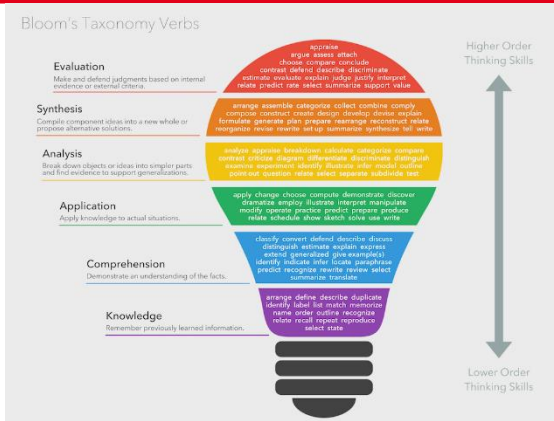
Master Software Technology
Business Process Technology – [08] Workflows

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)

Bloom's Taxonomy Verbs
by Fractus Learning,
Lizenz: CC-BY-SA 4.0



Learning Goals

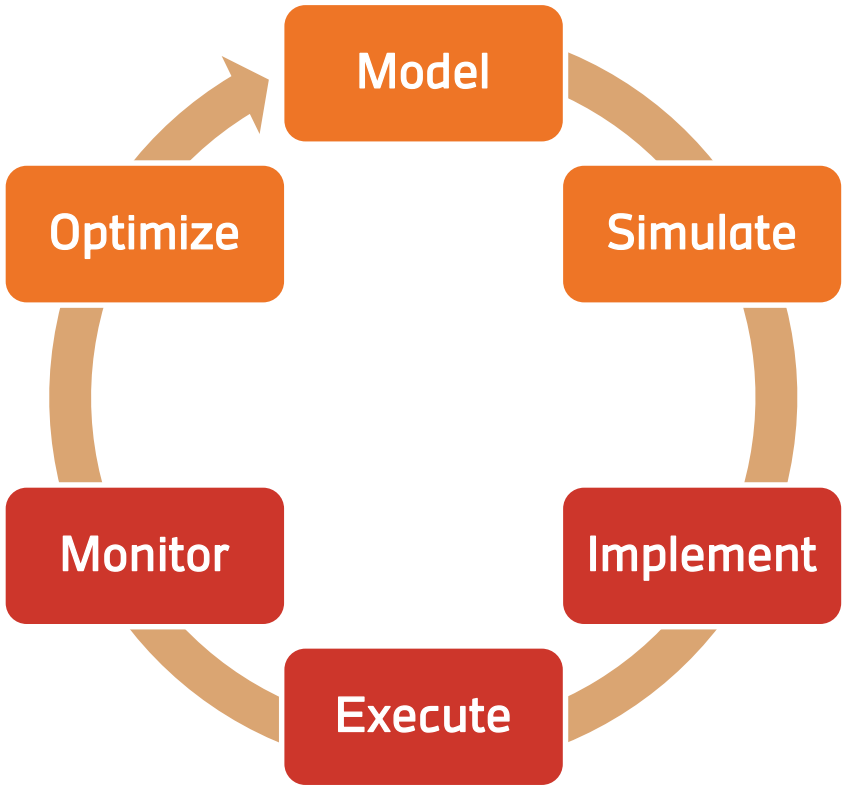
- ✓ Understand the area of Workflow Management
- ✓ Understand its relation to other BPM areas
- ✓ Understand key tasks in Workflow Management
- ✓ Model and execute workflows using Web Services and BPEL

Workflow Management in the BPM cycle

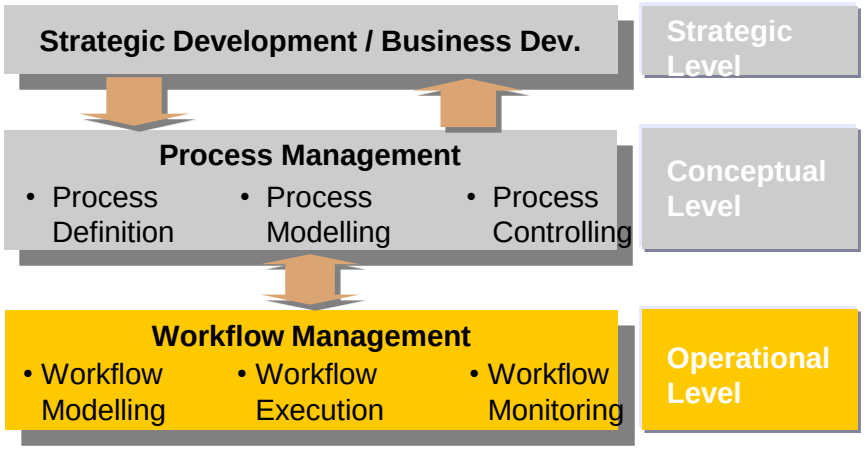
Implement
- Resolve all open questions, incl. technical

Execute
- Earn the merits of previous efforts

Monitor
- Gather data to feedback on performance



Workflow Management key tasks (Recap)



