Complex Information Systems

Workflow Management Systems

In this section

- Workflow management via BPMN

Introduction

Two ongoing trends

The globalized world marked.

Need for quick adaptation to changes in the market

Decrease expenses

Make organization more efficient.

Raise of level of abstraction in IT infrastructure

IT systems are exposed as web-services

Service Oriented Architecture

Enables web service composition into business processes

Leeds to Business Process Modeling and Automation

Business Process Modeling

What is a business process?

What is a model of a business process?

What is a business process?

It is the way a company handles a business request, eg. a loan request in a bank, or an incoming order in a shipping company. It is nothing concrete, it is the way people and systems interact to handle a business request.

What is a model of a business process?

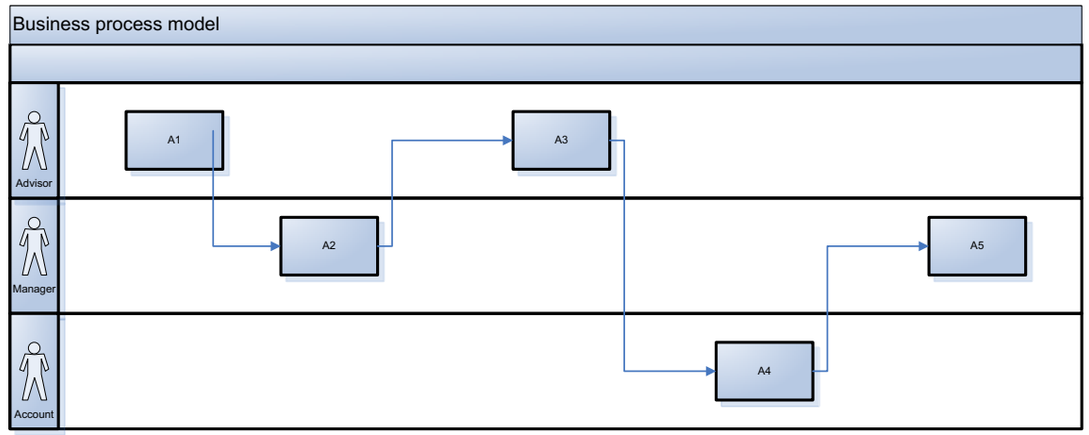
An abstraction of the way people an systems interact to handle a business request described in some kind of language, eg. UML Activity diagrams.

A simplified view of the complex reality

Externalization and formalization of knowledge and expertise within applications and minds.

Business Process Modelling

Real business process



Model of business process

Advantages of modeling the business processes

Better understanding of existing business processes

Documents the business process

Basis for improving existing business processes

Basis for experiencing and simulating new concepts and impact on the organization

Basis for continued optimization

Basis for creating information systems that support the business processes

- One type is known as Workflow Management Systems

Challenges

Difficult to model the world with people and systems interacting together.

Real world process is not understood

Different people has different views of the process

Processes often cross organizational borders

No common vocabulary to use

Many different aspects of a process. It can consist of several models at different abstraction levels linked together.

Business Process Modeling

Live Example: Loan request

Using interviews, analysis and other ethnographic methods, a business process can be surveyed

Traditional process for customer adviser:

Business Procedure:

Receives the loan request at a paper.

Collect information about customer

If loan amount > 10000 Euro, give case to manager

Else make a risk analyses. From result approve or give case to manager.

Requires lot of manual work for customer adviser

Business Process Modeling

Many modeling techniques and tools

DFD, ISAC, SADT, PN, HLPN, PA, FC, UML, BPMN, ...

Simulation tools, design tools, CASE tools, WFMS, ...

A hot standard is BPMN

(Business Process Modeling Notation) by BPMI.org

OMG and BPMI has recently merged.

Why use UML activity diagrams.

Well known standard

Proved its value. Large tool support

Very high expressiveness

Can implement all 20 workflow patterns

Business Process Modeling Notation (BPMN)

Similar to UML activity diagrams

Contains much more symbols-> easier to visualize how the process should behave.

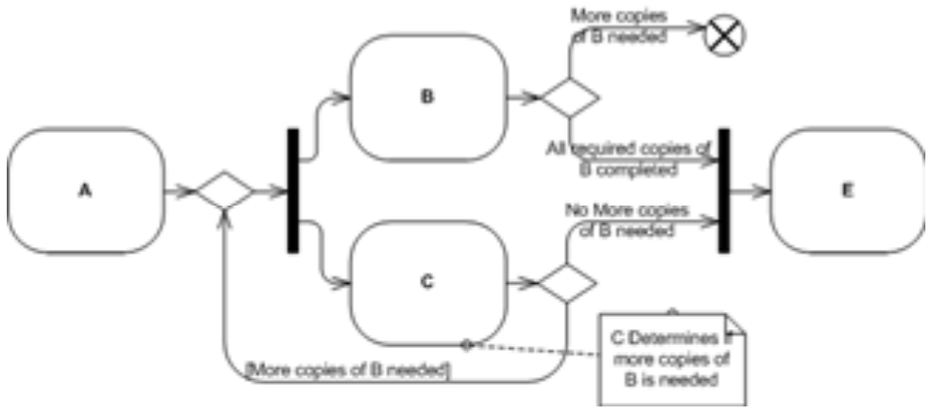
Can model most of the 21 workflow patterns

Business Process Modeling

UML Activity Diagram primer

Consists of

- Actions
- Control flow
- Split and Join
- Decisions
- Swimlanes



Customization of Activity Diagrams

Use of UML profiling for making diagrams specific for business process modeling

Possible to create own activity types by using stereotypes

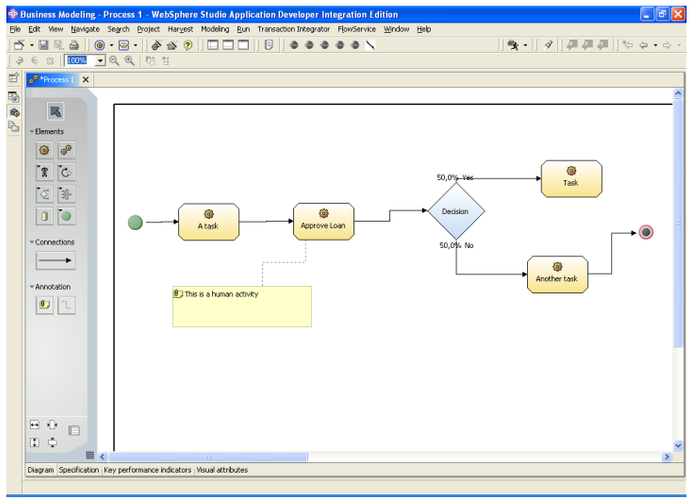
Business Process Modeling

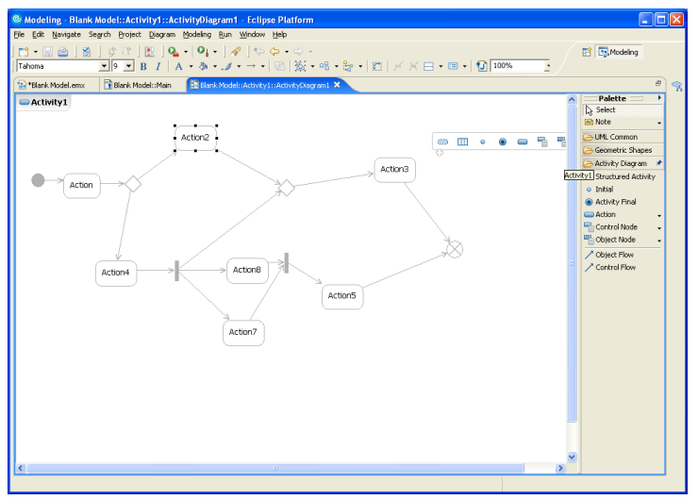
Live Example

Modeling the loan approval process by

BPMN in Web-sphere Business modeler

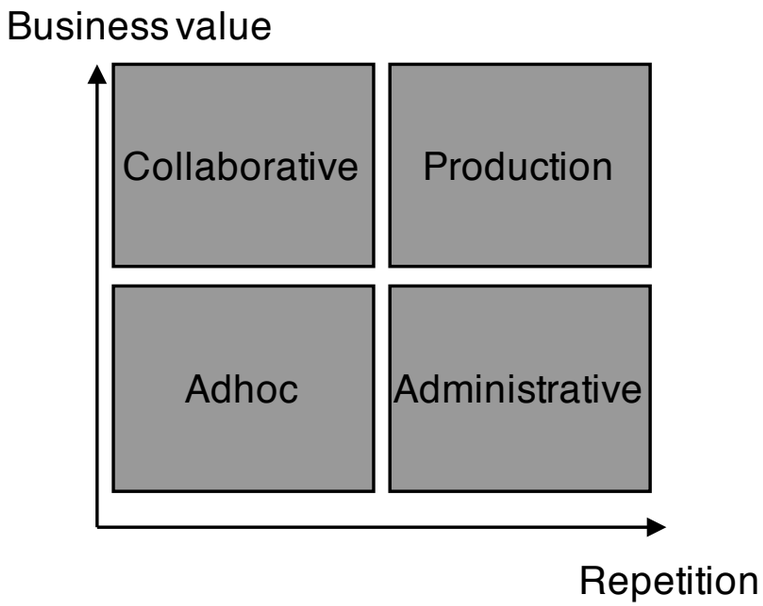
UML Activity diagram in Rational Software Architect



