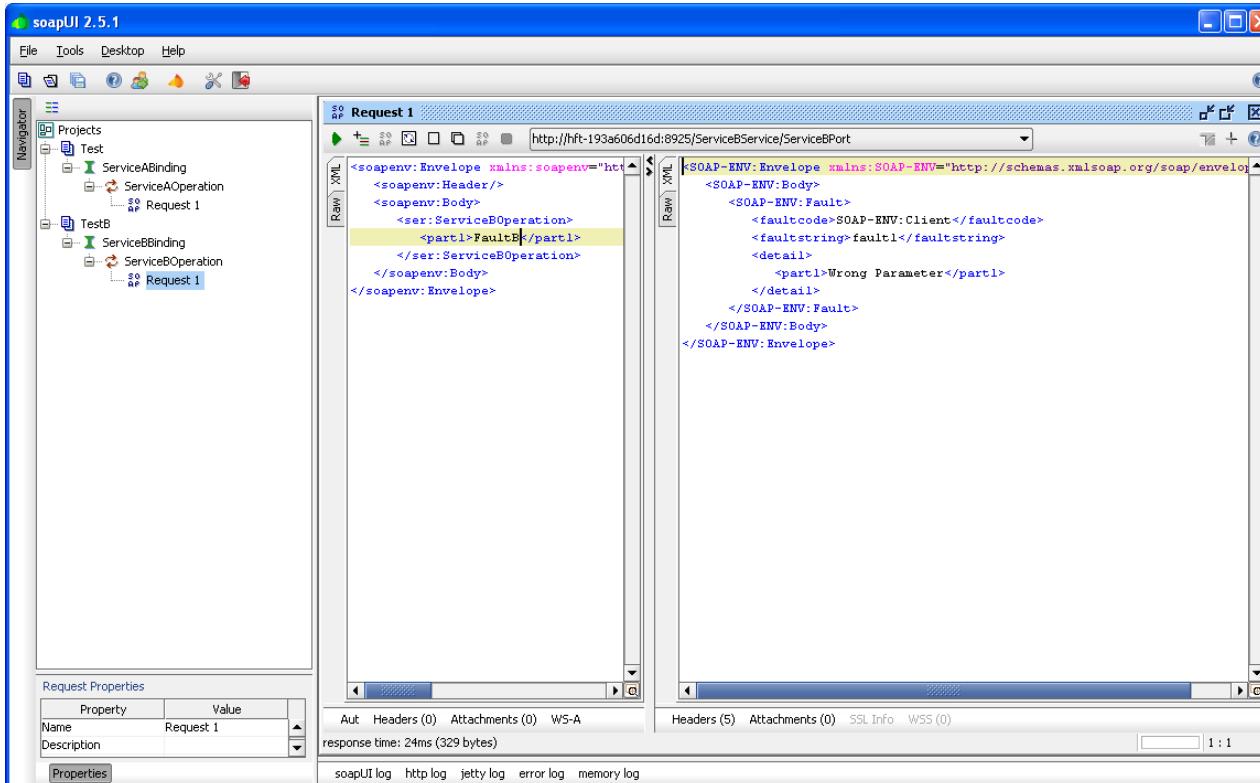
BPEL example: no fault



BPEL example: fault by process A



BPEL example: fault by process B



Long running processes

Until now our BPEL processes were relatively short running (milli-/microseconds)

We did not care about

- Identifying specific instances
- Instances vs. process templates
- Storing the state (persistence)

In reality many processes are “long running”

- E.g. a credit approval process (multiple days/weeks/months)