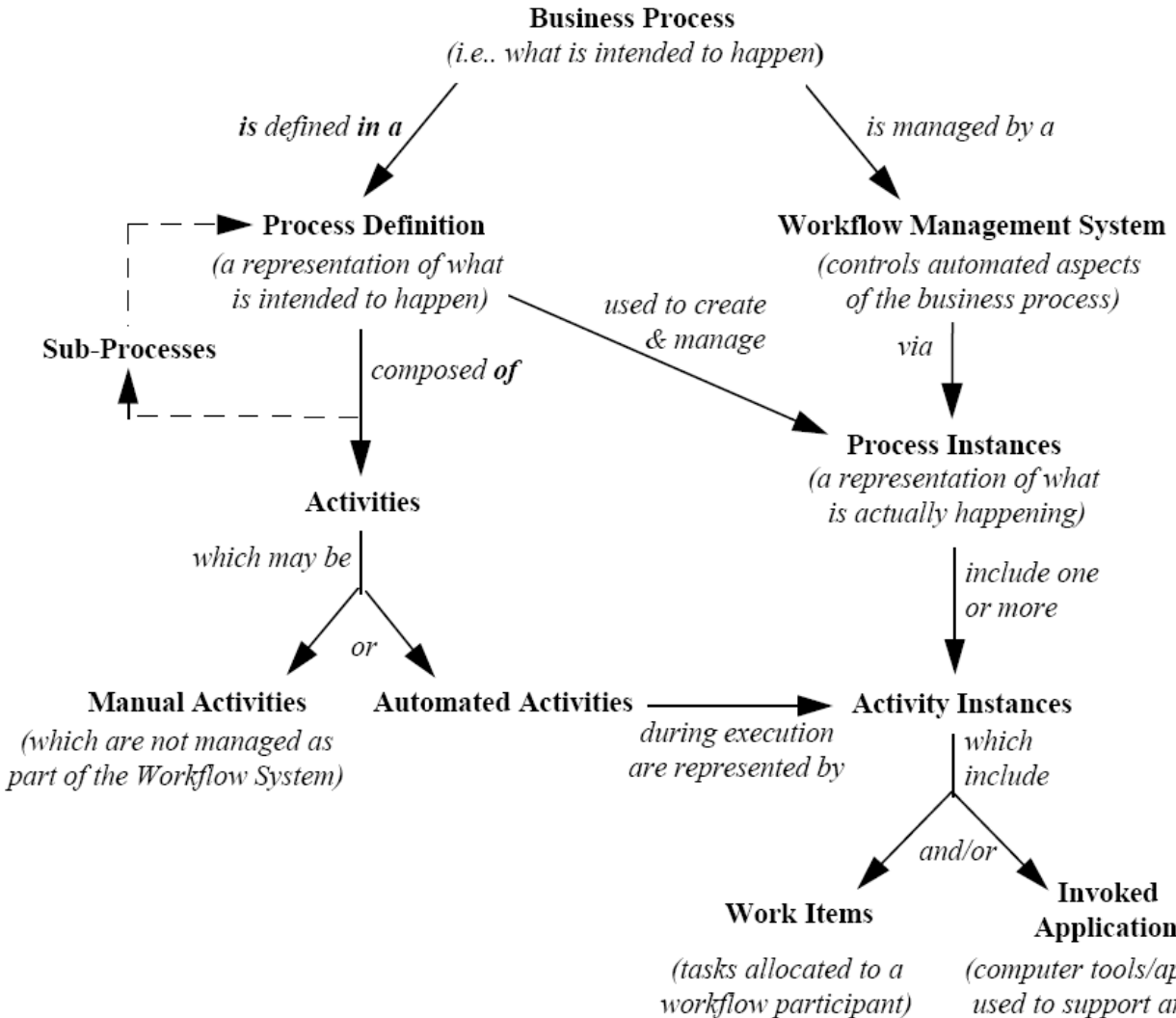
Key Tasks

- 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

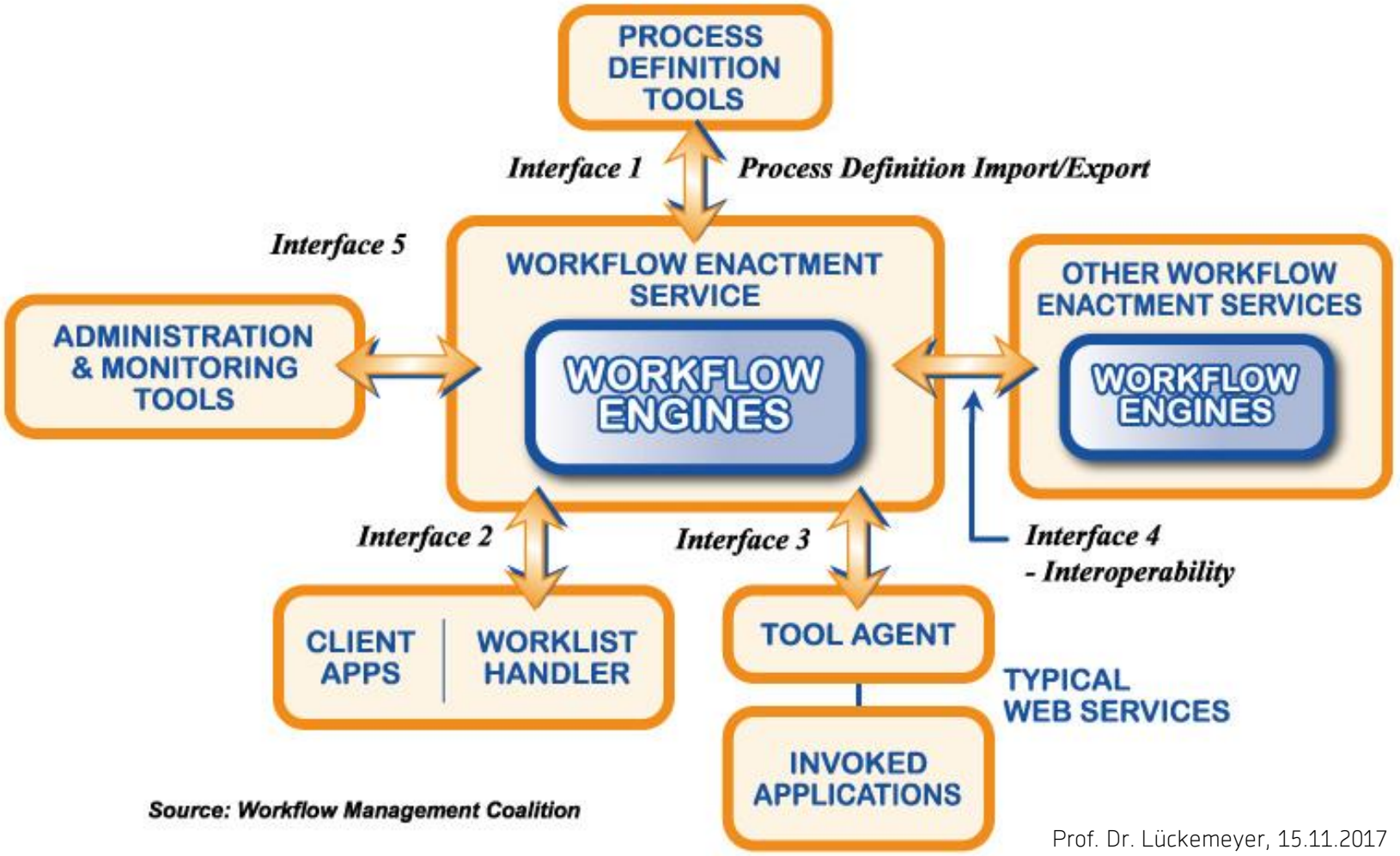
Terms and Definitions

- 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

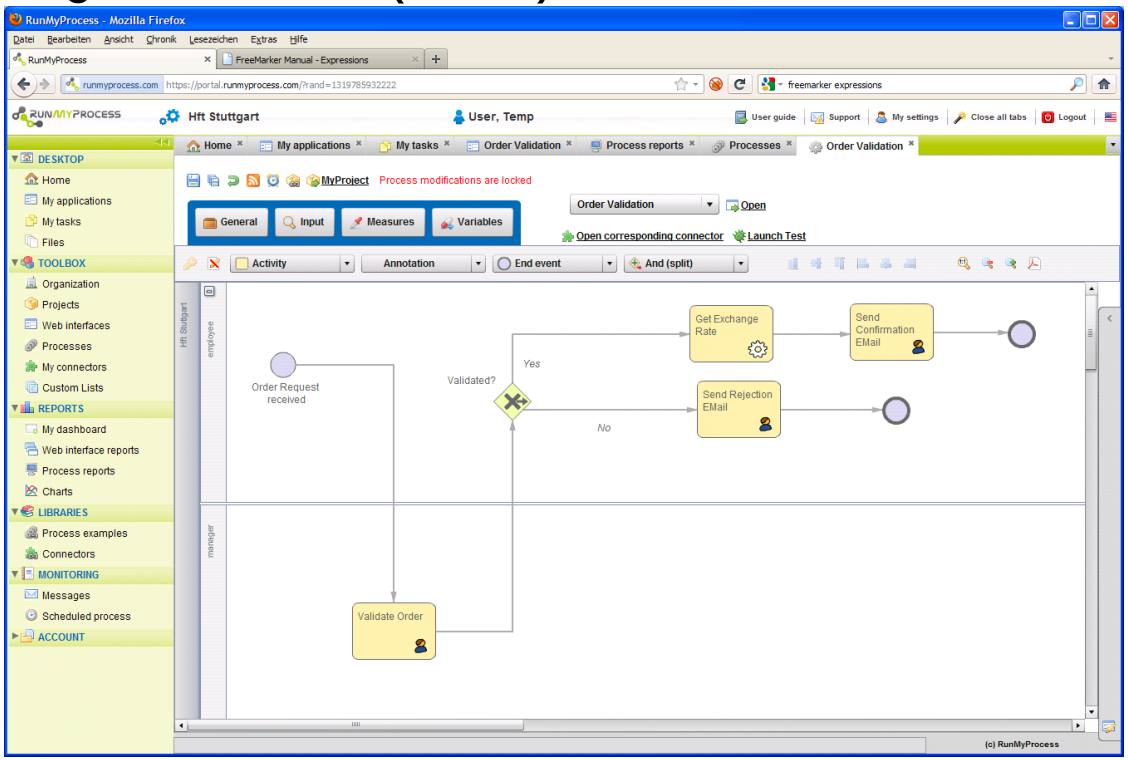


The WfMC Reference Model

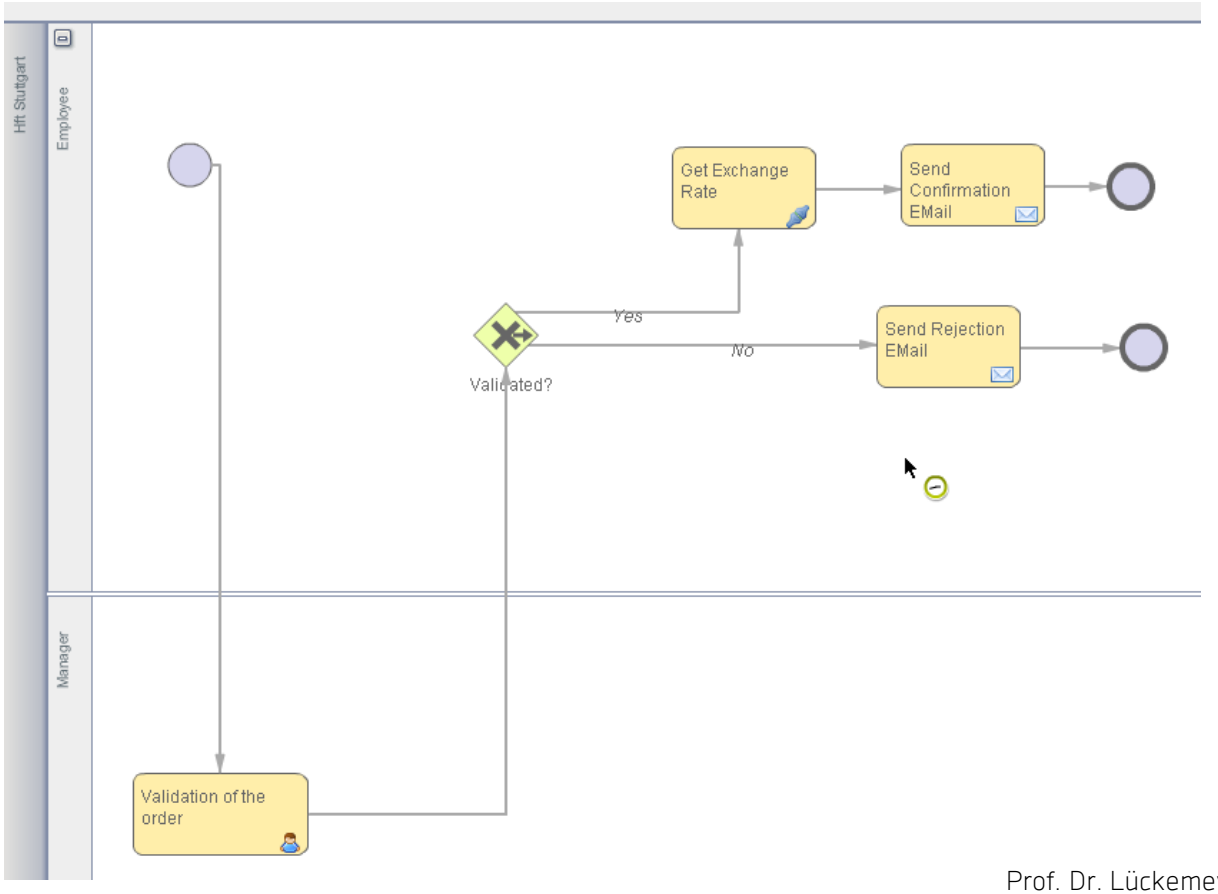


Fujitsu RunMyProcess

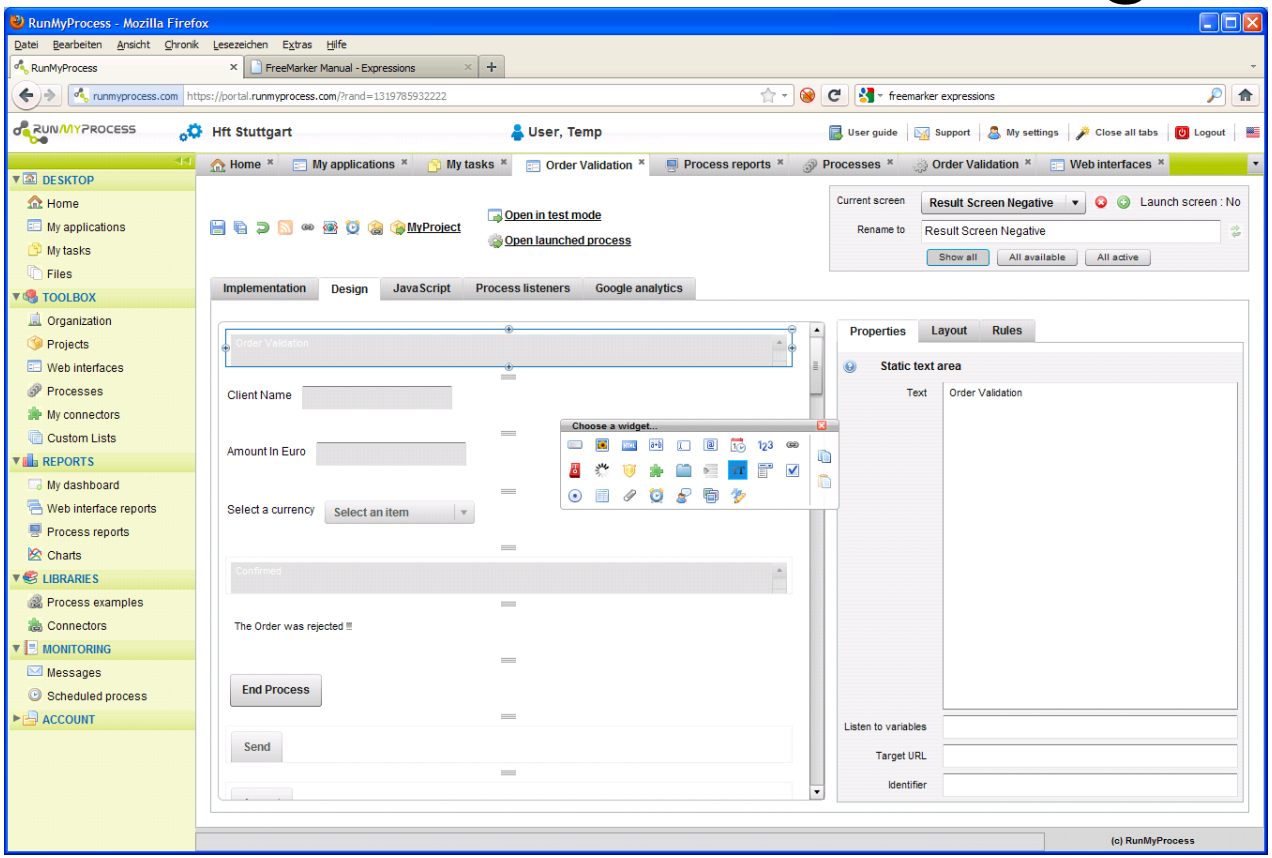
Web-based Workflow-Management-Tool (Flash)
Executable BPMN
Web-Forms to enter data in manual tasks
Support of Web-Service-Tasks and E-Mail-Tasks



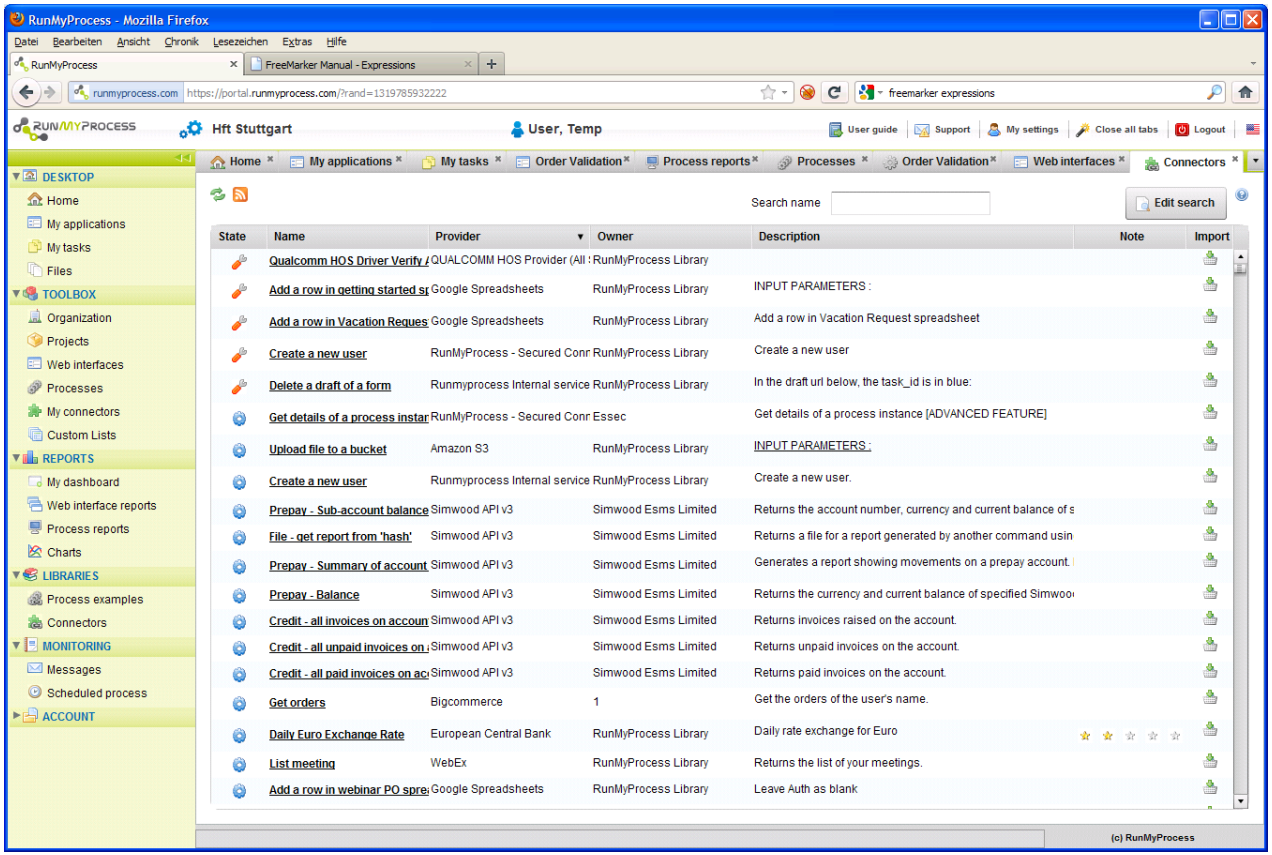
Fujitsu RunMyProcess example: order validation



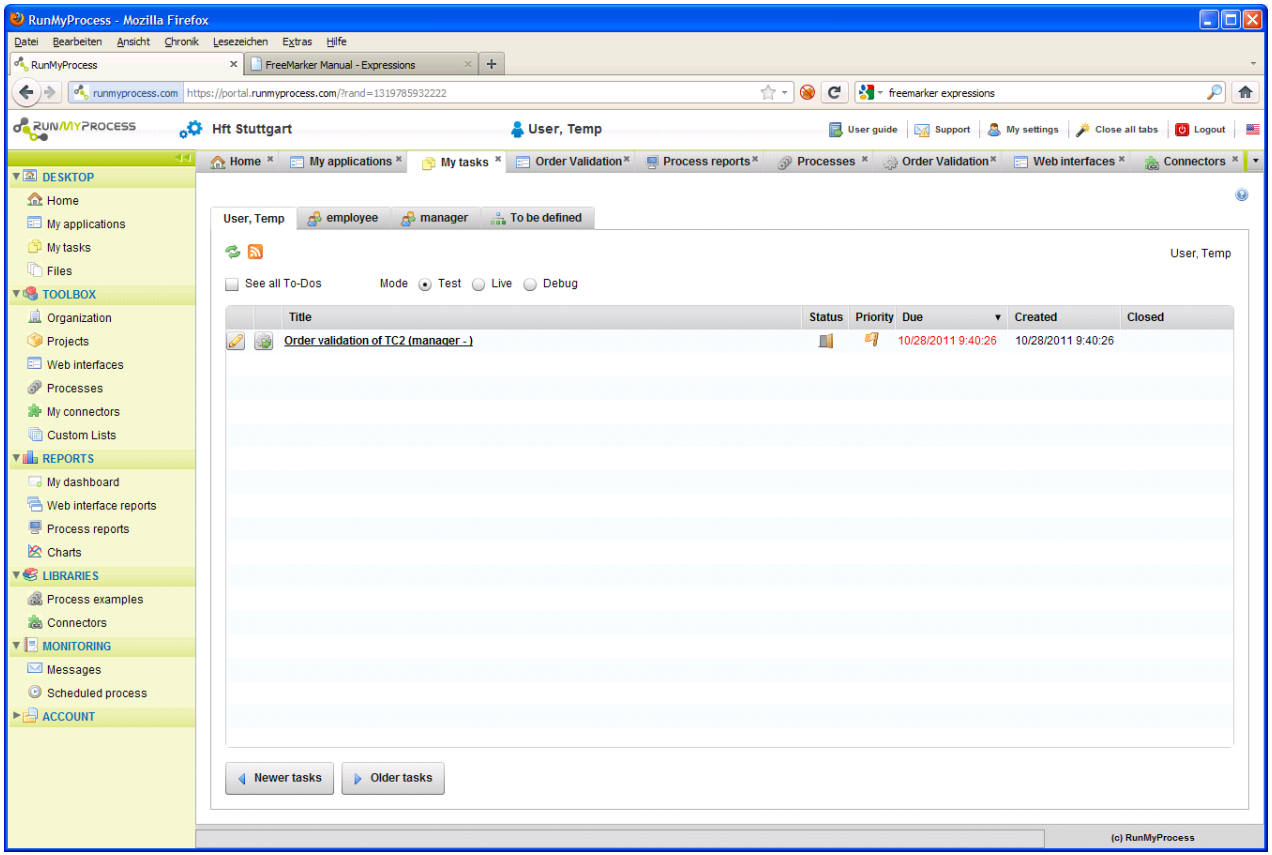
Fujitsu RunMyProcess features: Forms / Web-Interface Designer



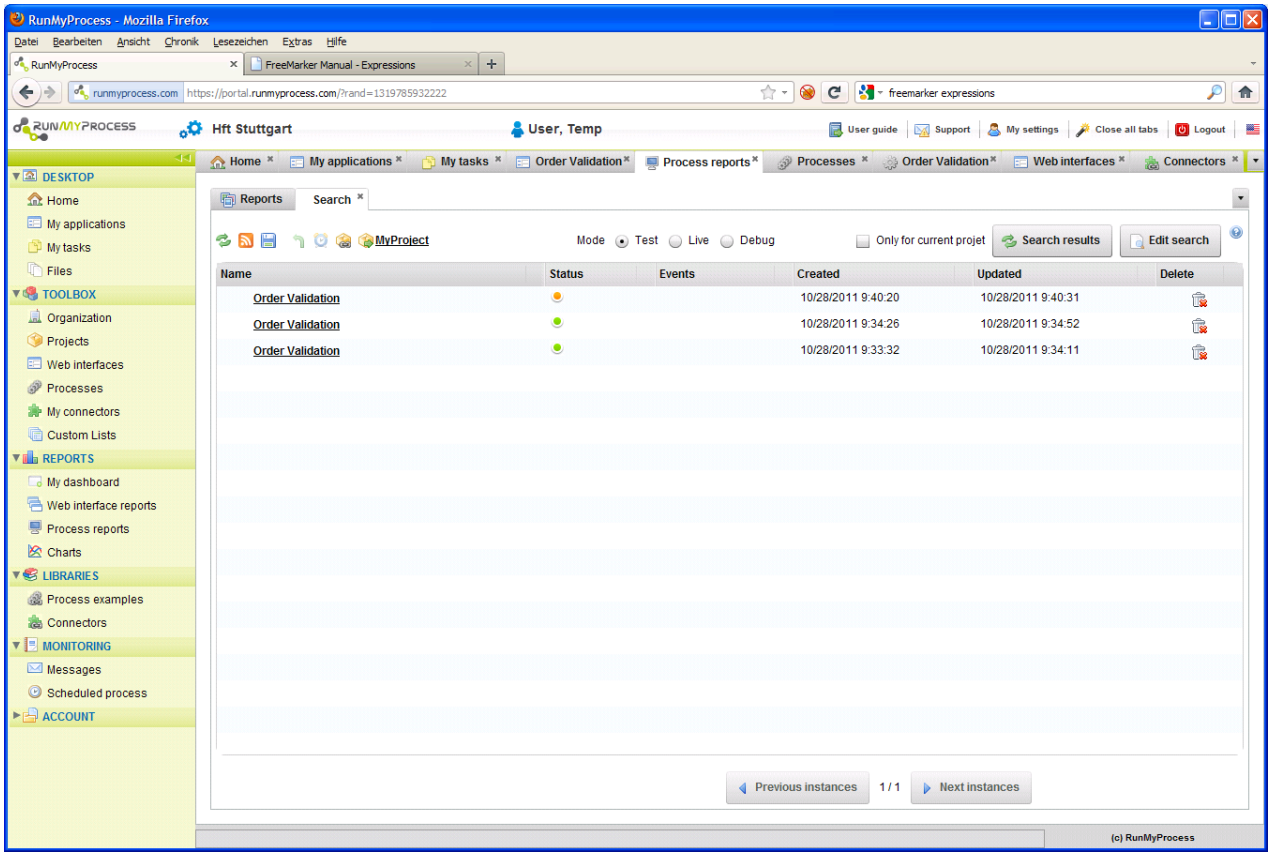
Fujitsu RunMyProcess features: Predefined Connectors & Services