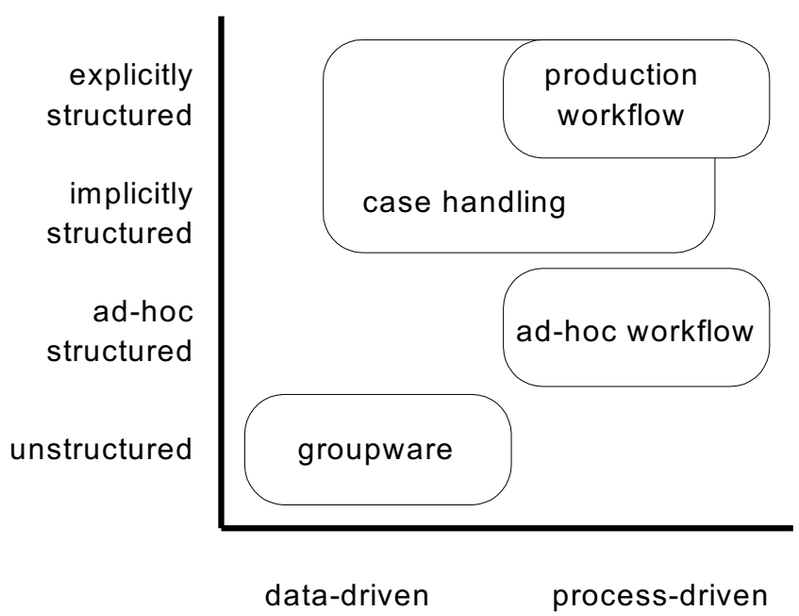


IT support for business processes

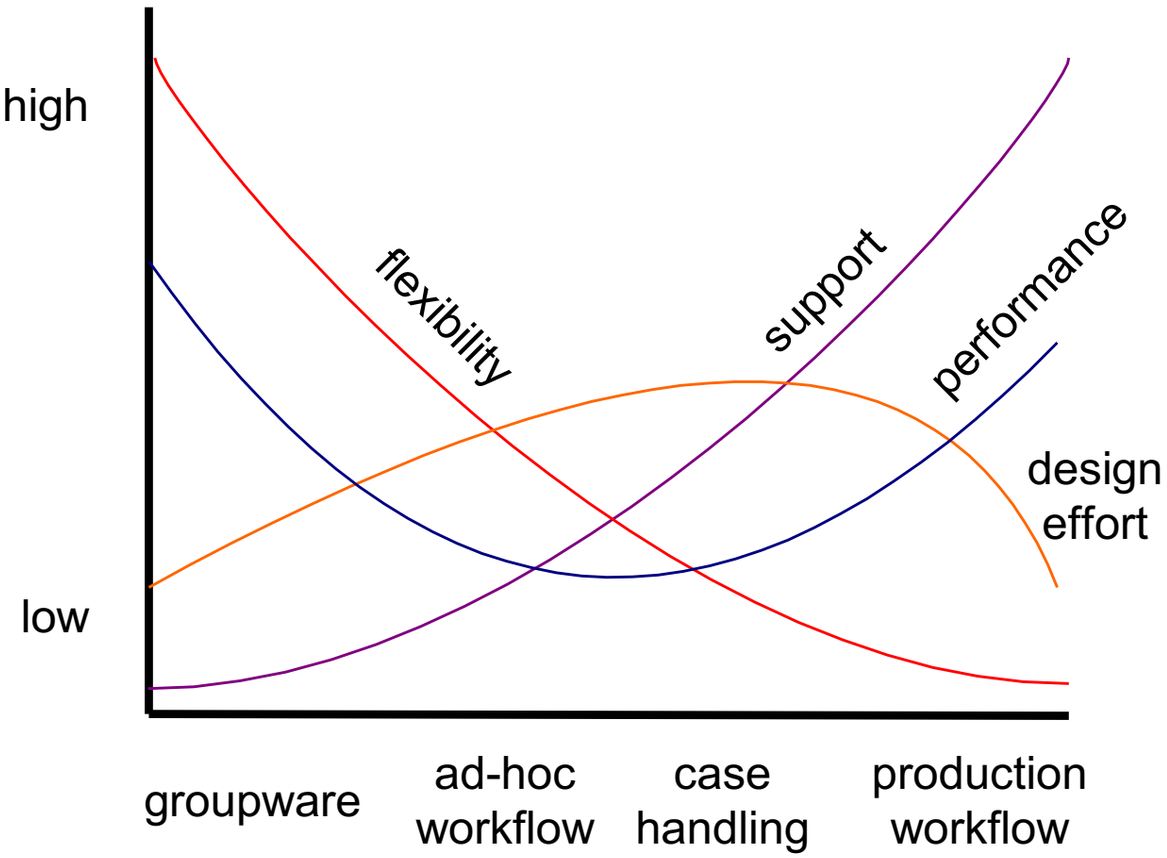
- Business process types



- System types



System trade offs



IT support for business processes

Business process models can be implemented as executable workflows

A Workflow is

also a model of the business process, but contains much more details about technical issues