Therefore

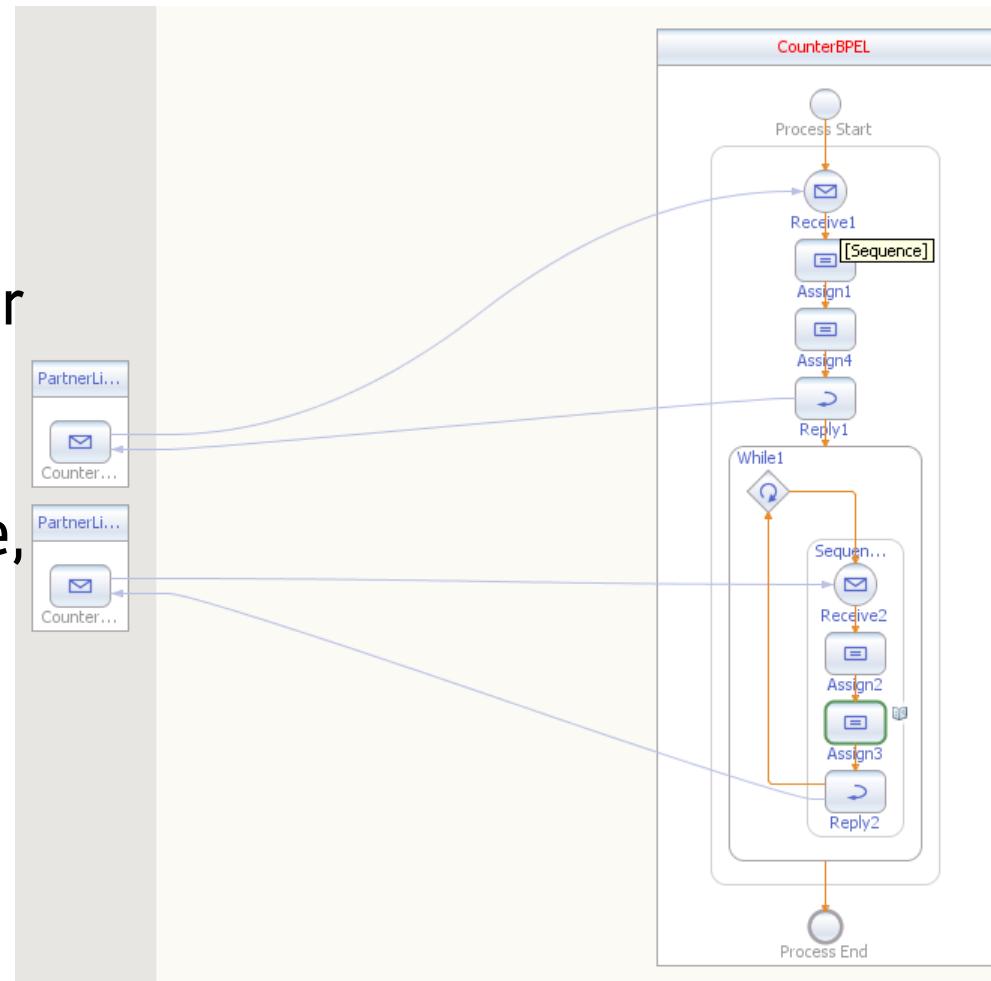
- messages have to be sent to specific instances (correlation)
- Processes have to be persistent to survive system shutdowns

BPEL example: counter

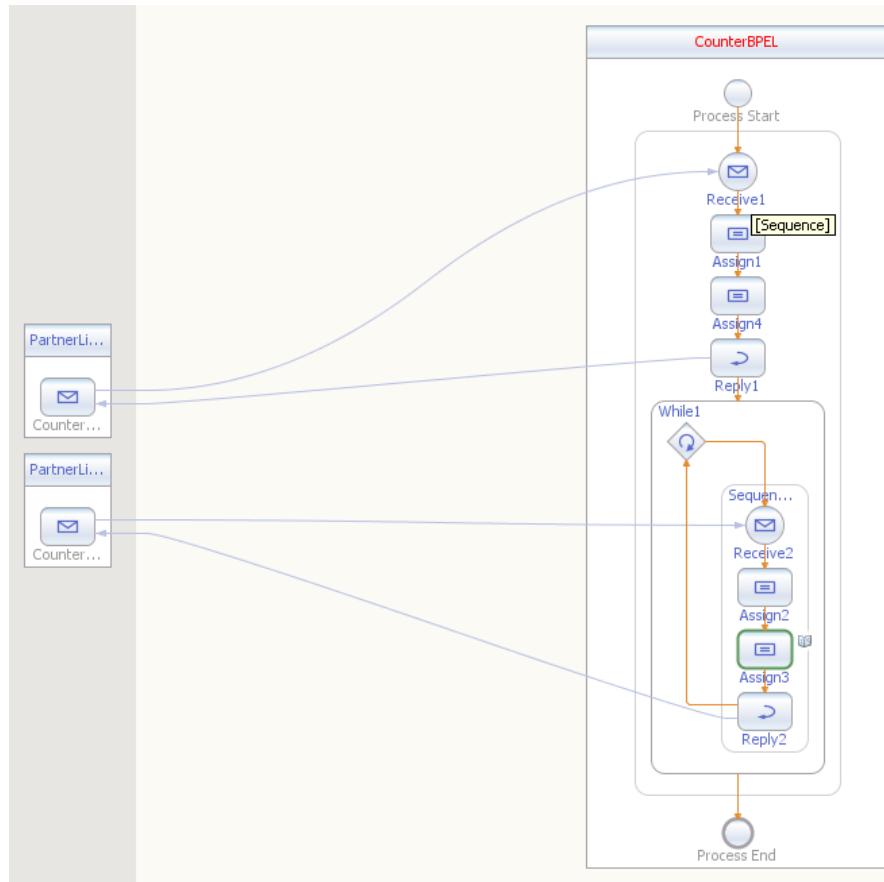
Partnerlinks

- Create: Creates counter with a name and initializes it to 0
- Get: Gets counter value, Increases it by 1
Identified via name