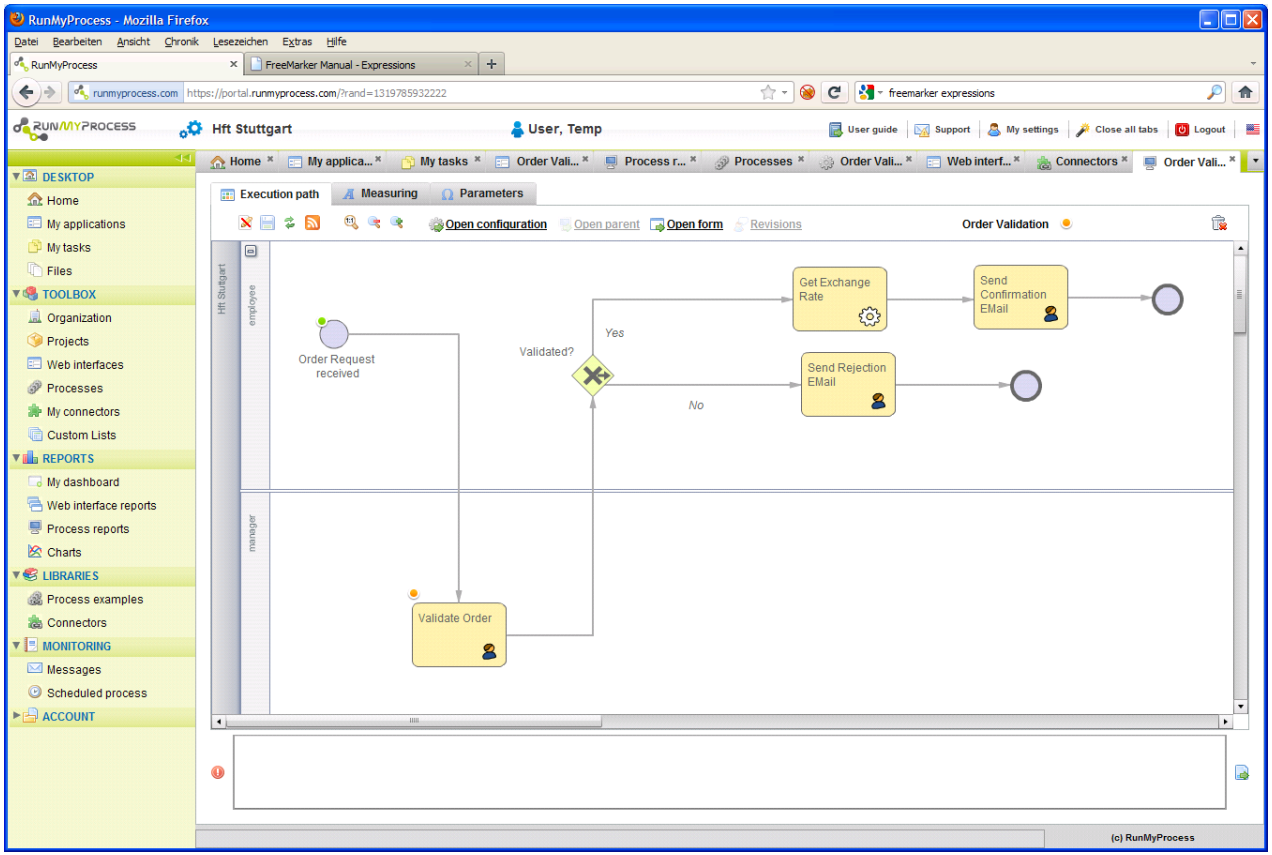
Fujitsu RunMyProcess execution: Tasks/ToDos



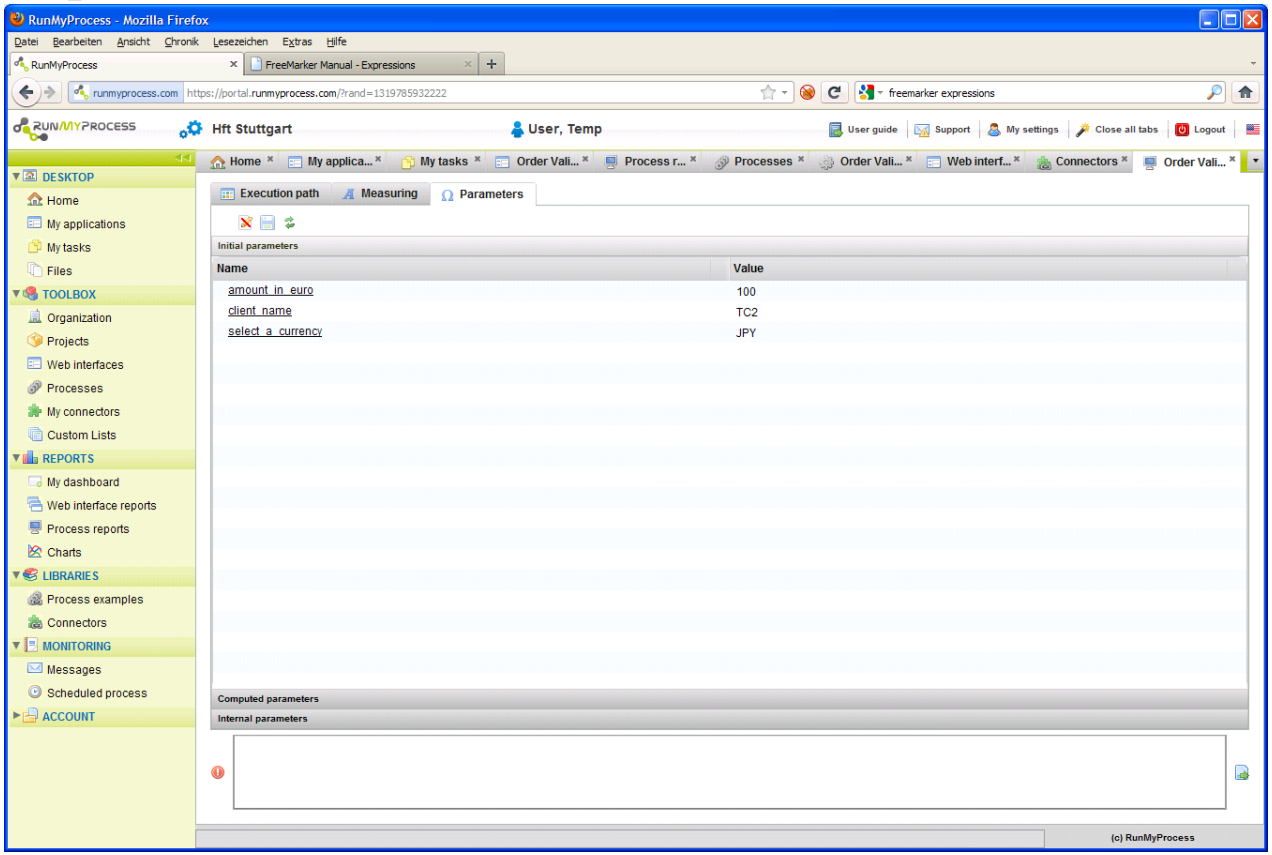
Fujitsu RunMyProcess monitoring: BAM overview



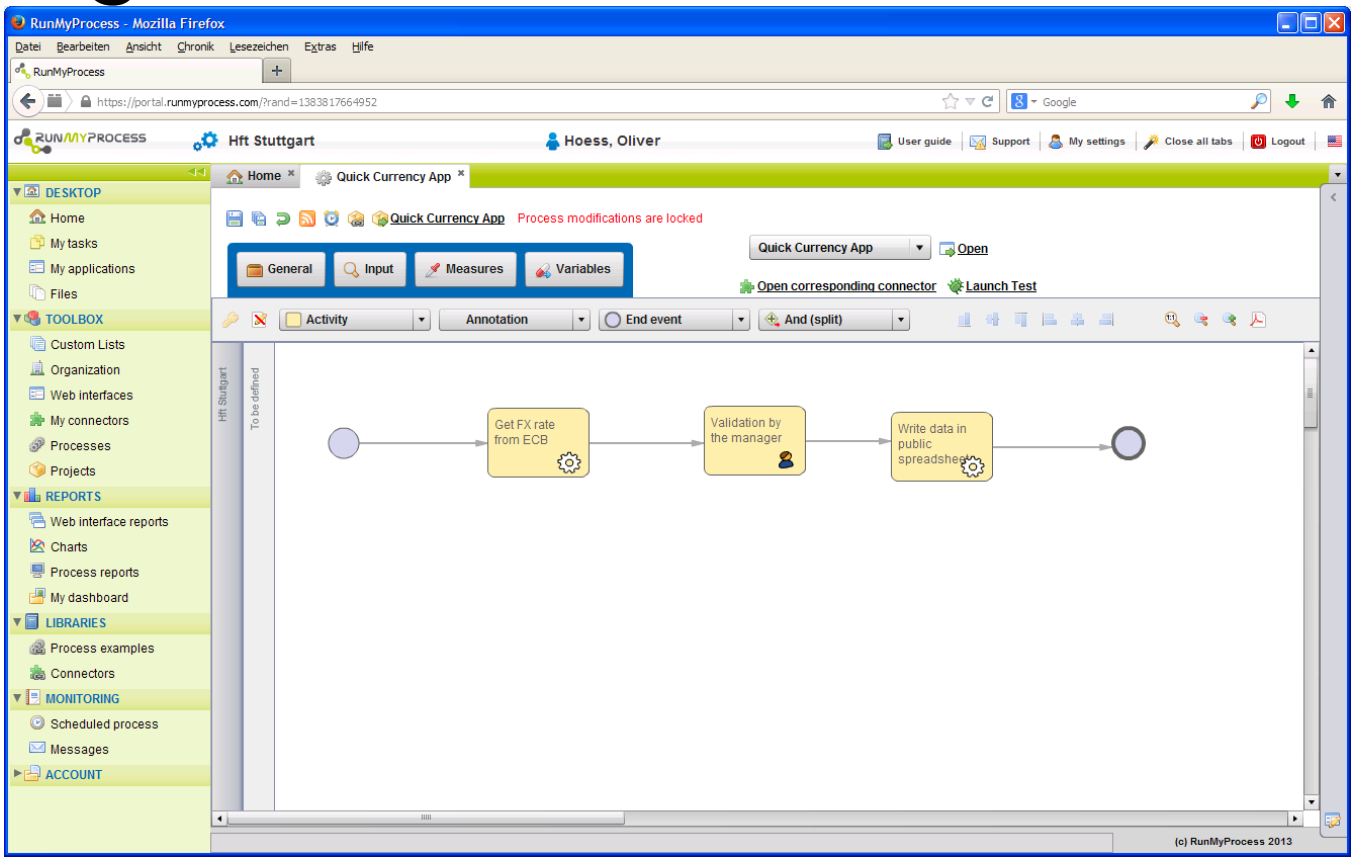
Fujitsu RunMyProcess monitoring: BAM detail view



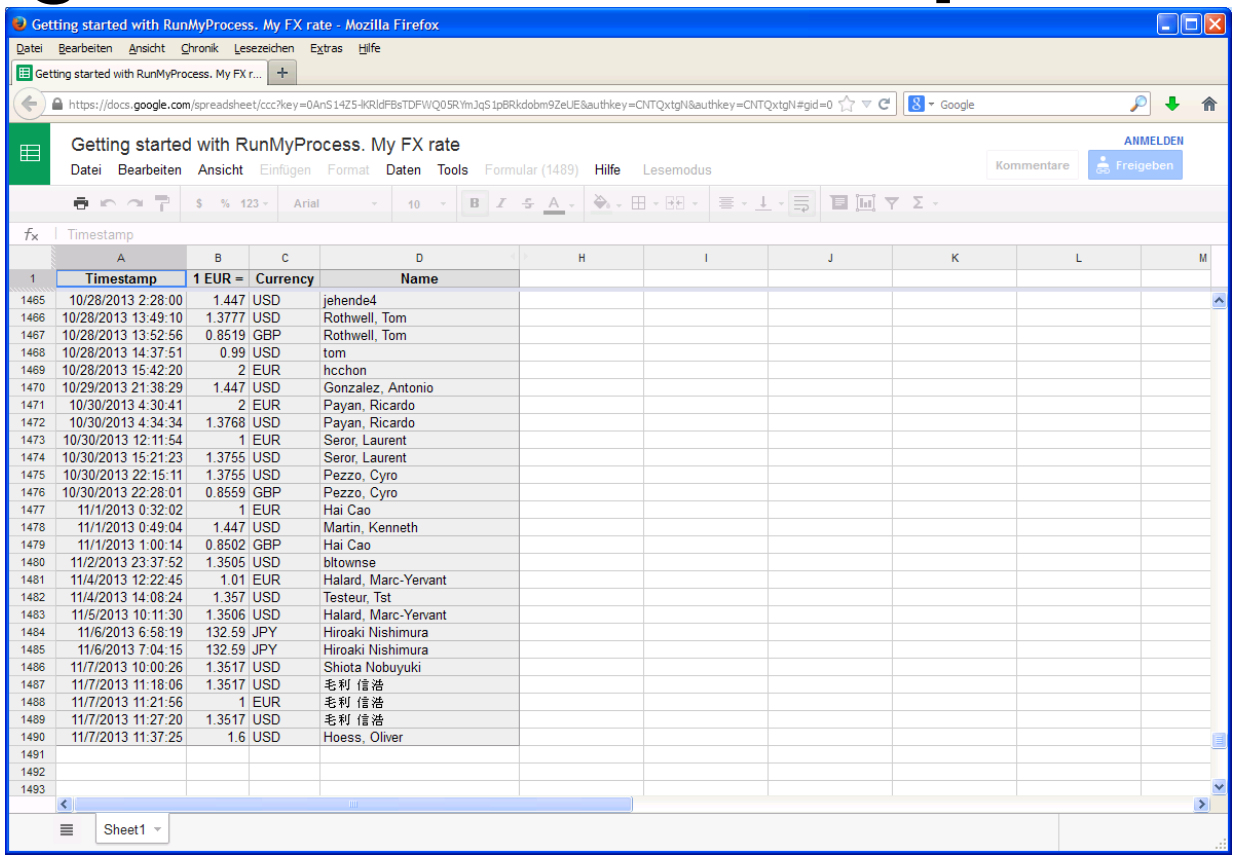
Fujitsu RunMyProcess monitoring: BAM quantitative view



Fujitsu RunMyProcess example: getting started tutorial



Fujitsu RunMyProcess example: getting started tutorial Spreadsheet



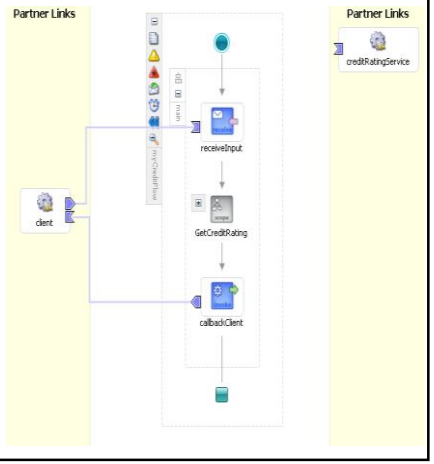
	A	B	C	D
1	Timestamp	1 EUR =	Currency	Name
1465	10/28/2013 2:28:00	1.447	USD	jehende4
1466	10/28/2013 13:49:10	1.3777	USD	Rothwell, Tom
1467	10/28/2013 13:52:56	0.8519	GBP	Rothwell, Tom
1468	10/28/2013 14:37:51	0.99	USD	tom
1469	10/28/2013 15:42:20	2	EUR	hcchon
1470	10/29/2013 21:38:29	1.447	USD	Gonzalez, 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	bltownse
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

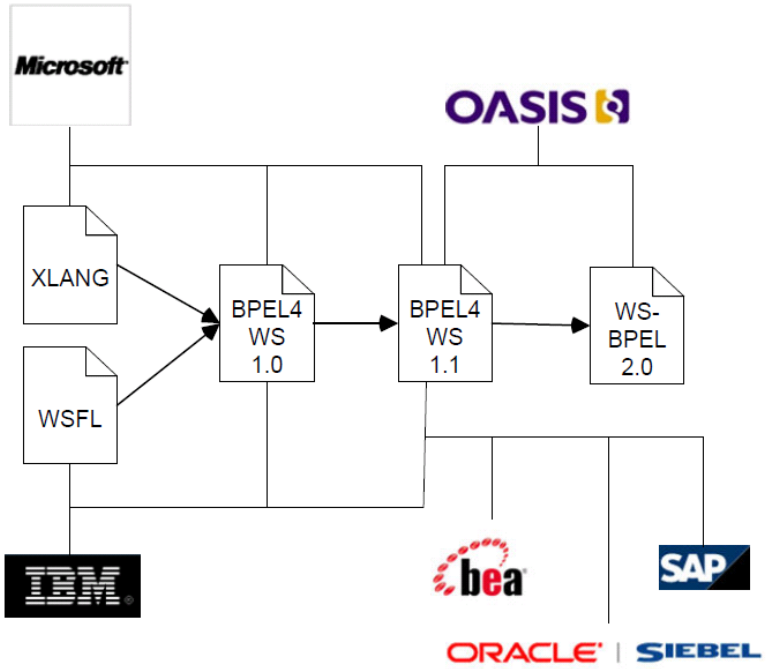
WSFL	XPDL	BPML	XLANG
Web-Services Flow Language (IBM)	XML Process Definition Language (WfMC)	Business Process Modeling Language (BPMI)	(Microsoft)