a program written in a declarative language, most often XML based

executable by a runtime engine, which can interprete the workflow language

IT support for business processes

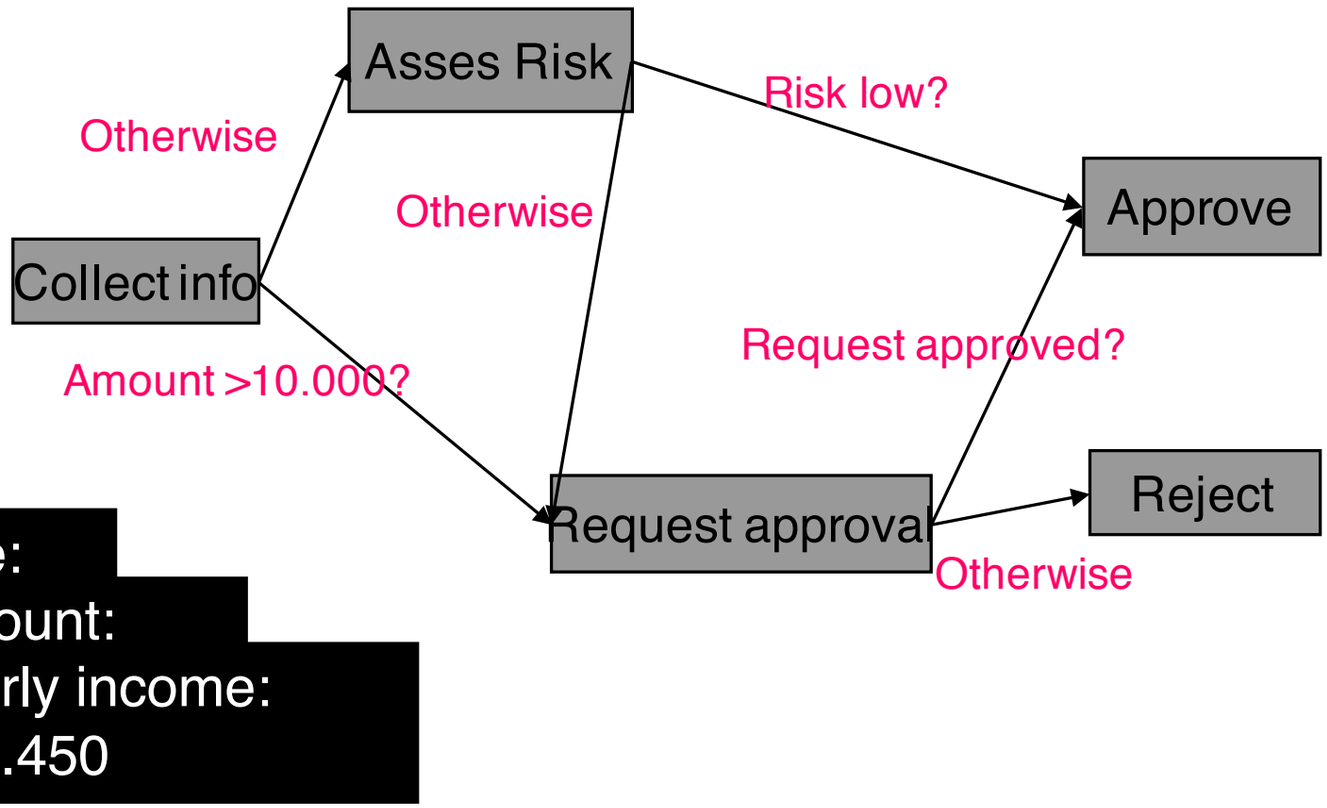
Building blocks in a workflow

Activities

Control flow

Conditions

Data



Age:
Amount:
Yearly income:
231.450

IT support for business processes

Workflow Management System

IT infrastructure to build, execute, monitor and optimize workflows

Consists of

Language,

Some kind of web service composition language, eg. BPEL, BPML.

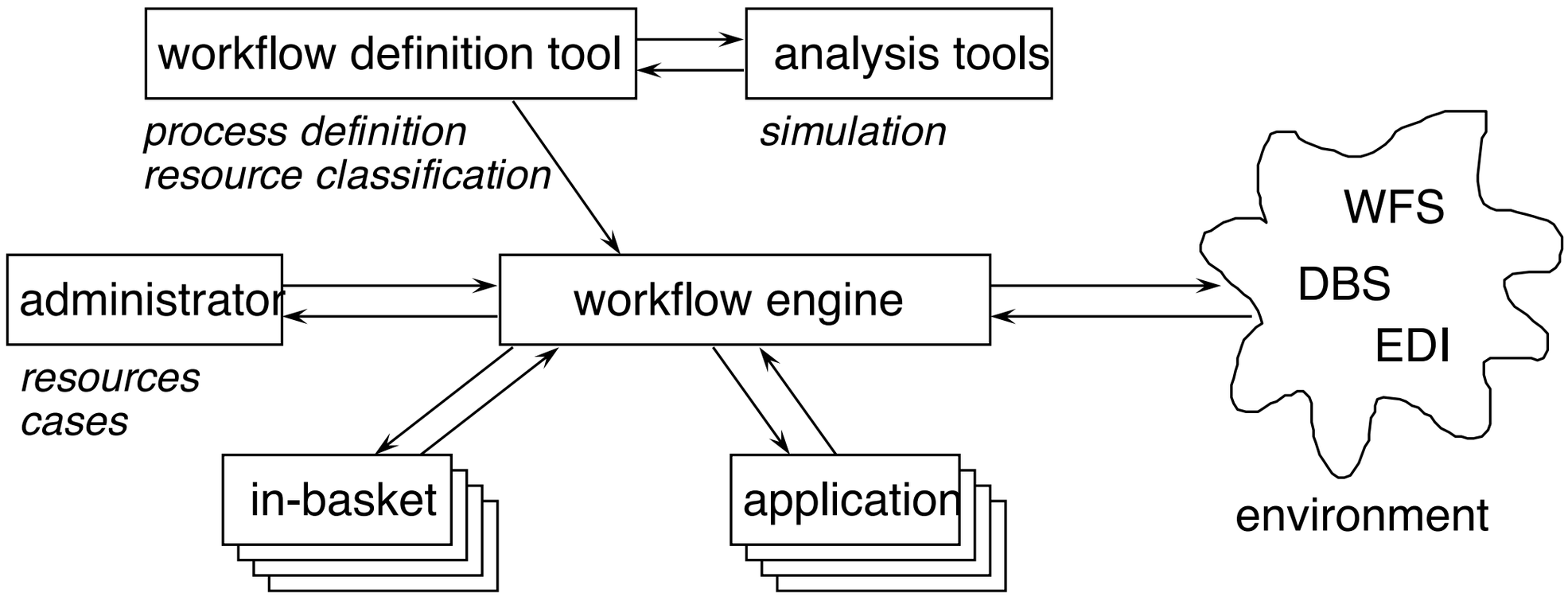
Often based on graphs

Build time IDE to build the workflows. Gives a graphical user interface to the language

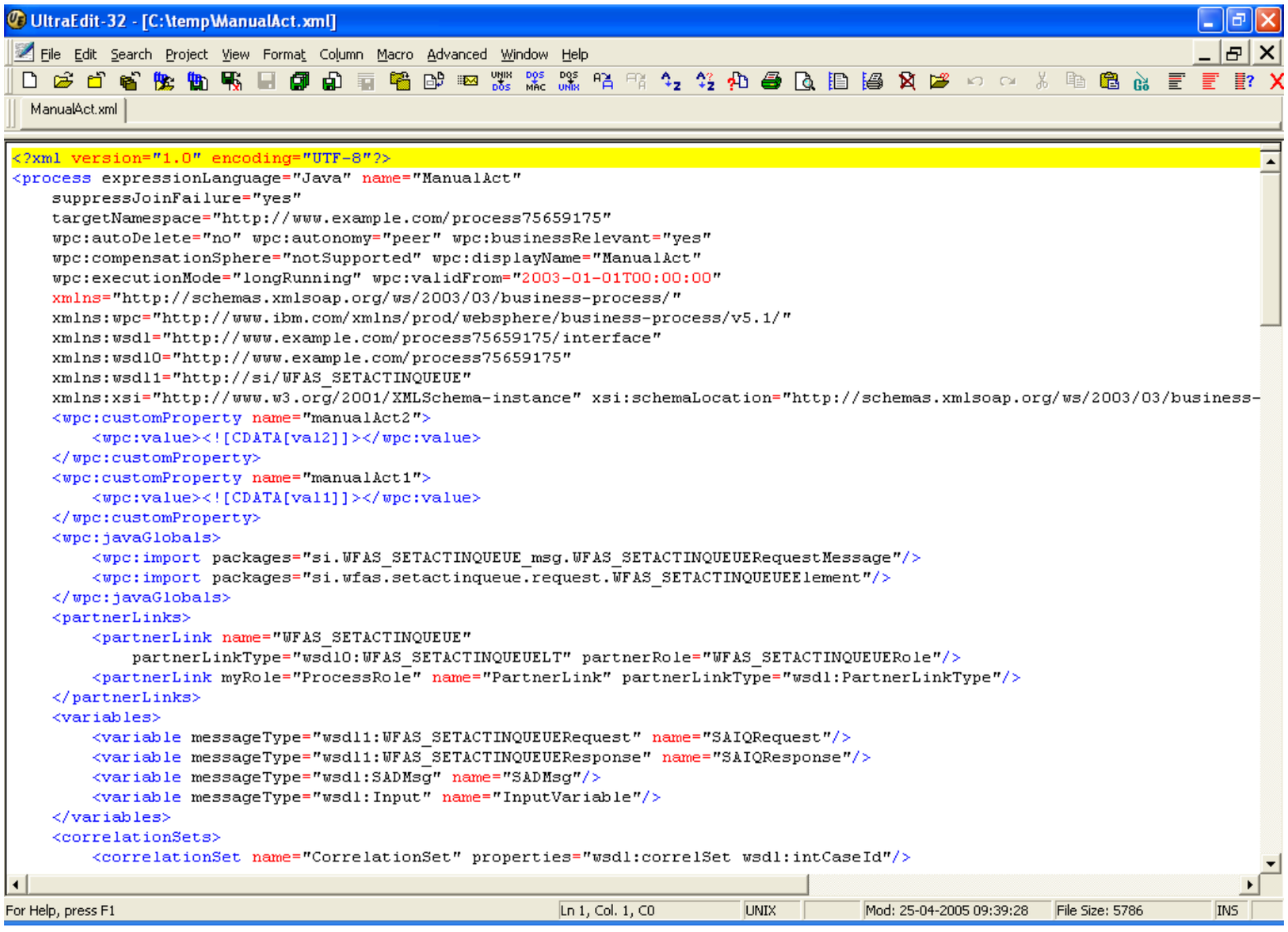
Runtime engine to handle instances of the workflows