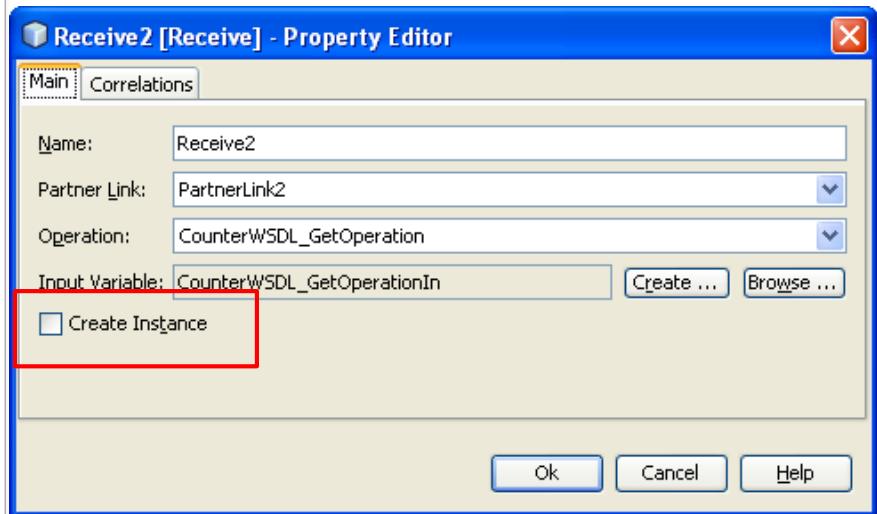
Identification of Instances using correlations