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

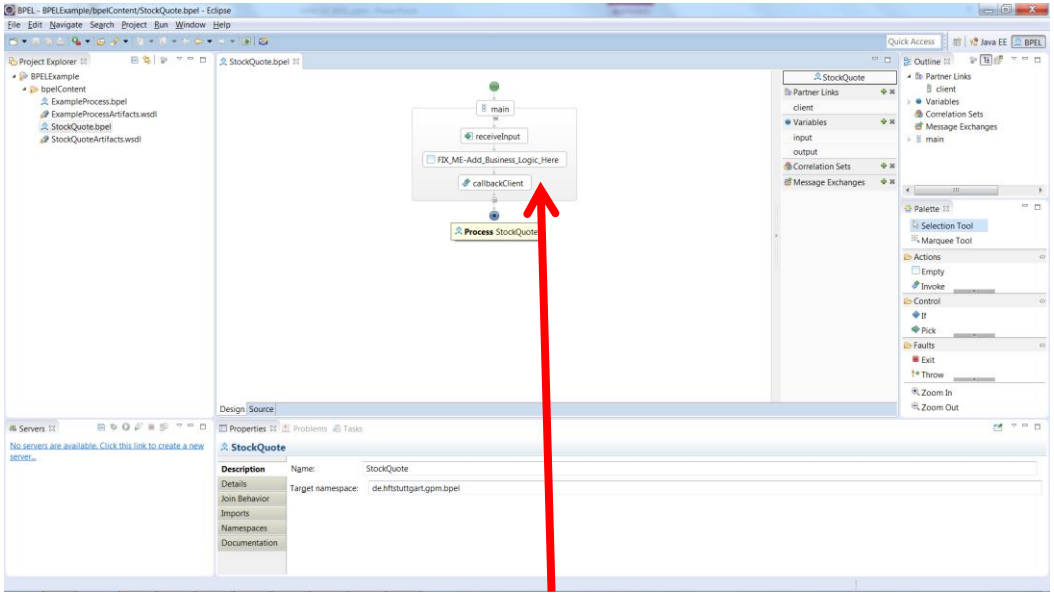
<process name=„Stock Quote“>

– …

</process>

Consists of

– Variables
– Partnerlinks
– Further elements

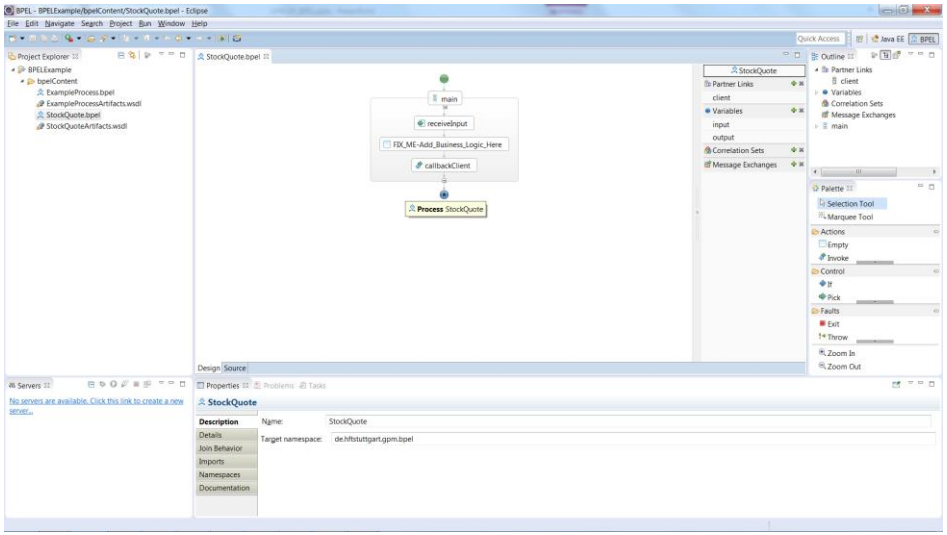


Process

BPEL Elements: Containers

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

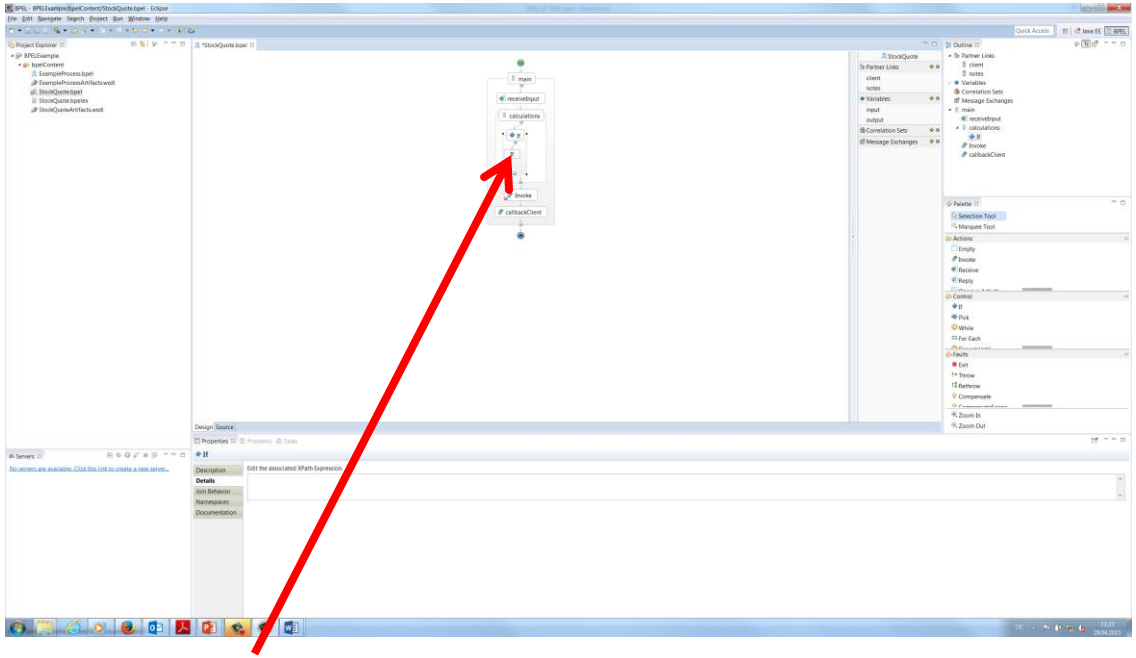
<flow/>
- Parallel execution of all subnodes
- <link/> for synchronization of the execution



BPEL Elements: Control Structures - if

<if>: Testing a condition

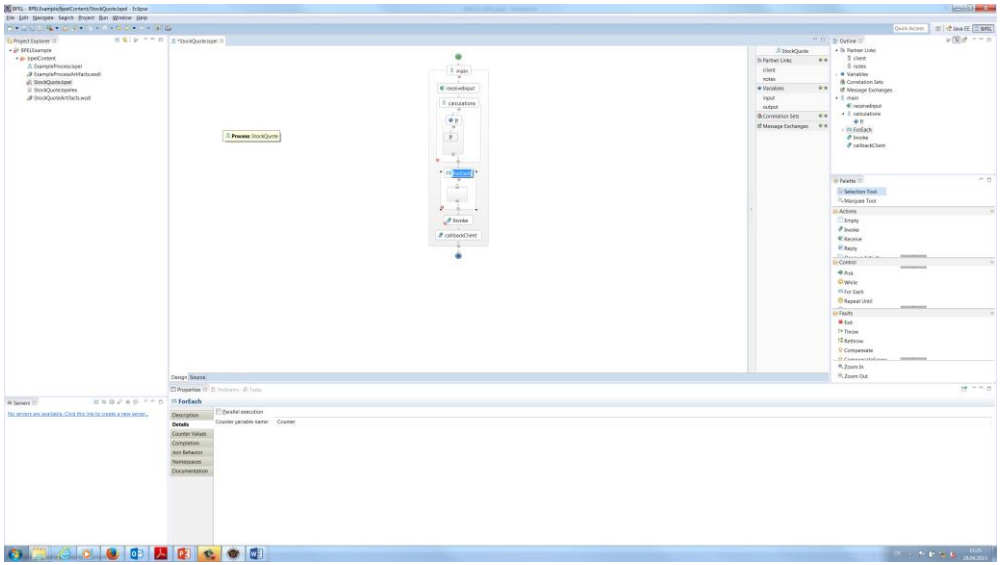
- If-then-else semantics
- Condition in Eclipse BPEL specified as an Xpath expression



<if>

BPEL Elements: Control Structures (2) - loops

Derived from classical programming languages
<while>
<RepeatUntil>
<ForEach>



BPEL Elements: <partnerlink>

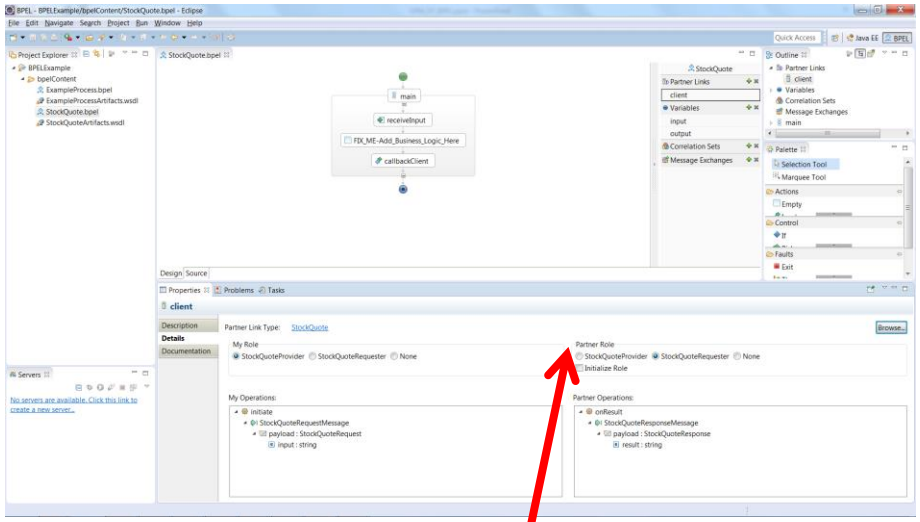
Basic Idea

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

<partnerlink>

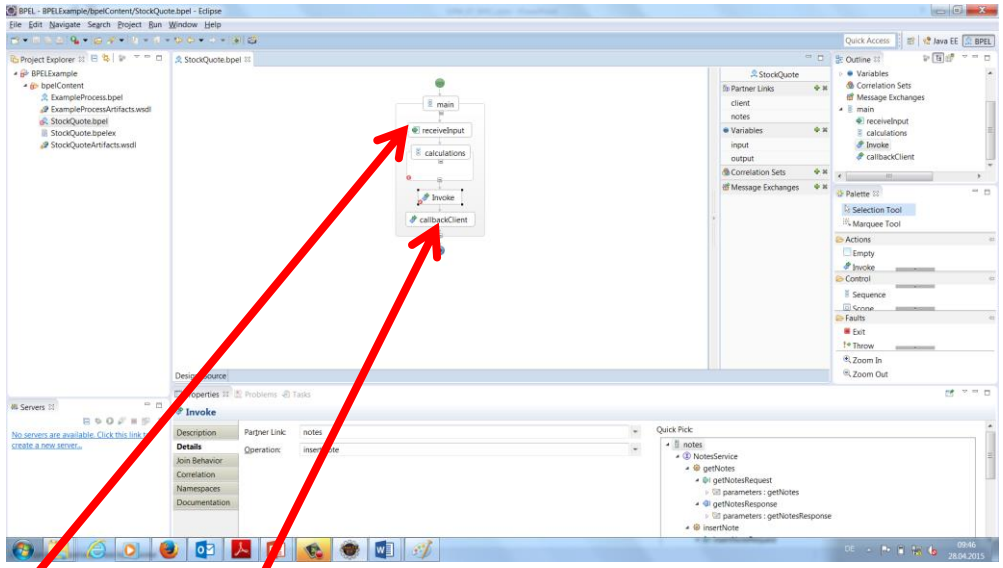
- Denotes a web service which is used or provided by a BPEL process

Description of the interface via WSDL



BPEL Elements: Communication

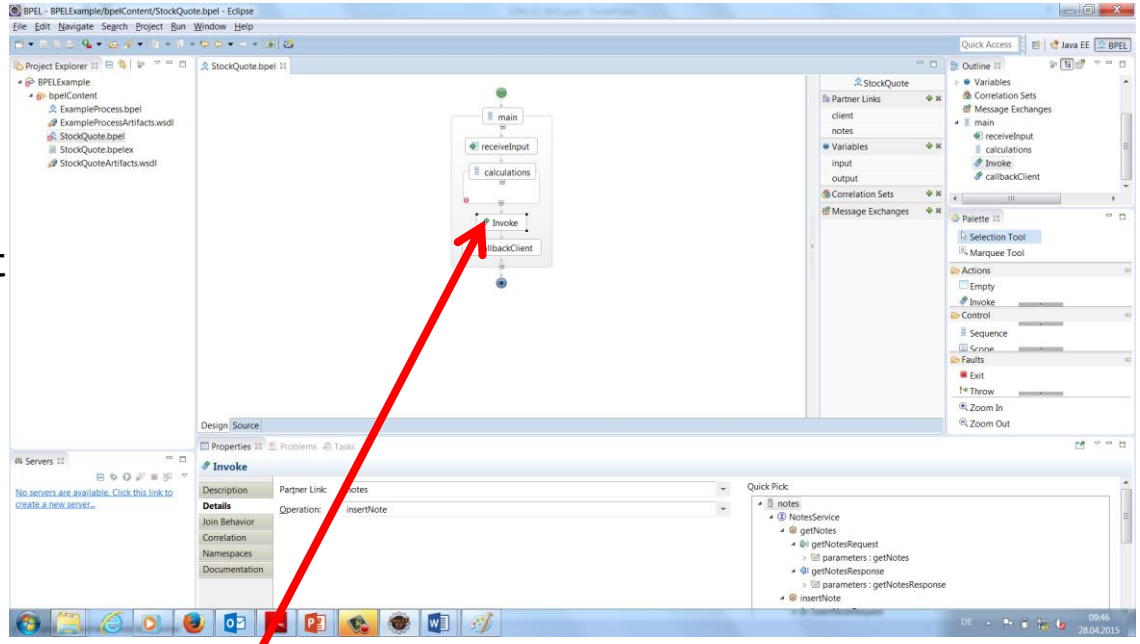
<receive/>: Receive request from the calling party
<reply/>: Answer request



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



<invoke>

BPEL Elements: Assignments

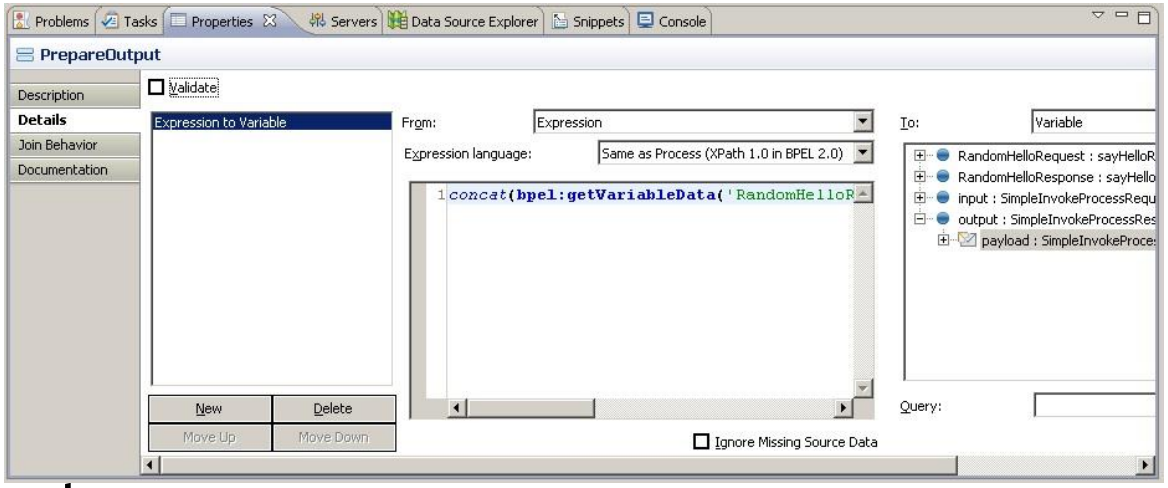
<assign/>: Node for assignments

<copy/>

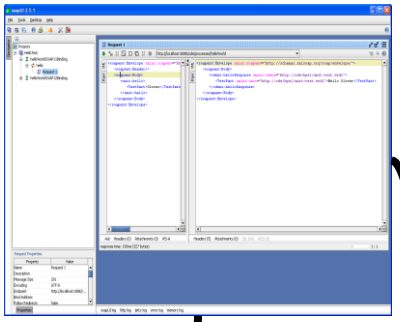
- Copy operation
- One or more <copy> operations per <assign/>