Monitor to get overview of running and finished processes

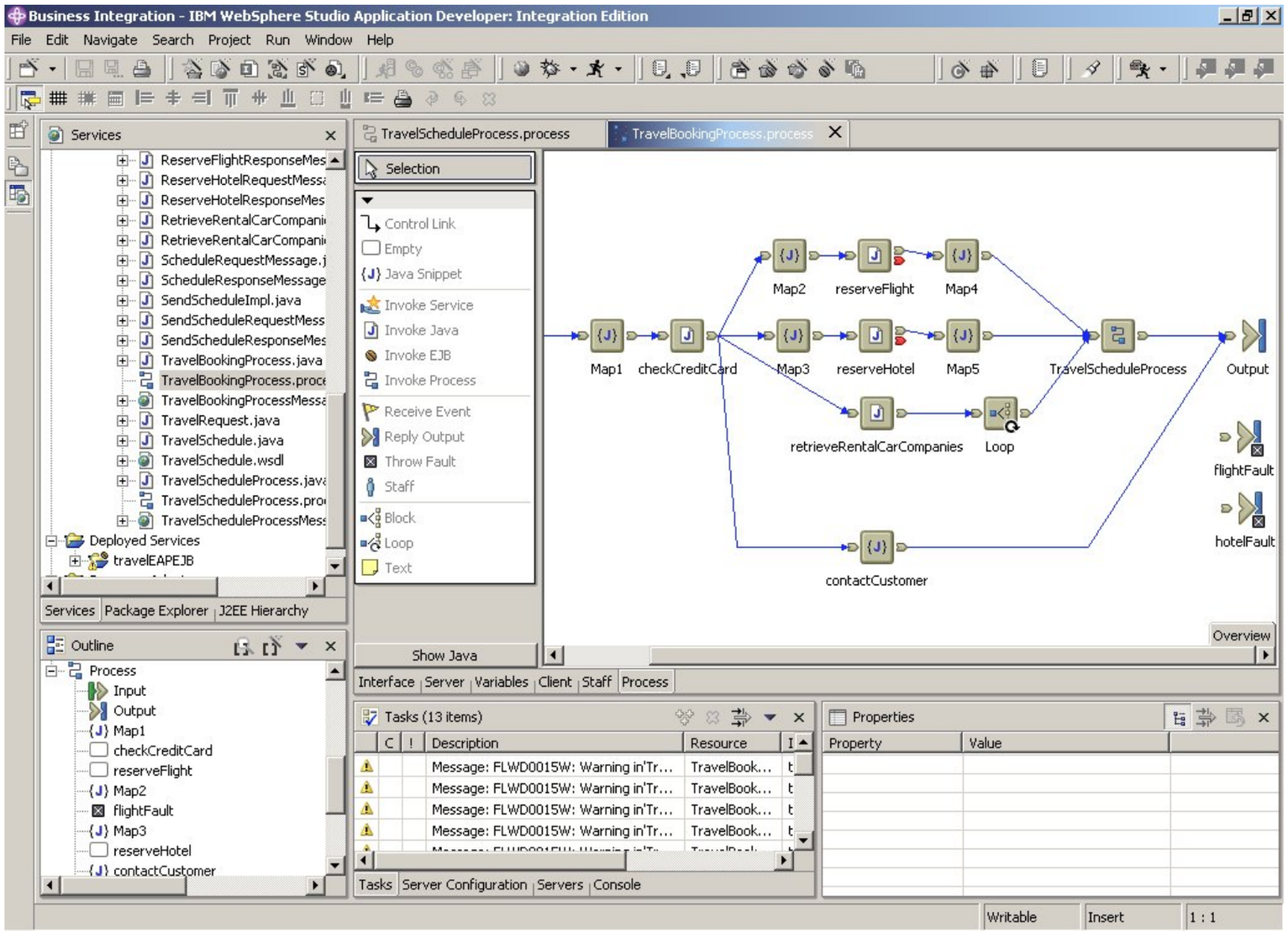
Architecture of a Workflow management System



