



Process Modelling and Standardization

Jan Mendling
Dept. of IS and New Media,
WU Vienna, Austria
jan.mendling@wu-wien.ac.at



Agenda



- 1. Business Process Management and Lifecycle**
- 2. Process Design**
 - Process Implementation**
 - Conclusions**

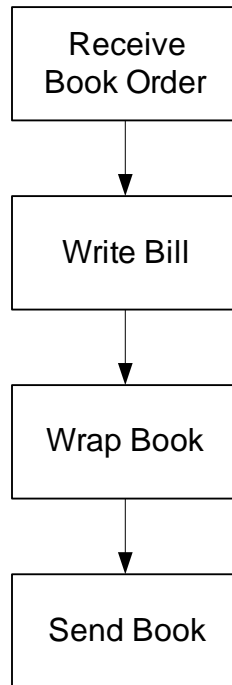


Business Process Management and Lifecycle

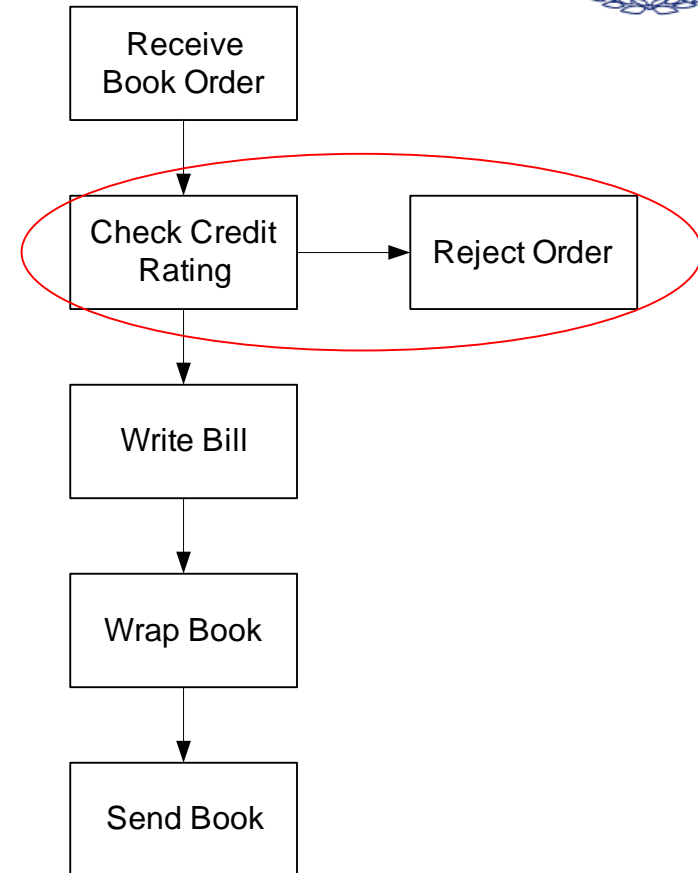
Why Processes and Workflow?



Old process



New process



Why Processes and Workflow?



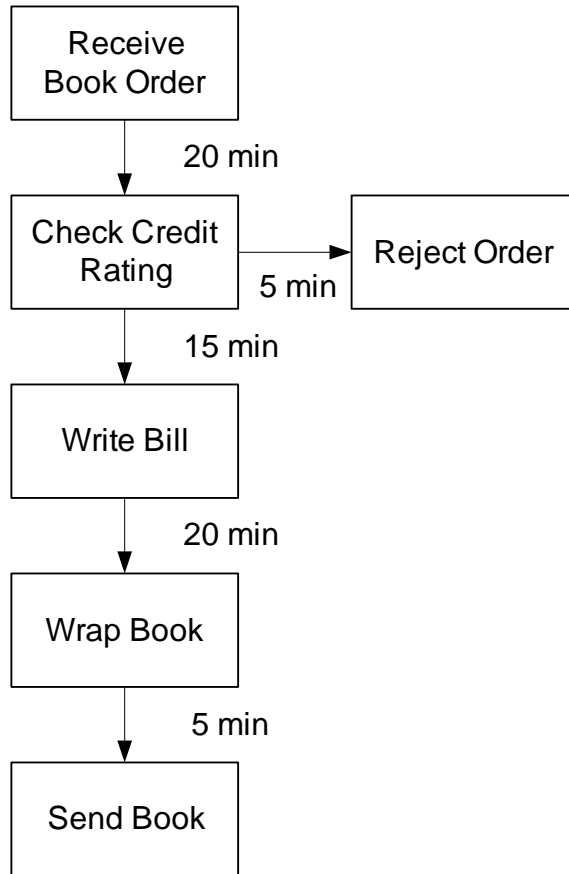
1st Reason: Flexibility

- Introduction of a separate process layer
- Functions tend to be stable
- Ordering of functions is subject to change
- Processes become more easy to adapt