<from/><to/>: from … to

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

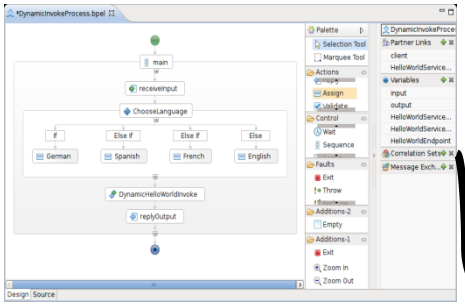


BPEL: Example technology stack



soapUI: calling Web Services

Testing



Eclipse: Creating/Running BPEL-Models

Apache Tomcat

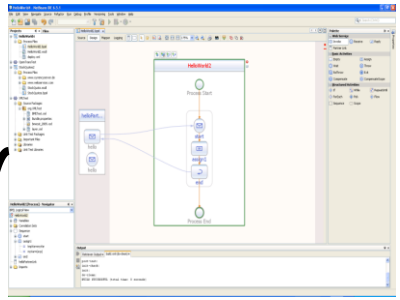
Apache ODE BPEL-Engine

Deployment

Glassfish ESB

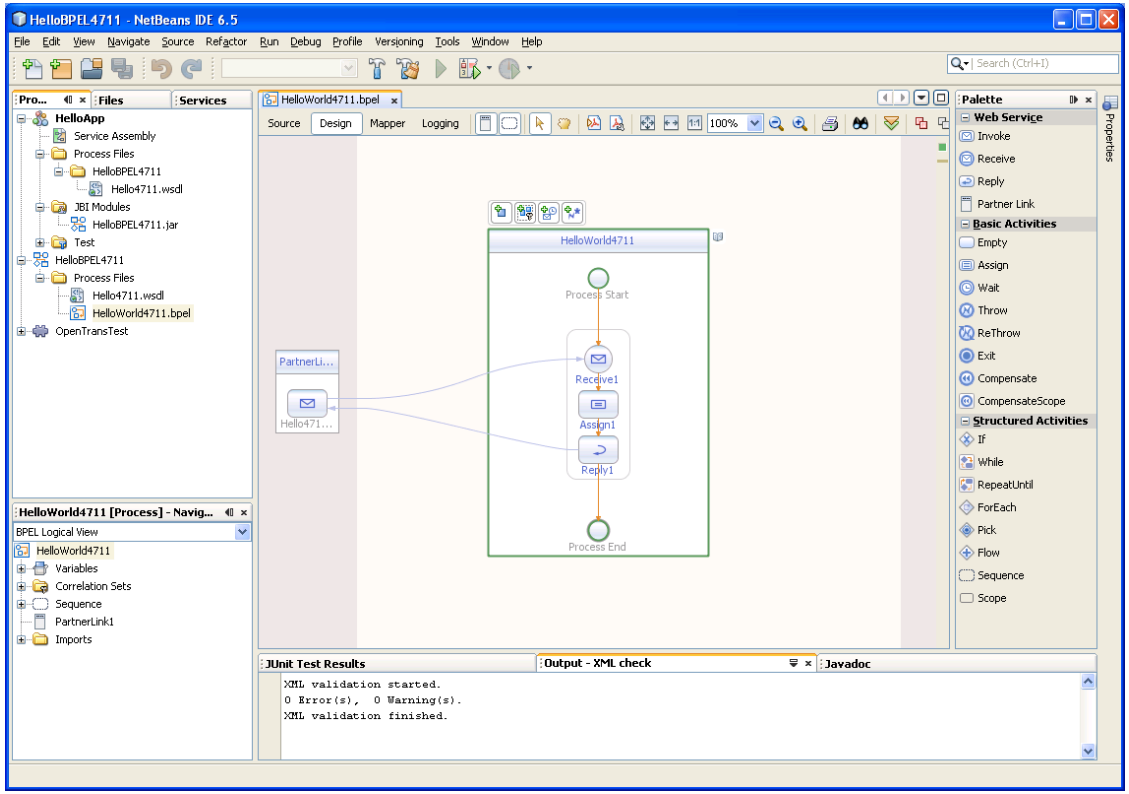
Sun BPEL-Engine

Deployment

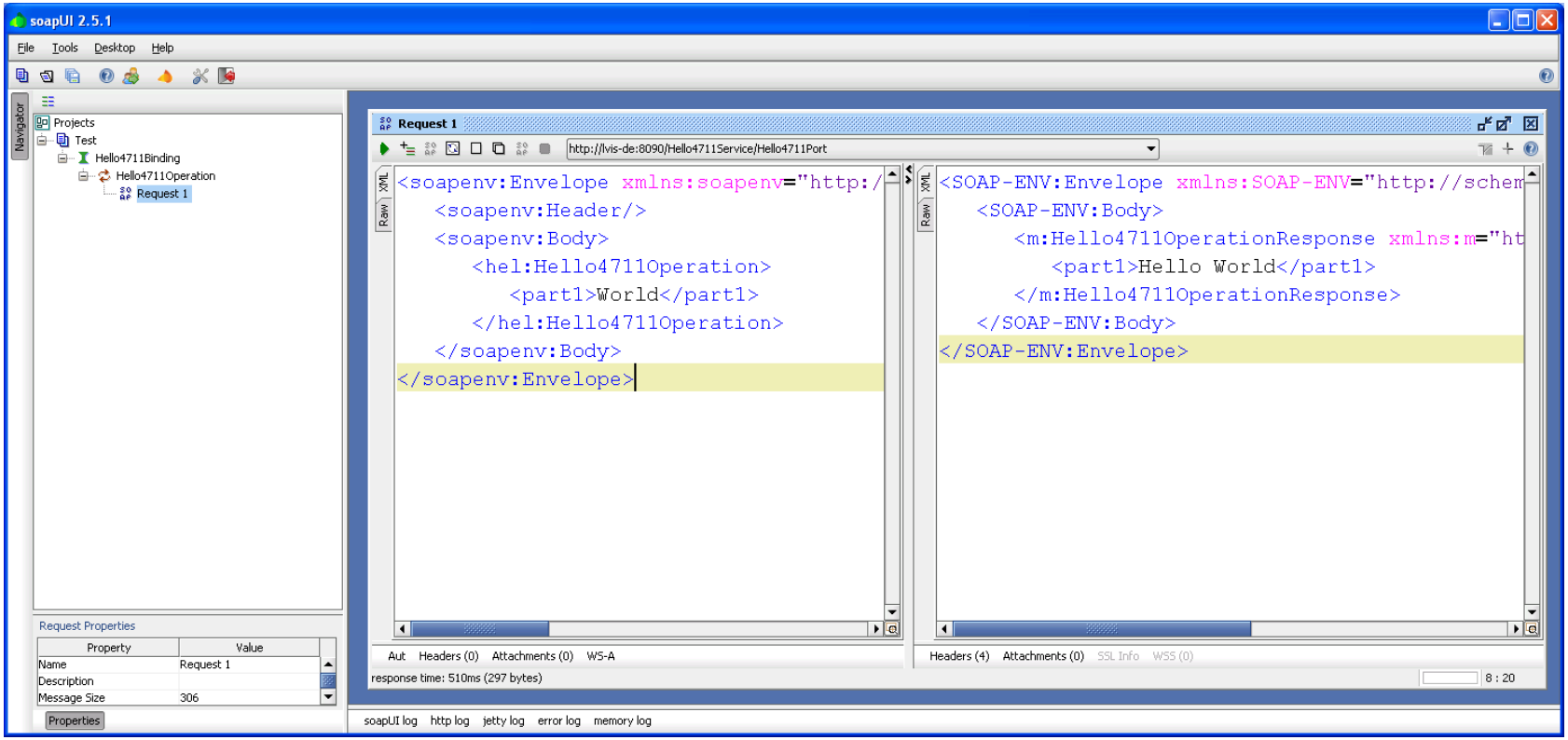


Netbeans: Creating/Running BPEL-Models

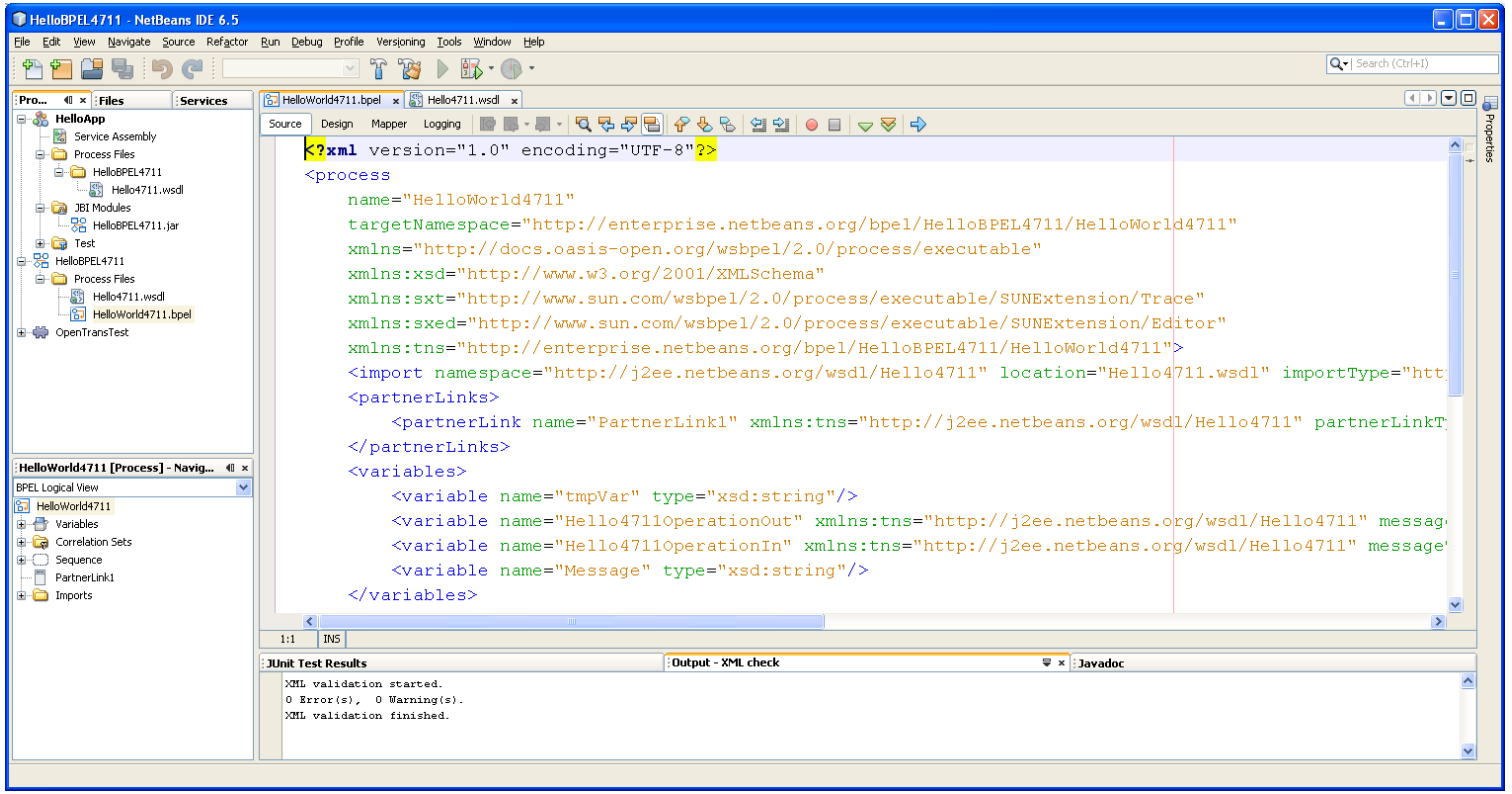
BPEL: Simple process example



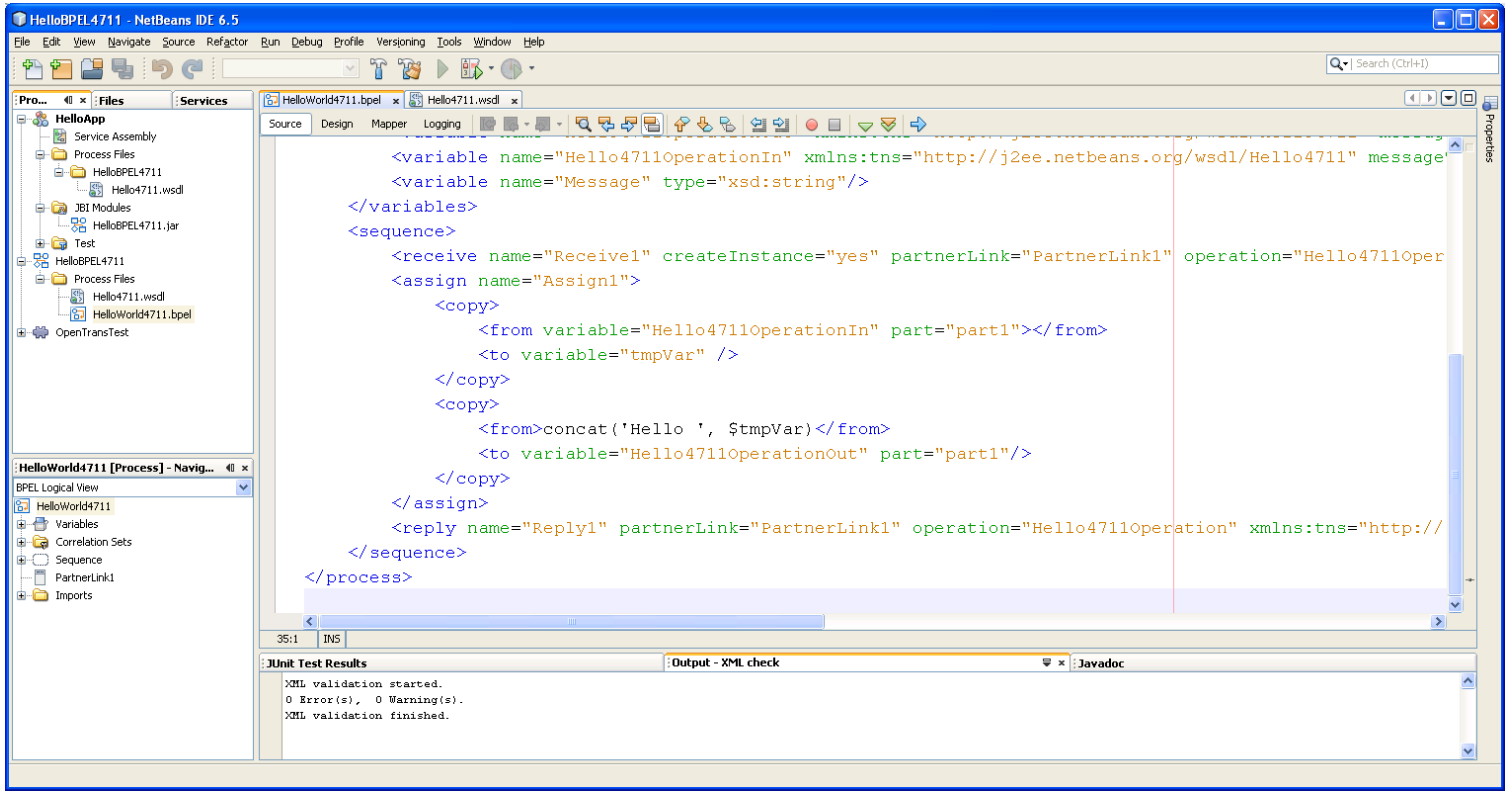
BPEL: example test using soapUI



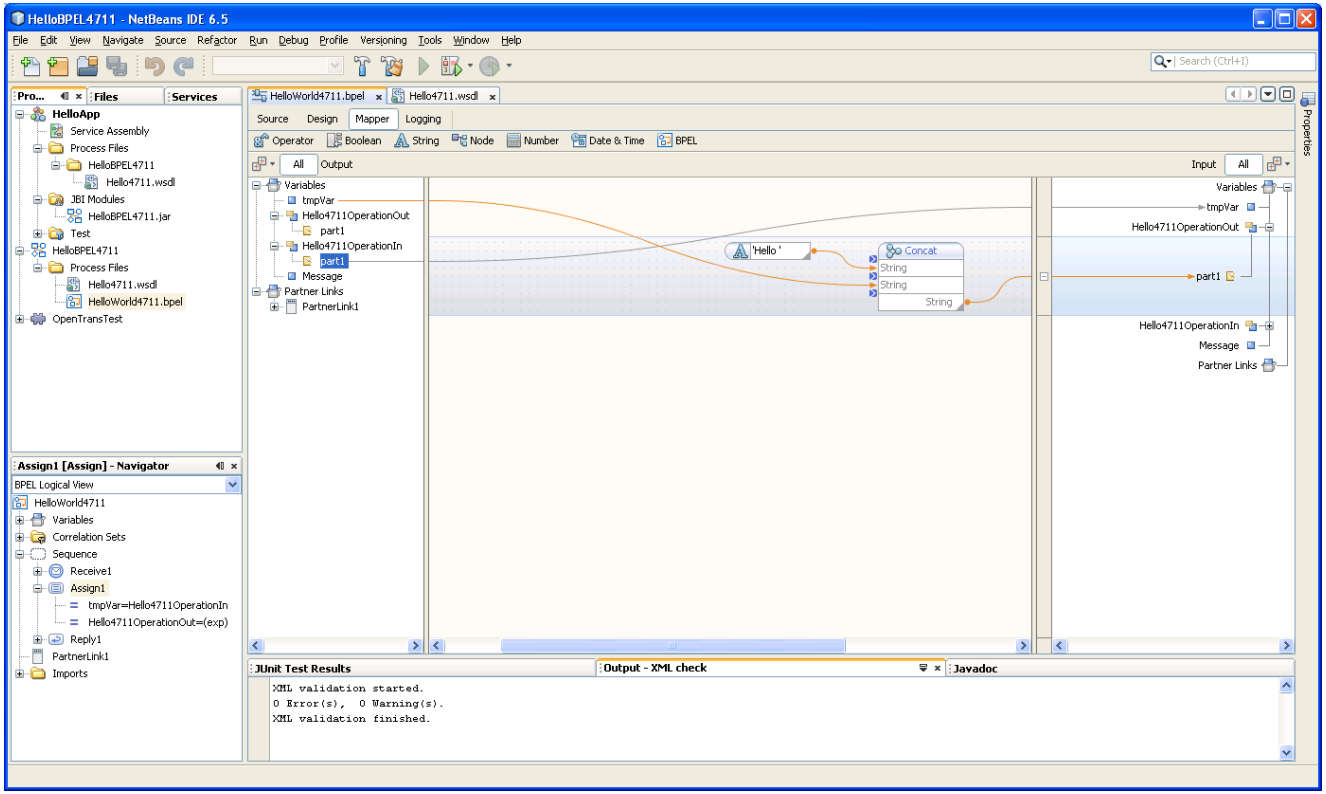
BPEL example: XML source code (1)