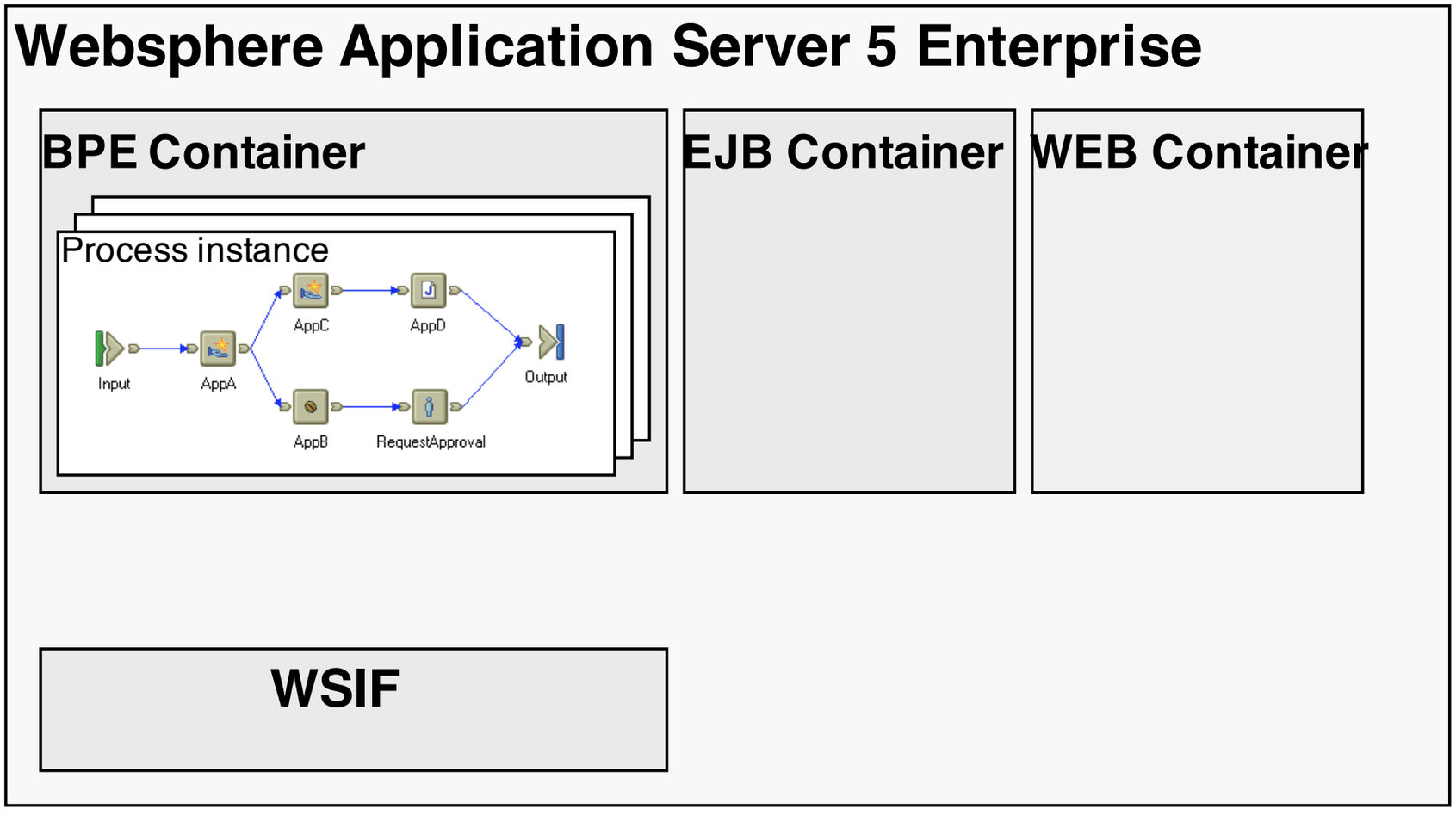
Language


UltraEdit-32 - [C:\temp\ManualAct.xml]
File Edit Search Project View Format Column Macro Advanced Window Help
ManualAct.xml
<?xml version="1.0" encoding="UTF-8"?>
<process expressionLanguage="Java" name="ManualAct"
suppressJoinFailure="yes"
targetNamespace="http://www.example.com/process75659175"
wpc:autoDelete="no" wpc:autonomy="peer" wpc:businessRelevant="yes"
wpc:compensationSphere="notSupported" wpc:displayName="ManualAct"
wpc:executionMode="longRunning" wpc:validFrom="2003-01-01T00:00:00"
xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-process/v5.1/"
xmlns:wsdl="http://www.example.com/process75659175/interface"
xmlns:wsdl0="http://www.example.com/process75659175"
xmlns:wsdl1="http://si/WFAS_SETACTINQUEUE"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://schemas.xmlsoap.org/ws/2003/03/business-
<wpc:customProperty name="manualAct2">
<wpc:value><![CDATA[val2]]></wpc:value>
</wpc:customProperty>
<wpc:customProperty name="manualAct1">
<wpc:value><![CDATA[val1]]></wpc:value>
</wpc:customProperty>
<wpc:javaGlobals>
<wpc:import packages="si.WFAS_SETACTINQUEUE_msg.WFAS_SETACTINQUEUERequestMessage"/>
<wpc:import packages="si.wfas.setactinqueue.request.WFAS_SETACTINQUEUEElement"/>
</wpc:javaGlobals>
<partnerLinks>
<partnerLink name="WFAS_SETACTINQUEUE"
partnerLinkType="wsdl0:WFAS_SETACTINQUEUELT" partnerRole="WFAS_SETACTINQUEUERole"/>
<partnerLink myRole="ProcessRole" name="PartnerLink" partnerLinkType="wsdl:PartnerLinkType"/>
</partnerLinks>
<variables>
<variable messageType="wsdl1:WFAS_SETACTINQUEUERequest" name="SAIQRequest"/>
<variable messageType="wsdl1:WFAS_SETACTINQUEUEResponse" name="SAIQResponse"/>
<variable messageType="wsdl:SADMsg" name="SADMsg"/>
<variable messageType="wsdl:Input" name="InputVariable"/>
</variables>
<correlationSets>
<correlationSet name="CorrelationSet" properties="wsdl:correlSet wsdl:intCaseId"/>
For Help, press F1
Ln 1, Col. 1, C0
UNIX
Mod: 25-04-2005 09:39:28
File Size: 5786
INS

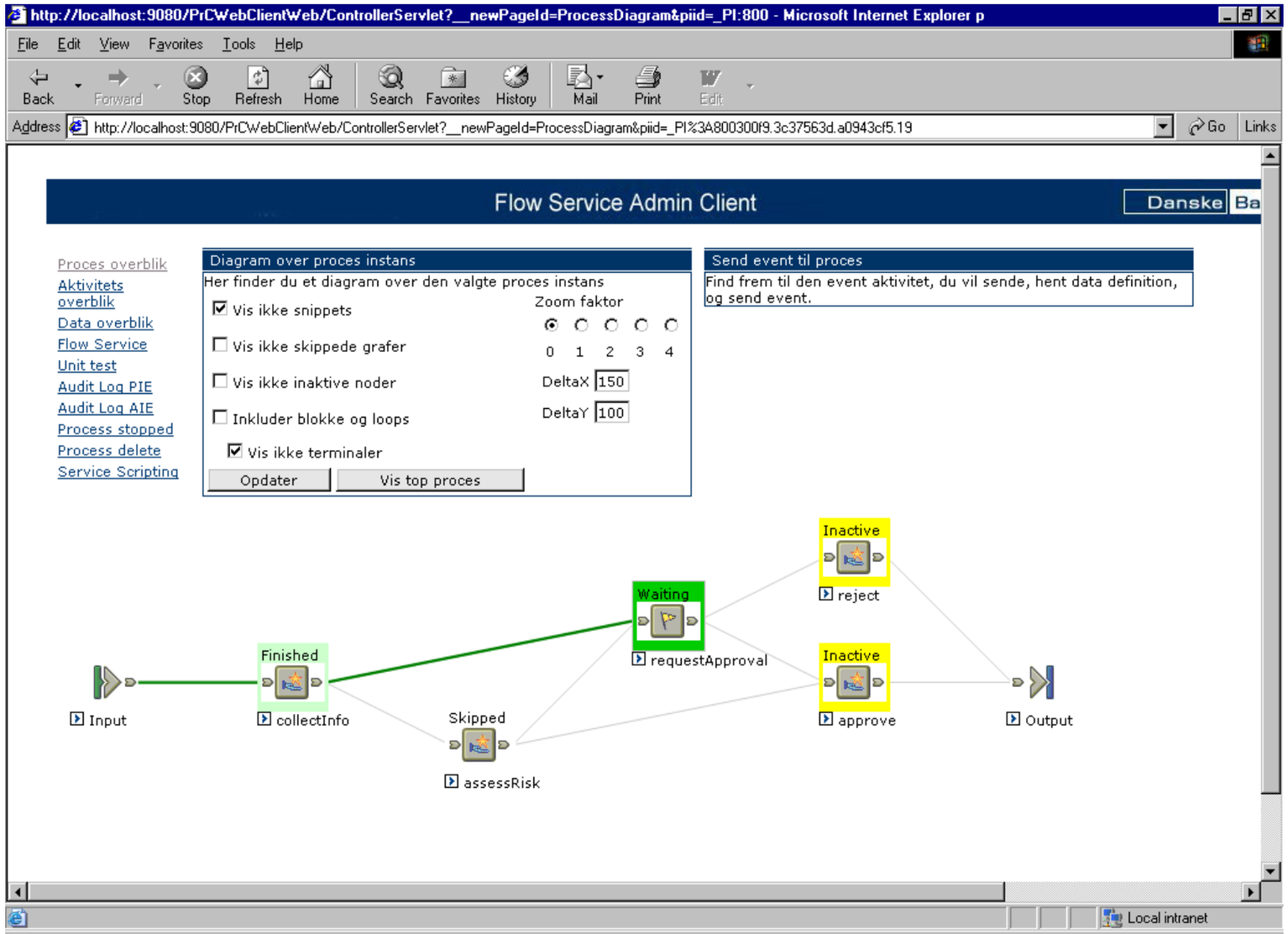
IDE


Business Integration - IBM WebSphere Studio Application Developer: Integration Edition
File Edit Navigate Search Project Run Window Help
Services
ReserveFlightResponseMes
ReserveHotelRequestMess
ReserveHotelResponseMes
RetrieveRentalCarCompani
RetrieveRentalCarCompani
ScheduleRequestMessage.j
ScheduleResponseMessage
SendScheduleImpl.java
SendScheduleRequestMess
SendScheduleResponseMes
TravelBookingProcess.java
TravelBookingProcess.proce
TravelBookingProcessMessa
TravelRequest.java
TravelSchedule.java
TravelSchedule.wsdl
TravelScheduleProcess.java
TravelScheduleProcess.pro
TravelScheduleProcessMess
Deployed Services
travelEAPEJB
Services Package Explorer J2EE Hierarchy
Outline
Process
Input
Output
Map1
checkCreditCard
reserveFlight
Map2
flightFault
Map3
reserveHotel
contactCustomer
TravelScheduleProcess.process
TravelBookingProcess.process
Selection
Control Link
Empty
Java Snippet
Invoke Service
Invoke Java
Invoke EJB
Invoke Process
Receive Event
Reply Output
Throw Fault
Staff
Block
Loop
Text
Show Java
Map2
reserveFlight
Map4
Map1
checkCreditCard
Map3
reserveHotel
Map5
TravelScheduleProcess
Output
retrieveRentalCarCompanies
Loop
flightFault
hotelFault
contactCustomer
Overview
Interface Server Variables Client Staff Process
Tasks (13 items)
C ! Description Resource
Message: FLWD0015W: Warning in'Tr... TravelBook...
Tasks Server Configuration Servers Console
Properties
Property Value
Writable Insert 1 : 1

Runtime


Websphere Application Server 5 Enterprise
BPE Container
Process instance
Input
AppA
AppC
AppB
AppD
RequestApproval
Output
EJB Container
WEB Container
WSIF

Monitor


http://localhost:9080/PrCWebClientWeb/ControllerServlet?__newPageId=ProcessDiagram&piid=_PI:800 - Microsoft Internet Explorer p
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address http://localhost:9080/PrCWebClientWeb/ControllerServlet?__newPageId=ProcessDiagram&piid=_PI%3A800300f9.3c37563d.a0943cf5.19
Go Links
Flow Service Admin Client
Danske Ba
Proces overblik
Aktivitets overblik
Data overblik
Flow Service
Unit test
Audit Log PIE
Audit Log AIE
Process stopped
Process delete
Service Scripting
Diagram over proces instans
Her finder du et diagram over den valgte proces instans
Vis ikke snippets
Vis ikke skippede grafer
Vis ikke inaktive noder
Inkluder blokke og loops
Vis ikke terminaler
Opdater
Vis top proces
Zoom faktor
0 1 2 3 4
DeltaX 150
DeltaY 100
Send event til proces
Find frem til den event aktivitet, du vil sende, hent data definition, og send event.
Input
Finished
collectInfo
Skipped
assessRisk
Waiting
requestApproval
Inactive
reject
Inactive
approve
Output
Local intranet

Advantages of Workflow Management

Higher productivity

Moves knowledge from people to documented process

Rapid adaptation to the market

Location of bottlenecks and runtime changes of process

Statistics about processes

Continued optimization

Reuse of services and processes

All processes are implemented in same framework

Requirements for implementing WFM system

Well established IT infrastructure

SOA: Applications called as services

Resources to model the business, to implement them and to support and monitor the running instances.