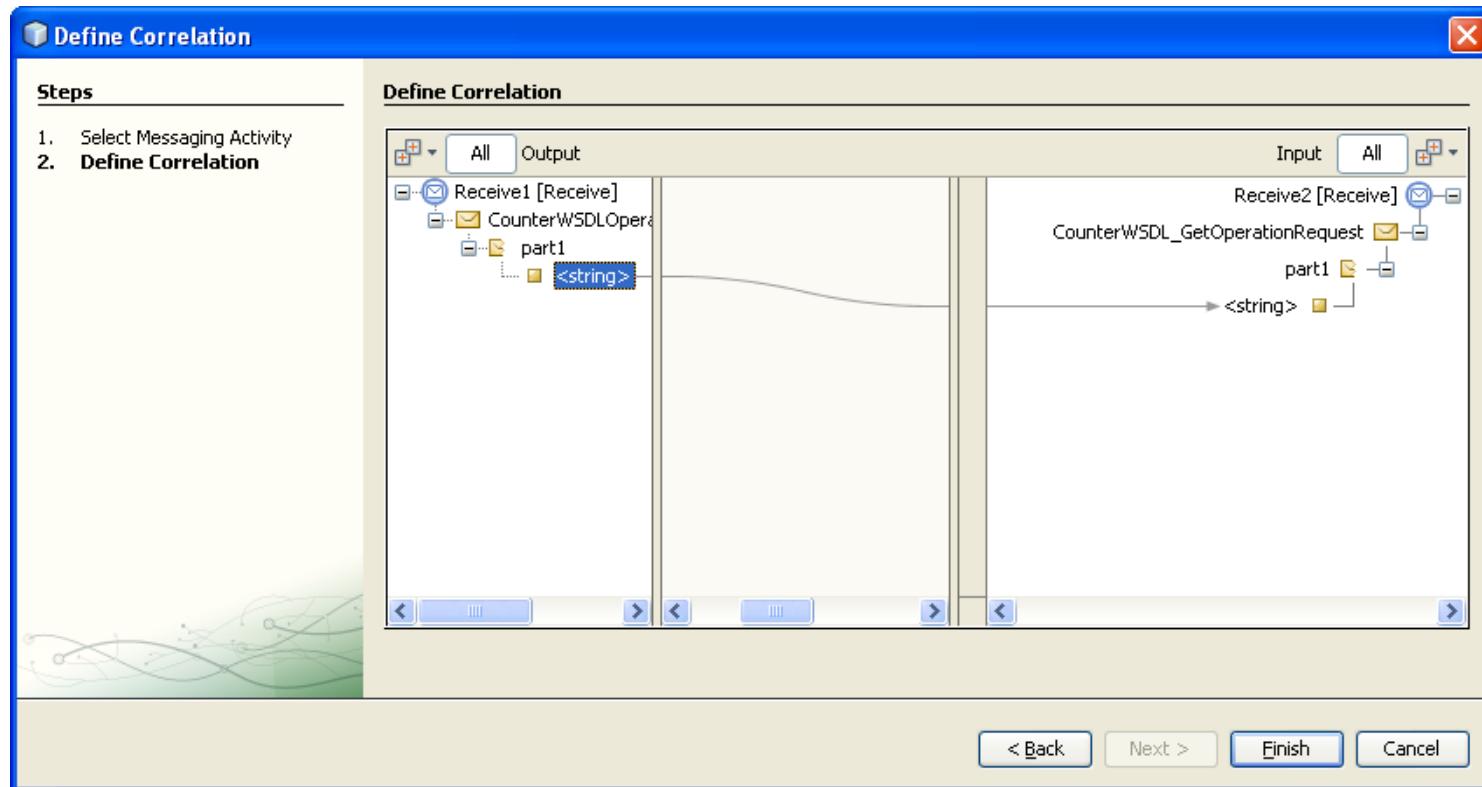
BPEL example: counter



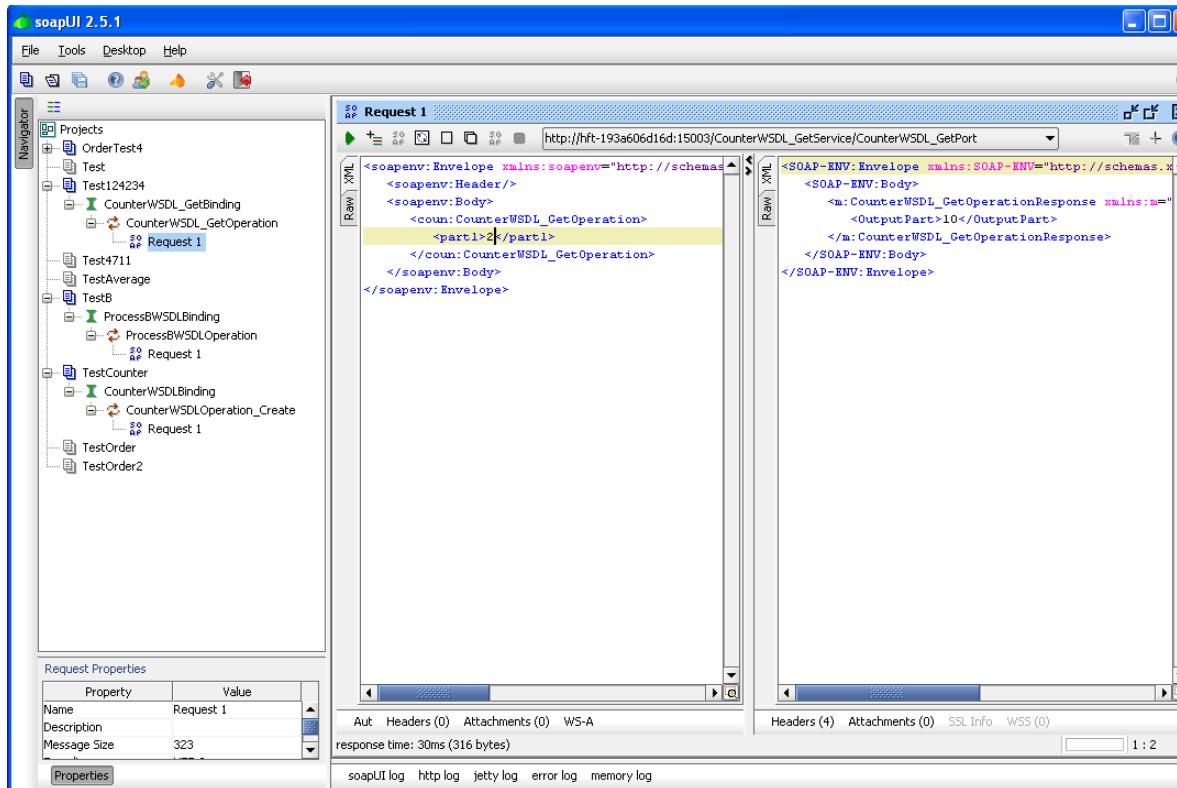
On the second receive no new instance is created



BPEL example: counter (3) – defining the correlation



BPEL example: counter (4) – testing with soapUI



Summary

- ✓ Workflows in the BPM cycle, background and systematics explained
- ✓ SaaS-Workflow Management example presented
- ✓ WS-BPEL as an international workflow management standard explained with its main constructs
- ✓ Long running instances and correlations highlighted

Questions? Questions!

**THANK YOU VERY MUCH FOR
YOUR ATTENTION!**