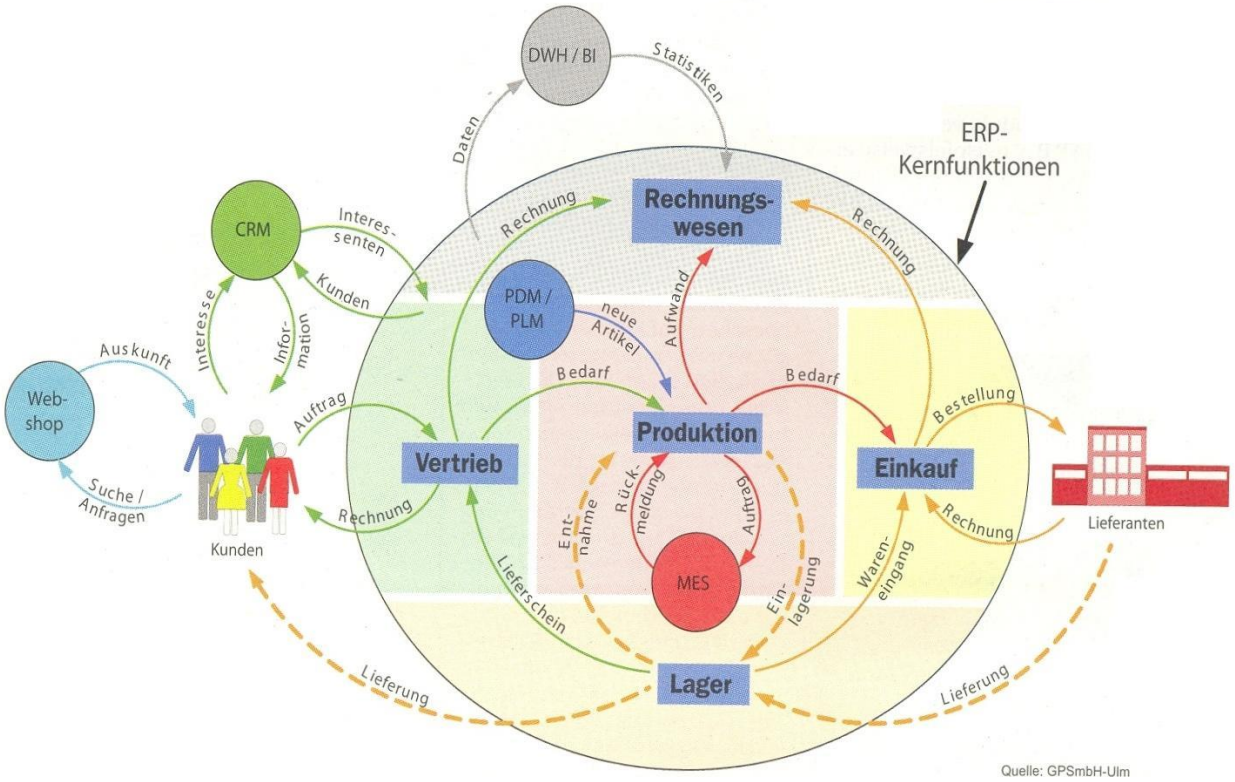
BPEL example: XML source code (2)



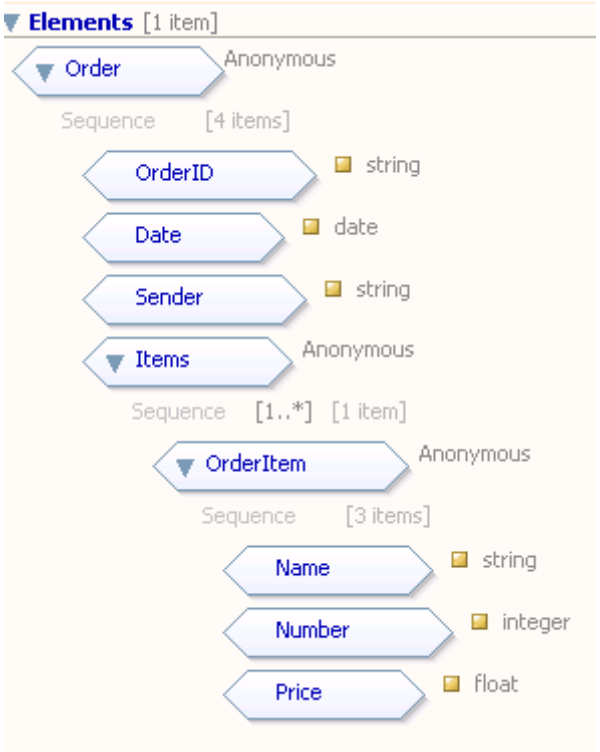
BPEL example: Netbeans' graphical mapper



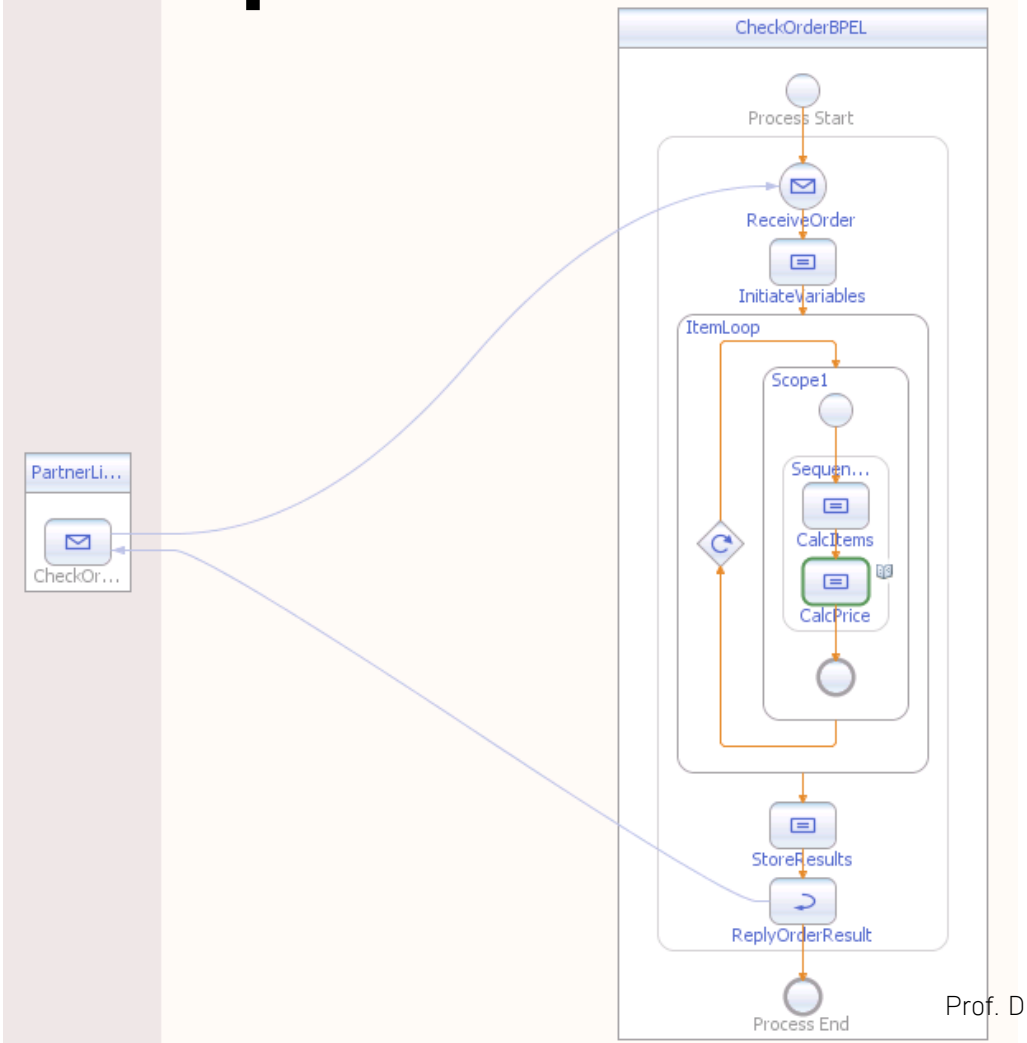
Aggregated services: reference company recap



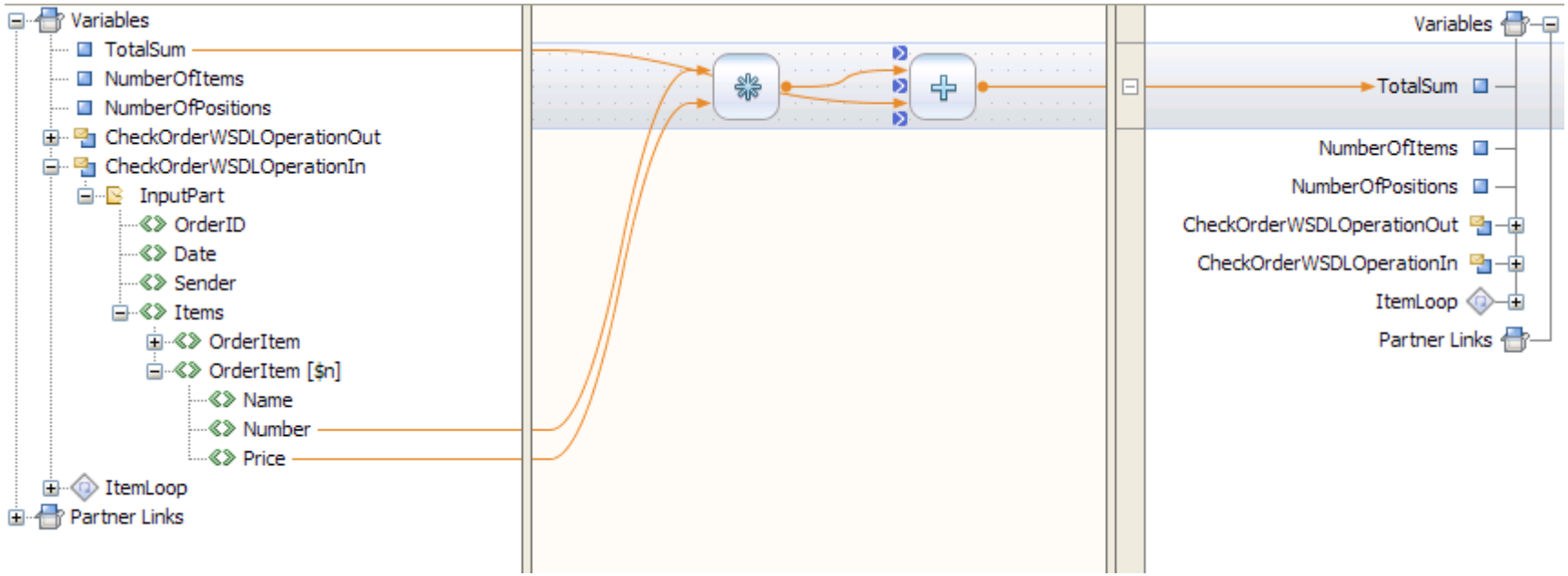
Aggregated services: order summation recap



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>
```

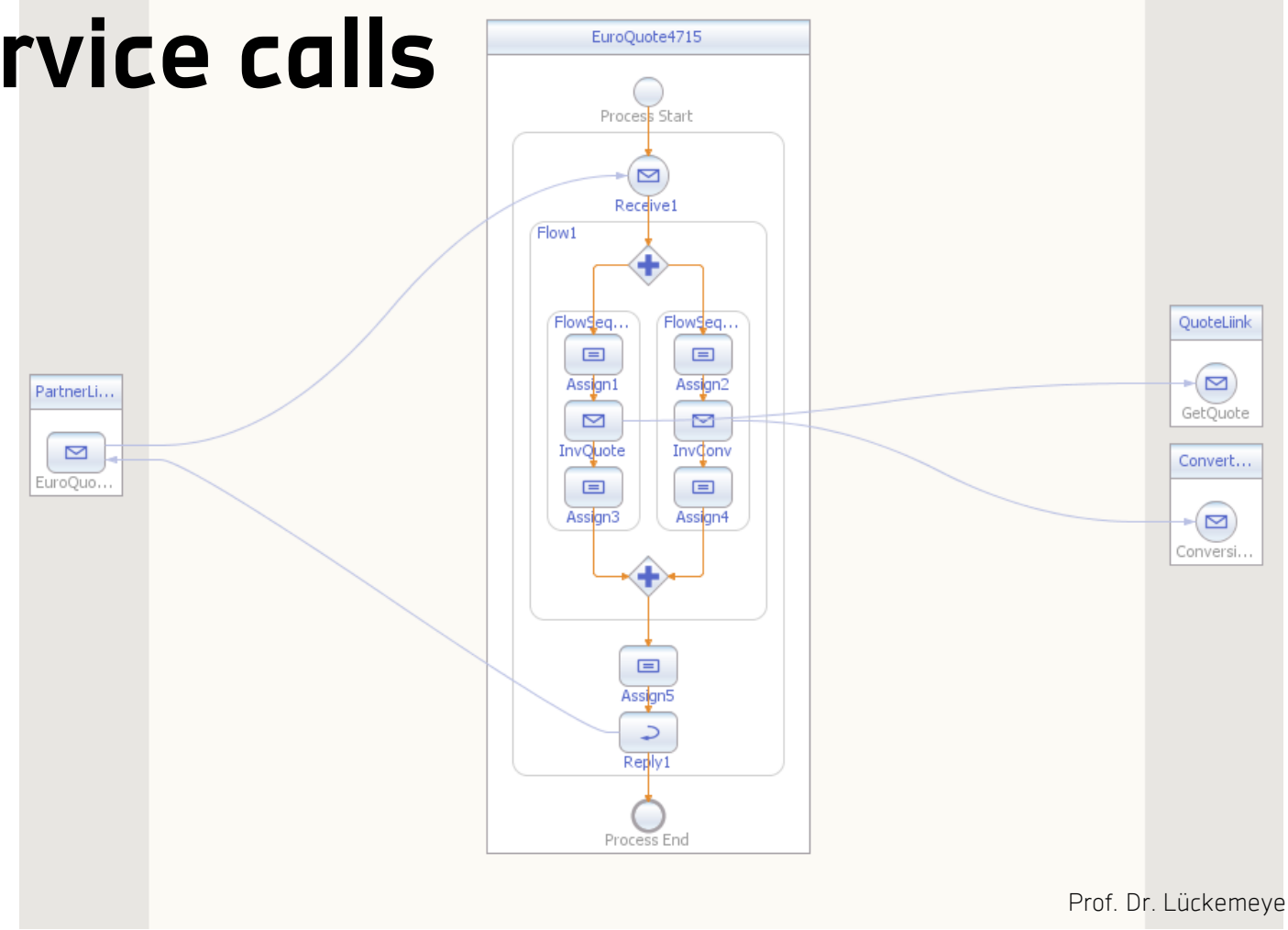
BPEL example: order summation – SOAP messages

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsc
   <soapenv:Header/>
   <soapenv:Body>
      <ord:Order>
         <ord:OrderID>4711</ord:OrderID>
         <ord:Date>2010-05-16</ord:Date>
         <ord:Sender>HFT</ord:Sender>
         <ord:Items>
            <ord:OrderItem>
               <ord:Name>Item 1</ord:Name>
               <ord:Number>2</ord:Number>
               <ord:Price>20.0</ord:Price>
            </ord:OrderItem>
            <ord:OrderItem>
               <ord:Name>Item 2</ord:Name>
               <ord:Number>6</ord:Number>
               <ord:Price>100.0</ord:Price>
            </ord:OrderItem>
         </ord:Items>
      </ord:Order>
   </soapenv:Body>
</soapenv:Envelope>
```

XML | Raw

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas
   <SOAP-ENV:Body>
      <OrderResult xmlns:msgns="http://j2ee.netbe
         <OrderID>4711</OrderID>
         <NumberOfPositions>2</NumberOfPositions>
         <NumberOfItems>8</NumberOfItems>
         <TotalSum>640.0</TotalSum>
      </OrderResult>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

BPEL example (2): parallel web service calls

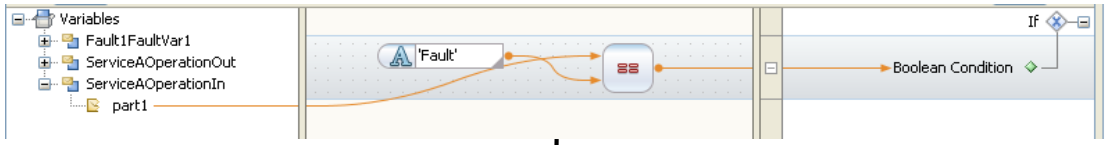


BPEL: fault handling

In distributed systems new error situations can occur
- Unavailability of a partner-service, e.g. due to system or network failure
- Wrong parameters