The organization must be ready for change. People get new ways of working.

Skilled people!

Challenges

Difficult to map from business process model to workflow

Services to be called are not documented or required data is missing in the process

Workflows can be very complex and they can be difficult to comprehend

Limitations in the WFM system compared to requirements

Many technologies and platforms are involved. It requires a lot of the process developer.

Development Process

Business analyst

Defines the AsIs, ToBe and the makroflow

IT Architect

- Ensures the makroflow conforms to the IT infrastructure
- Enriches the makroflow with information about the IT infrastructure
- Defines service interfaces

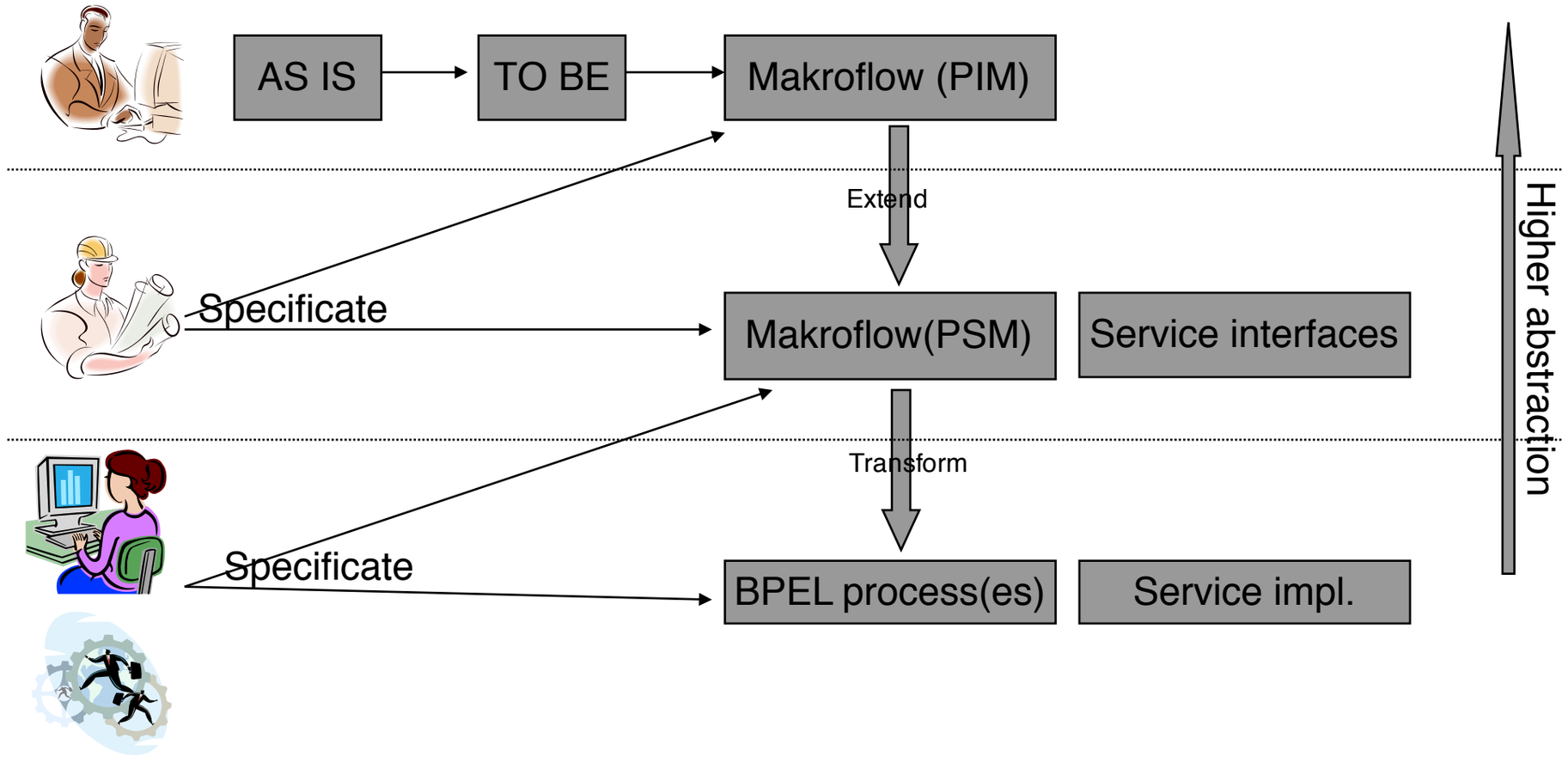
Process developer

Takes the makroflow and transforms it into “code”, eg. BPEL

Service developer

Develops implementations to the specified service interfaces

Development Process



Challenges

Models at different abstraction levels are not synchronized. Changes have to be made manually in all models.

The gab between makro-flow and code is large. Many changes has to be made in the code.

When a change is made in the business model, the same changes has to be made in all levels below.

- High risk of conflicts between models
- High risk of making coding errors

Difficult to coordinate work between the people with very different mindset

The Model Driven Engineering paradigm

The programming level is raised from code to models

The code is generated from the models

The process developer should work on the makro-flow and the code, eg. BPEL should be generated from the makro-flow

Working on higher abstraction level speeds up development time, decreases the amount of errors and decreases the reaction time to changes in the marked

Is still in a very early stage.

Workflow Patterns

21 patterns described by Vil van der Aalst

- Basic Patterns
- Advanced Branch and Join patterns
- Structural patterns
- Multiple Instances patterns
- State based patterns
- Cancellation patterns

Used to evaluate Business Process languages

Additional Patterns

- Communication patterns
- Human interaction patterns
- Enterprise specific patterns

Web Services Composition

Composition/Orchestration of web services into executable business processes/workflows.

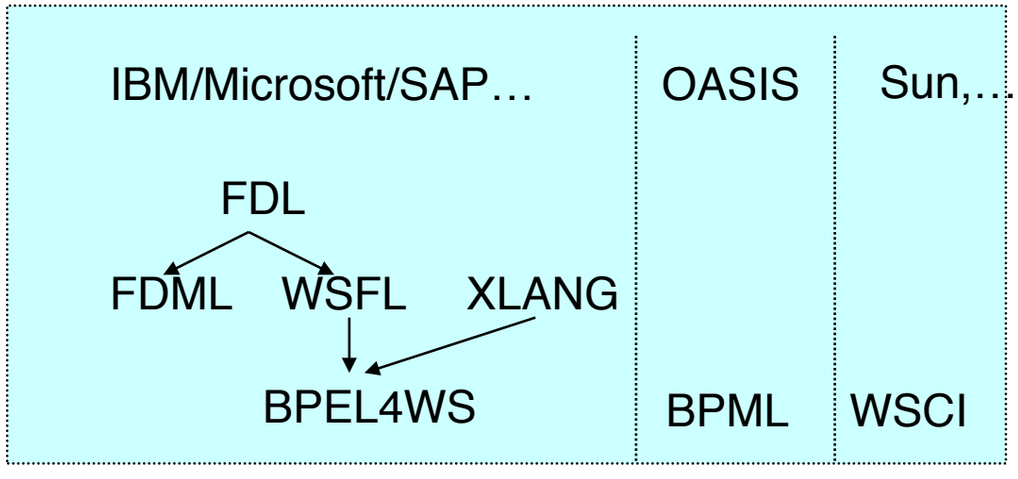
A Workflow Management System handles the definition and execution

Languages based on XML schemas

Builds on top of Web Services (WSDL)

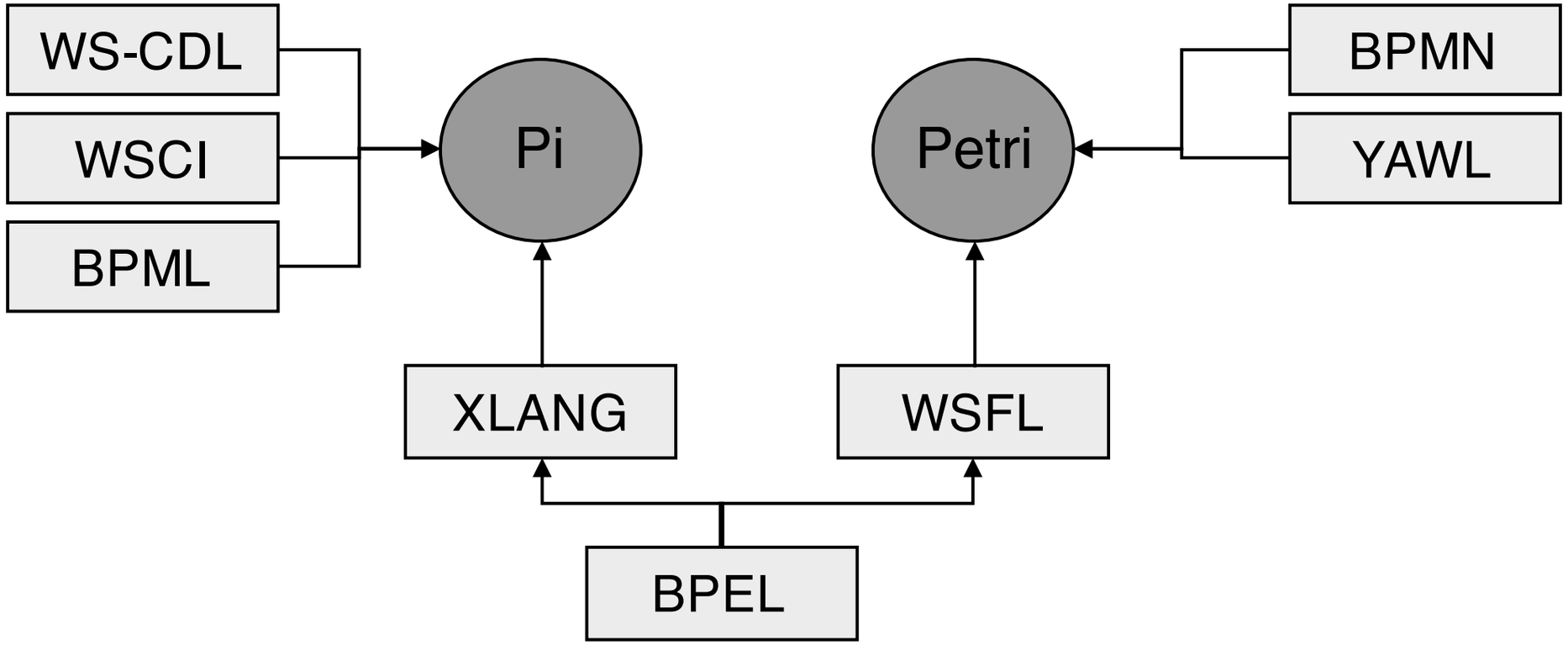
Many different languages

- BPEL4WS
- BPML
- WS-CDL
- WSCI
- WSFL
- XLANG
- FDML„
- FDL
- BPSS
- XPDL



Theoretical foundation of languages

Based on Petri net and / or Pi calculus



Technology stack

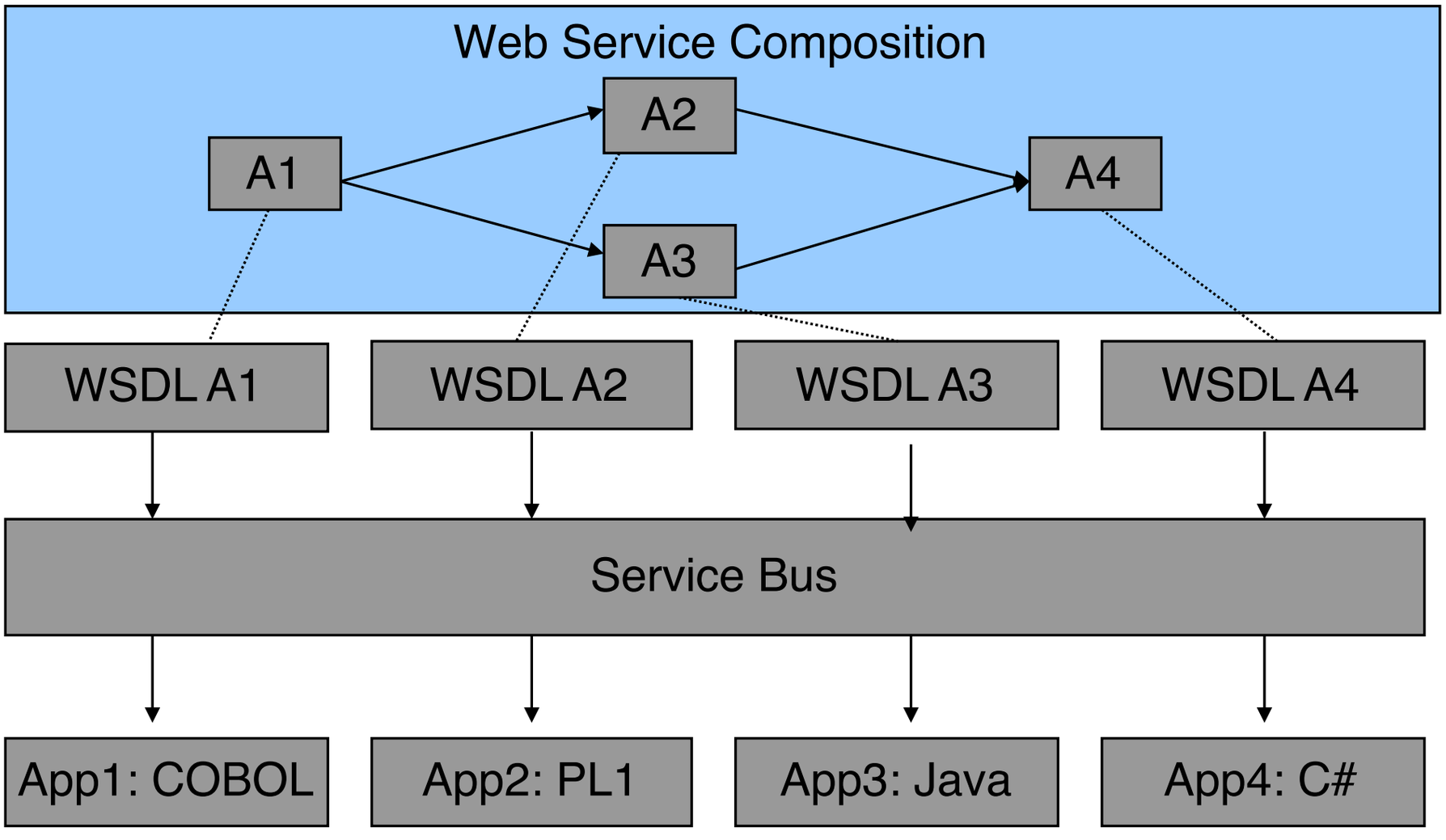
Web Service Composition: BPEL4WS, WSCI, etc.

Service Description layer: WSDL

XML Messaging Layer: SOAP

Transport Layer: HTTP, SMTP, FTP, JMS,,etc.