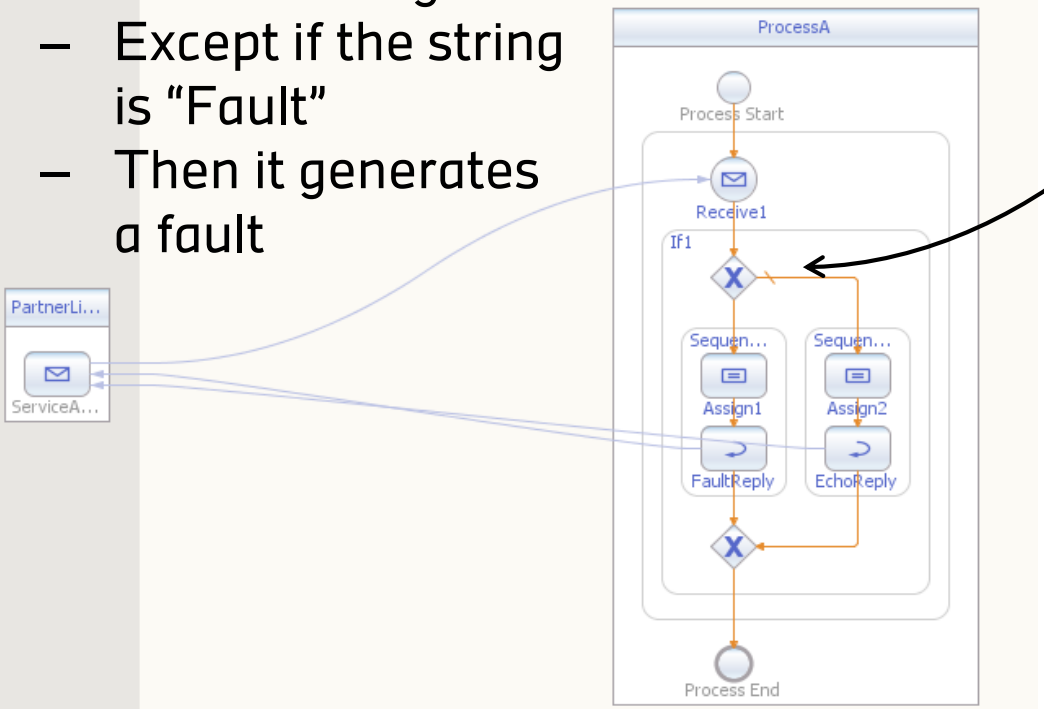
BPEL offers some constructs to deal with that fact:
- A fault type system
  - Pre-defined faults
  - User-defined faults
- Fault messages in addition to input/output parameters
- "Throw" activity to throw faults
- Fault handlers to catch faults
  - On a process level
  - On a scope level

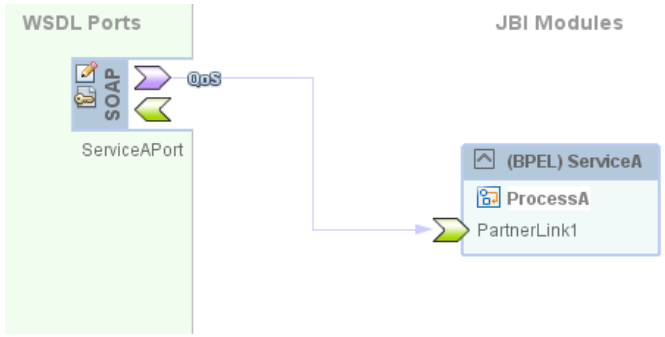
BPEL fault handling example: string echo



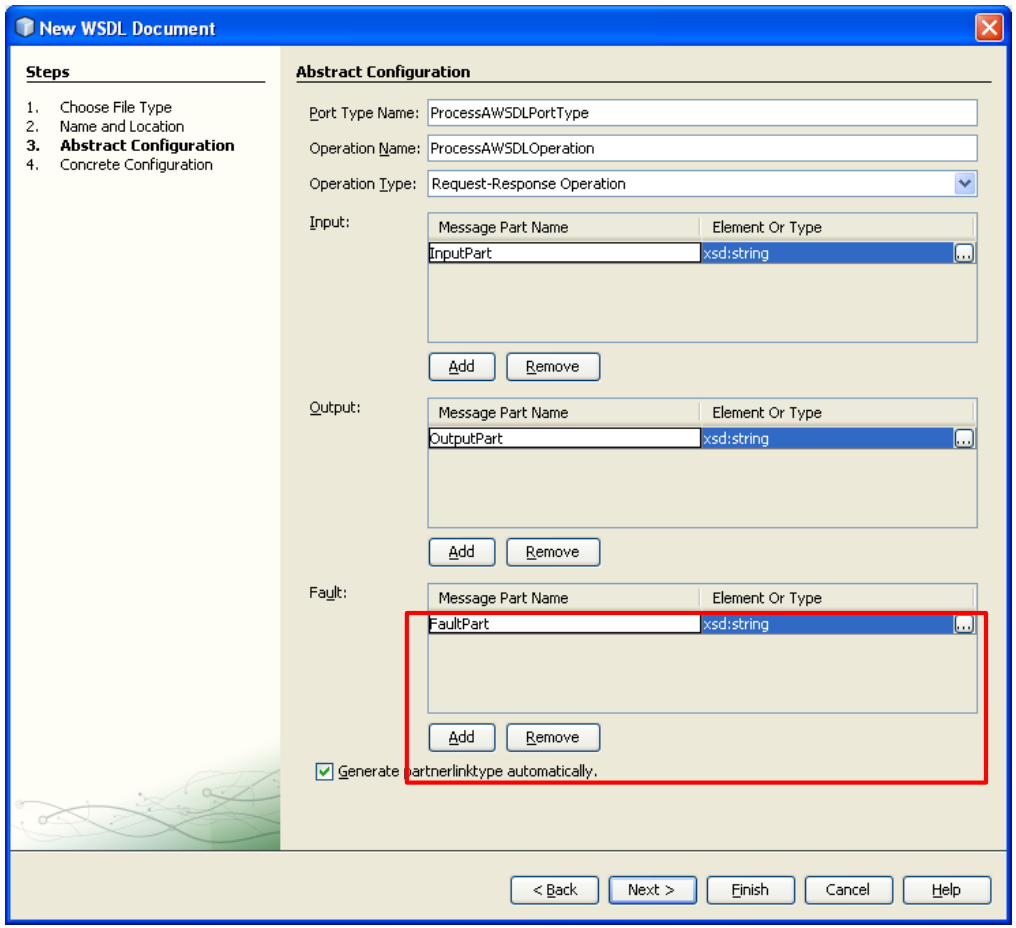
Process A

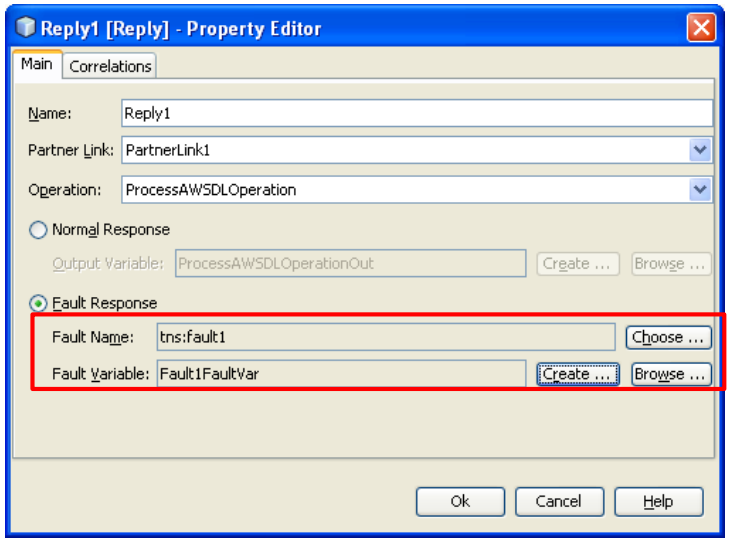
- Echoes string
- Except if the string is “Fault”
- Then it generates a fault