Web Services Composition



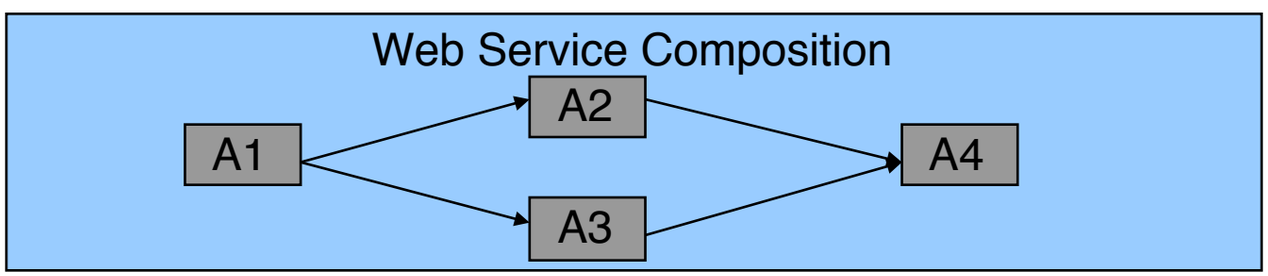
Execution engine

Interprets workflows based on a language

Knowns how to call Web Services through different protocols

Enables long running transactions (days, weeks, month) by persisting state information for the workflow

Enables ACID transactions and compensation in case of exceptions



BPEL4WS

Business Process Execution Language for Web Services

Defined by IBM, Microsoft, SAP, BEA, Siebel

Builds on top of XML and Web Services technology stack

Extensible for new language elements

Eg. BPELJ

IBM BPEL: Staff activity, Java snippet, Control links

Is like a programming language combined with graphically representation

BPEL4WS

Properties of a BPEL process

Is itself a web service. Has a WSDL interface

Can be synchronous or asynchronous

A BPEL process consists of

Activities

Structured activities

Control flow

Variables

Partner links

BPEL4WS

Activities

Waiting for the client to invoke the business process through sending a message, using <receive> (receiving a request)

Invoking other web services, using <invoke>

Generating a response for synchronous operations, using <reply>

Manipulating data variables, using <assign>

Indicating faults and exceptions, using <throw>

Waiting for some time, using <wait>

Terminating the entire process, using <terminate>, etc.

BPEL4WS

Structured Activities

Sequence (<sequence>), which allows us to define a set of activities that will be invoked in an ordered sequence

Flow (<flow>) for defining a set of activities that will be invoked in parallel

Case-switch construct (<switch>) for implementing branches

While (<while>) for defining loops

The ability to select one of a number of alternative paths, using <pick>

BPEL4WS

Control Flow

Defines the sequence of the activities

True/false evaluation is attached to each control link

Variables

Defines data for service invocations

Partner links

Defines all services to be called or which can call the process. A partner link is linked to a WSDL file

BPEL4WS

A BPEL process contains at least:

A client partner link and a corresponding receive activity to start the process

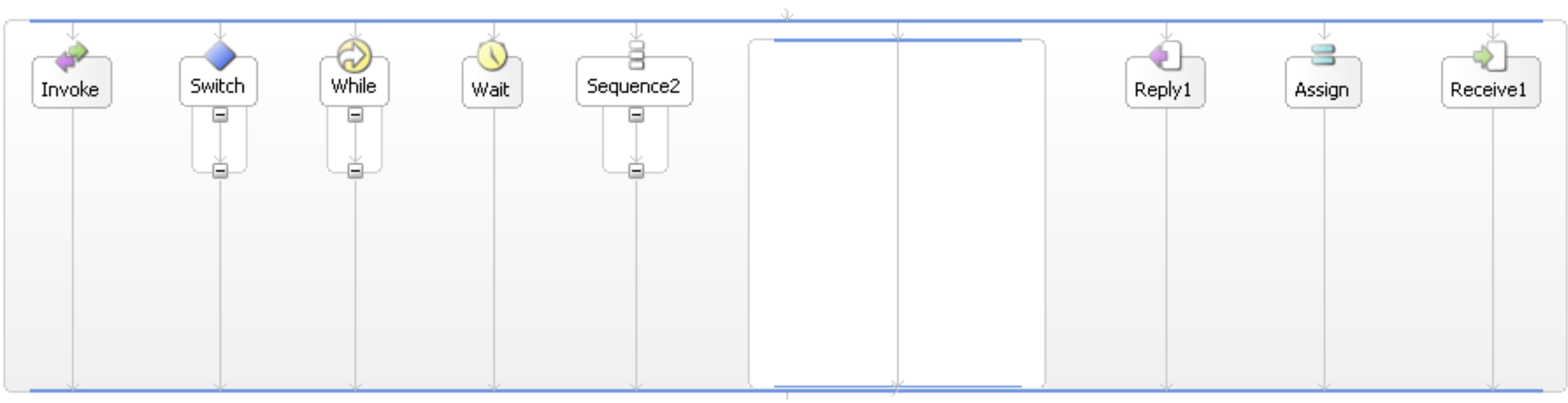
Normally at least one invoke activity and a partner link

At least one variable for the input data to the process

Allways two variables for an invoke activity

BPEL4WS

Graphical notation (IBM)



BPEL4WS

Select insurance offer process

Define 3 partner links

```
<?xml version="1.0" encoding="utf-8"?>

<process name="insuranceSelectionProcess"
         targetNamespace="http://packtpub.com/bpel/example/"
         xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
         xmlns:ins="http://packtpub.com/bpel/insurance/"
         xmlns:com="http://packtpub.com/bpel/company/" >

   <partnerLinks>
      <partnerLink name="client"
                   partnerLinkType="com:selectionLT"
                   myRole="insuranceSelectionService"/>

      <partnerLink name="insuranceA"
                   partnerLinkType="ins:insuranceLT"
                   myRole="insuranceRequester"
                   partnerRole="insuranceService"/>

      <partnerLink name="insuranceB"
                   partnerLinkType="ins:insuranceLT"
                   myRole="insuranceRequester"
                   partnerRole="insuranceService"/>

   </partnerLinks>
..
```

BPEL4WS

Variables for input to process and for services

```
...
   <variables>
      <!-- input for BPEL process -->
      <variable name="InsuranceRequest"
                messageType="ins:InsuranceRequestMessage"/>
      <!-- output from insurance A -->
      <variable name="InsuranceAResposne"
                messageType="ins:InsuranceResponseMessage"/>
      <!-- output from insurance B -->
      <variable name="InsuranceBResposne"
                messageType="ins:InsuranceResponseMessage"/>
      <!-- output from BPEL process -->
      <variable name="InsuranceSelectionResponse"
                messageType="ins:InsuranceResponseMessage"/>
   </variables>
...
```

BPEL4WS

Process steps


```
...
   <sequence>

      <!-- Receive the initial request from client -->
      <receive partnerLink="client"
               portType="com:InsuranceSelectionPT"
               operation="SelectInsurance"
               variable="InsuranceRequest"
               createInstance="yes" />

   <!-- Make concurrent invocations to Insurance A and B -->
   <flow>

      <!-- Invoke Insurance A web service -->
      <invoke partnerLink="insuranceA"
              portType="ins:ComputeInsurancePremiumPT"
              operation="ComputeInsurancePremium"
              inputVariable="InsuranceRequest"
              outputVariable="InsuranceAResposne" />

      <!-- Invoke Insurance B web service -->
      <invoke partnerLink="insuranceB"
              portType="ins:ComputeInsurancePremiumPT"
              operation="ComputeInsurancePremium"
              inputVariable="InsuranceRequest"
              outputVariable="InsuranceBResposne" />

   </flow>
```

```
   <!-- Select the best offer and construct the response -->
   <switch>

     <case condition="bpws:getVariableData('InsuranceAResposne',
                      'confirmationData','/confirmationData/Amount')
                   <= bpws:getVariableData('InsuranceBResposne',
                      'confirmationData','/confirmationData/Amount')">
        <!-- Select Insurance A -->
        <assign>
          <copy>
            <from variable="InsuranceAResposne" />
            <to variable="InsuranceSelectionResponse" />
          </copy>
        </assign>
      </case>

      <otherwise>
        <!-- Select Insurance B -->
        <assign>
          <copy>
            <from variable="InsuranceBResposne" />
            <to variable="InsuranceSelectionResponse" />
          </copy>
        </assign>
      </otherwise>
    </switch>

    <!-- Send a response to the client -->
    <reply partnerLink="client"
           portType="com:InsuranceSelectionPT"
           operation="SelectInsurance"
           variable="InsuranceSelectionResponse"/>

  </sequence>
</process>
```

BPEL4WS

Still need to define the three WSDL files

Complex and hard to model directly in XML

Several BPEL tools exists at the market

Examples are made with IBM Websphere Application Developer Integration Edition (WSADIE)