BPEL example: fault creation in the WSDL

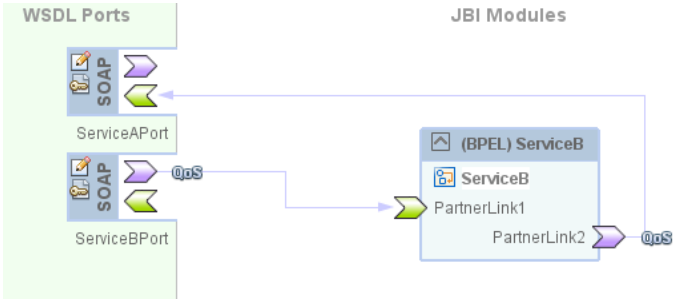


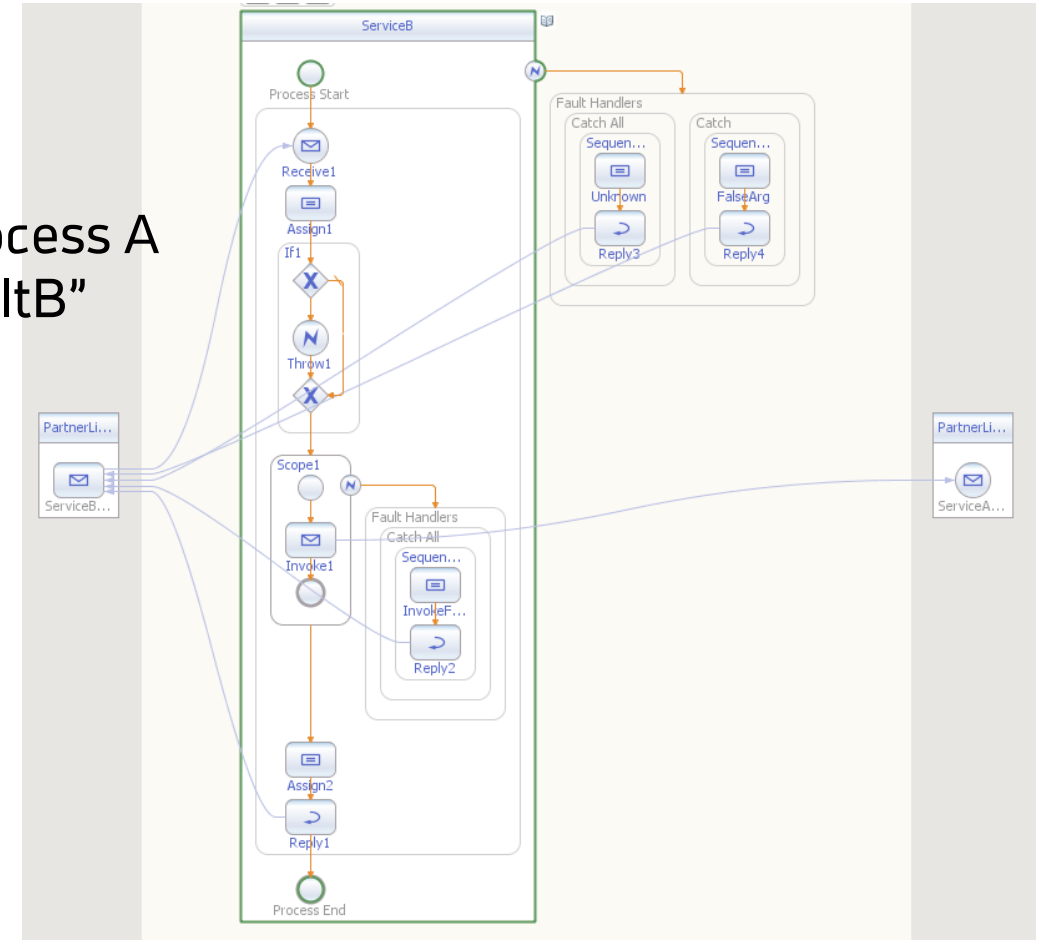


BPEL example: fault handling

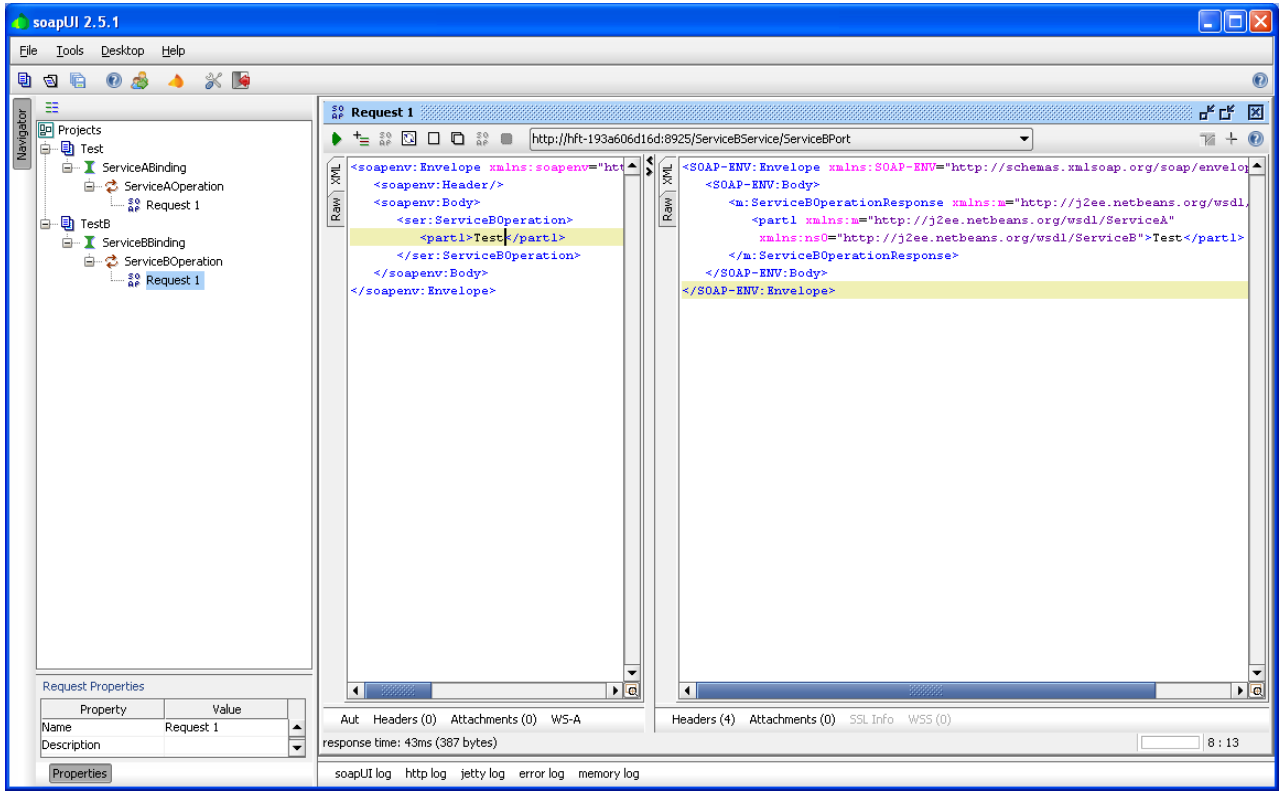
Process B

- 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

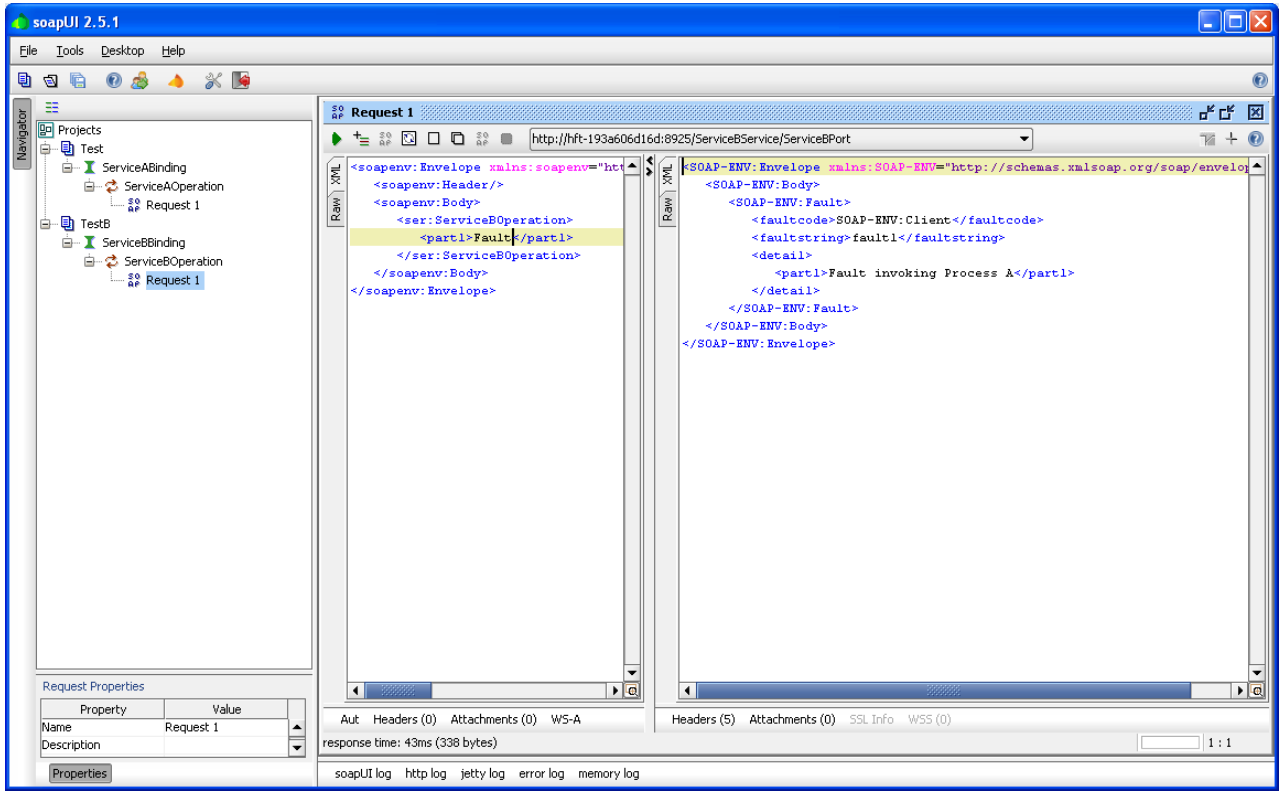




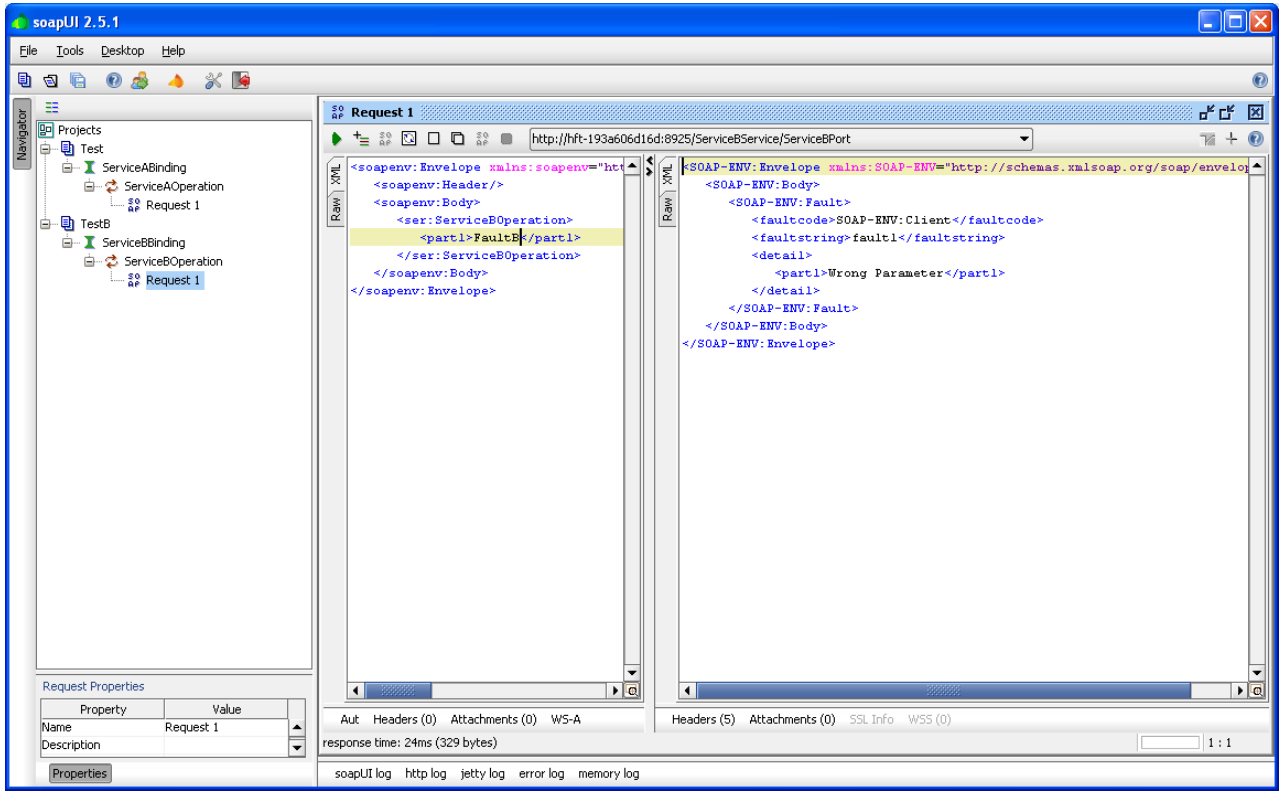
BPEL example: no fault



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)

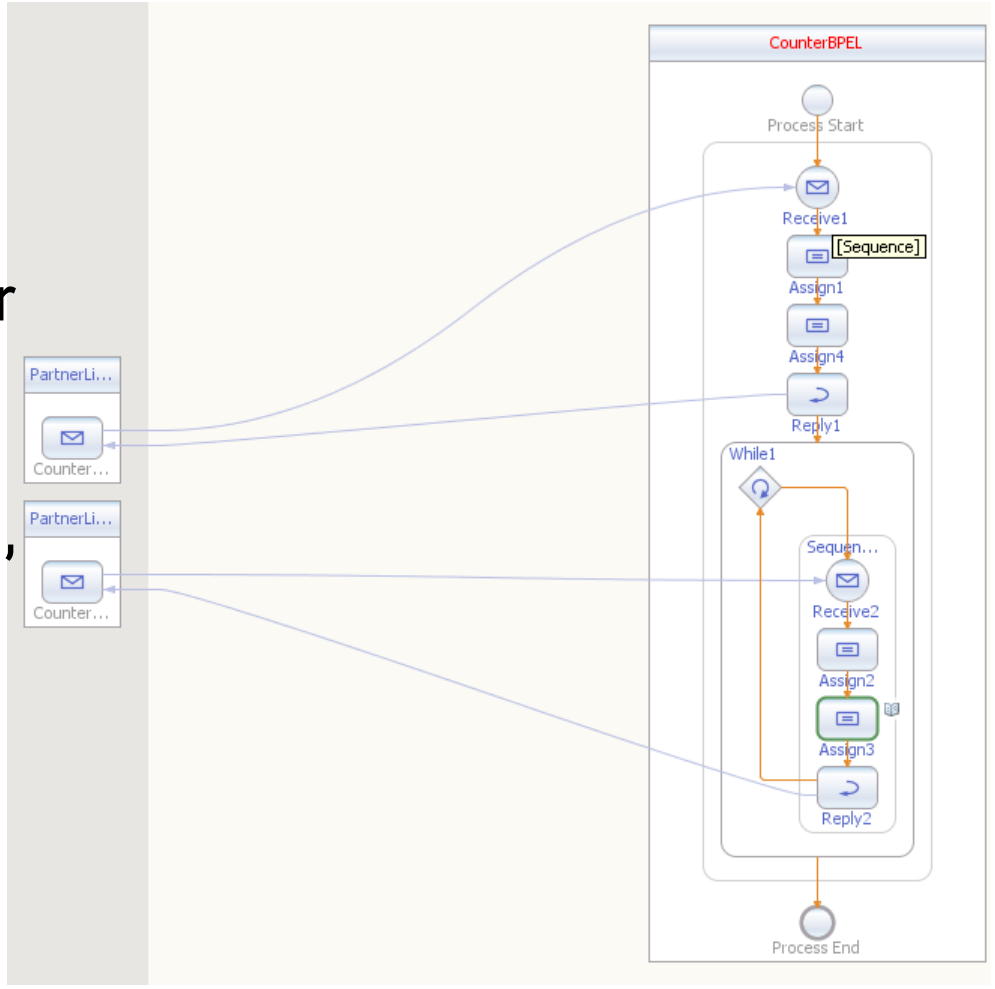
In reality many processes are “long running”
- E.g. a credit approval process (multiple days/weeks/months)

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

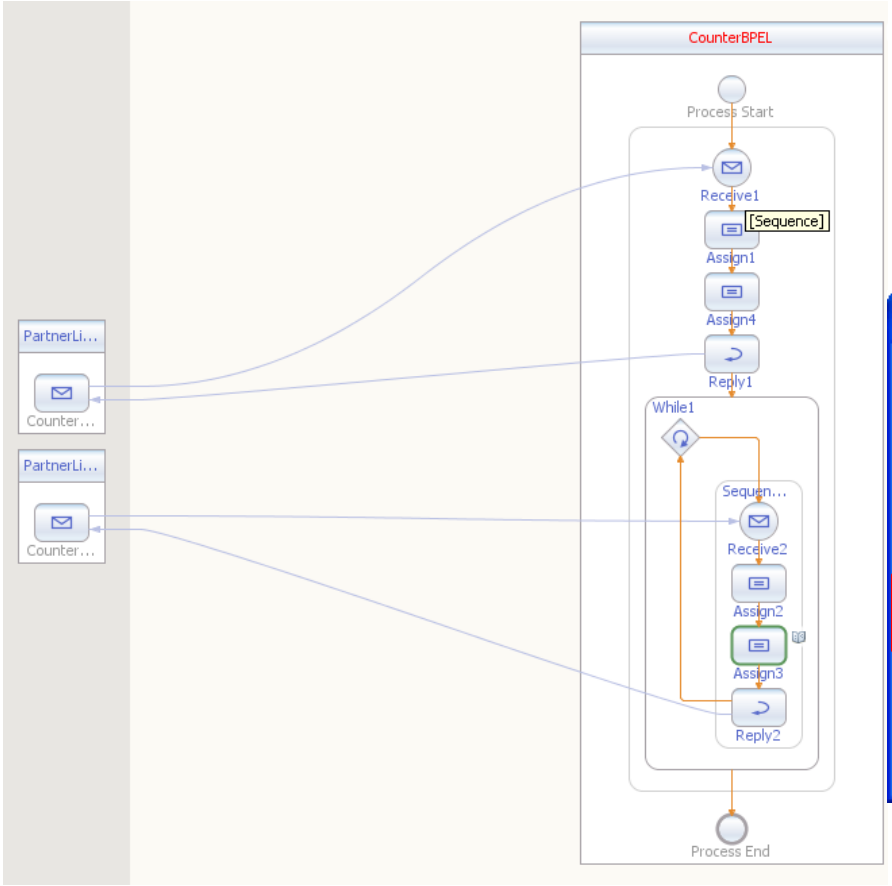
BPEL example: counter

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

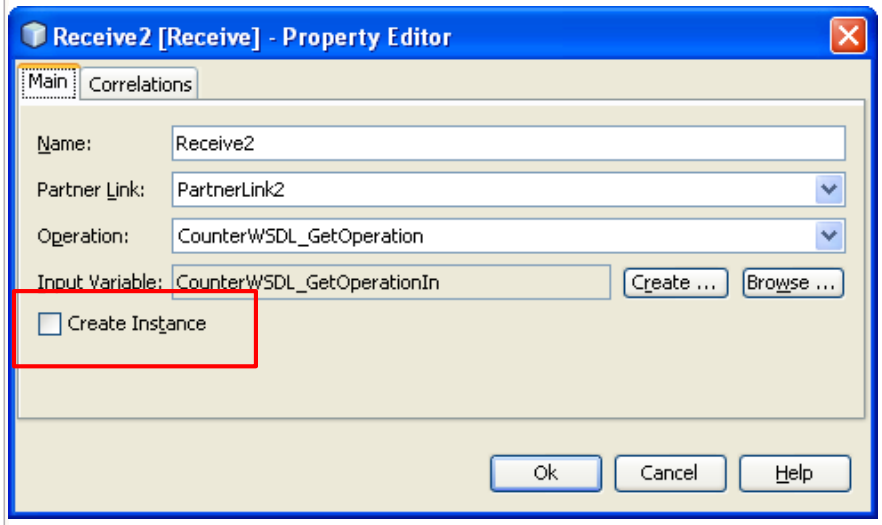
Identification of Instances using correlations



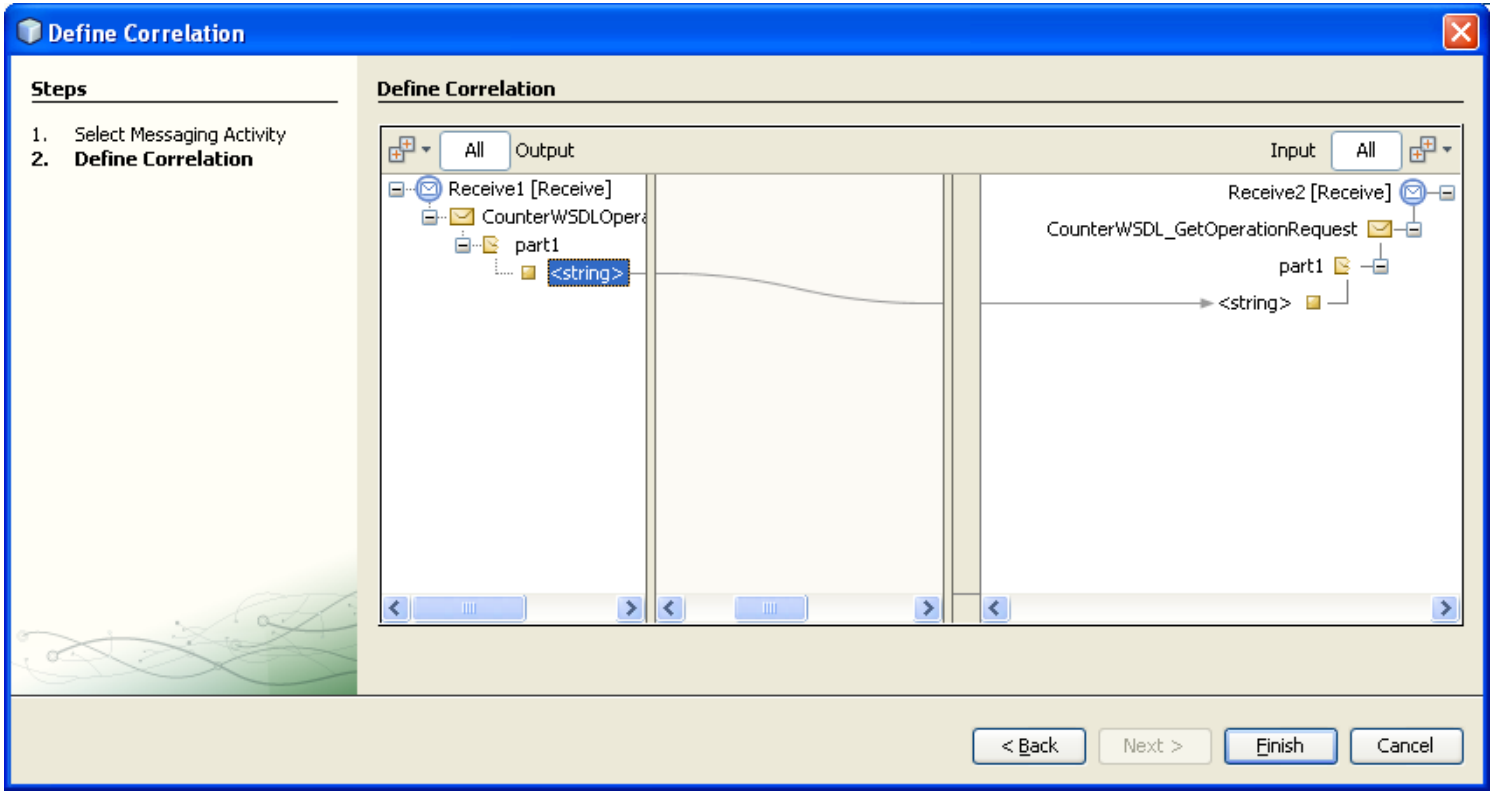
BPEL example: counter



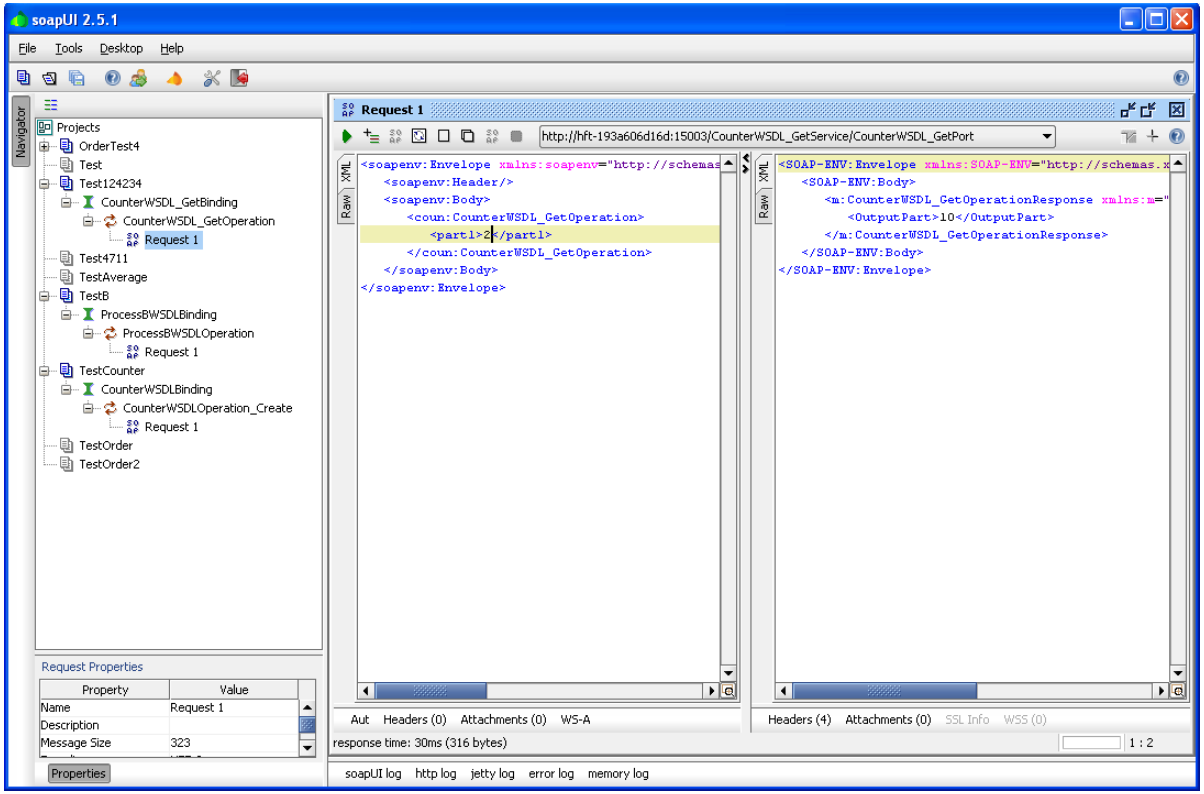
On the second receive no new instance is created



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!