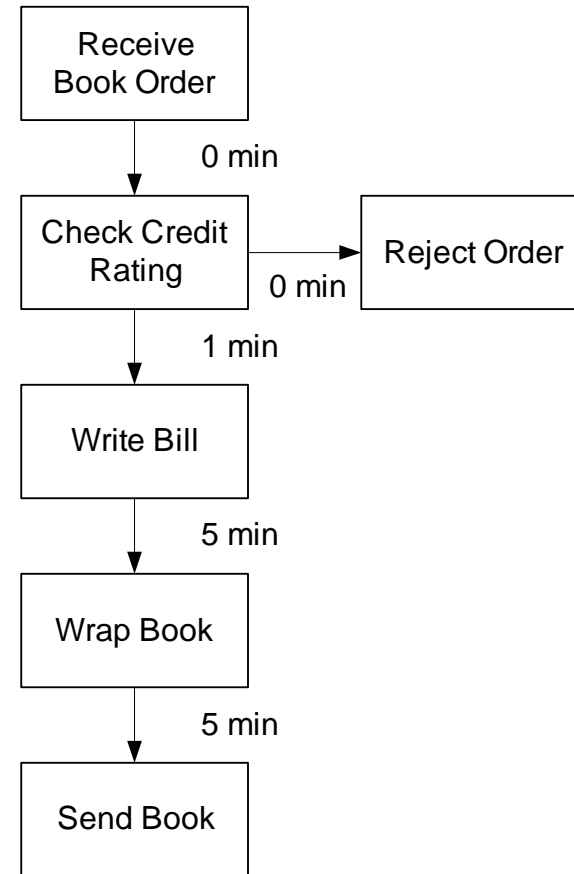
Why Processes and Workflow?



Manual process



Automatic process



Why Processes and Workflow?



2nd Reason: Productivity

- **Waiting times between functions minimized**
- **Automatic routing of work items**

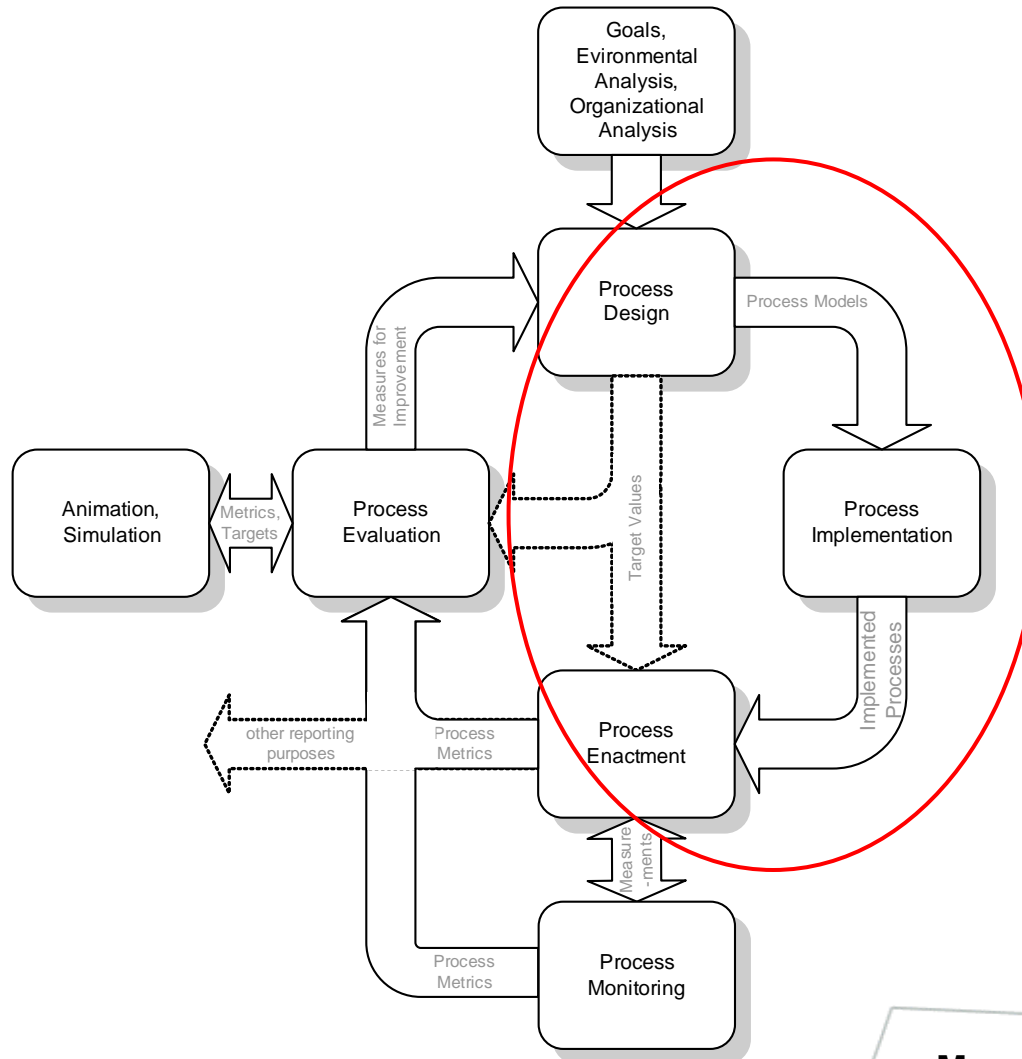
Why Processes and Workflow?



more reasons:

- **Better customer orientation**
- **Better controlling of processes**
- **Better documentation of enterprise**
- **Better communication between different departments**
- **Avoiding problems**
- **...**

Business Process Lifecycle



Modelling Languages for Business Processes and Workflow

Event-Driven Process
Chains (EPC)

Process
Design

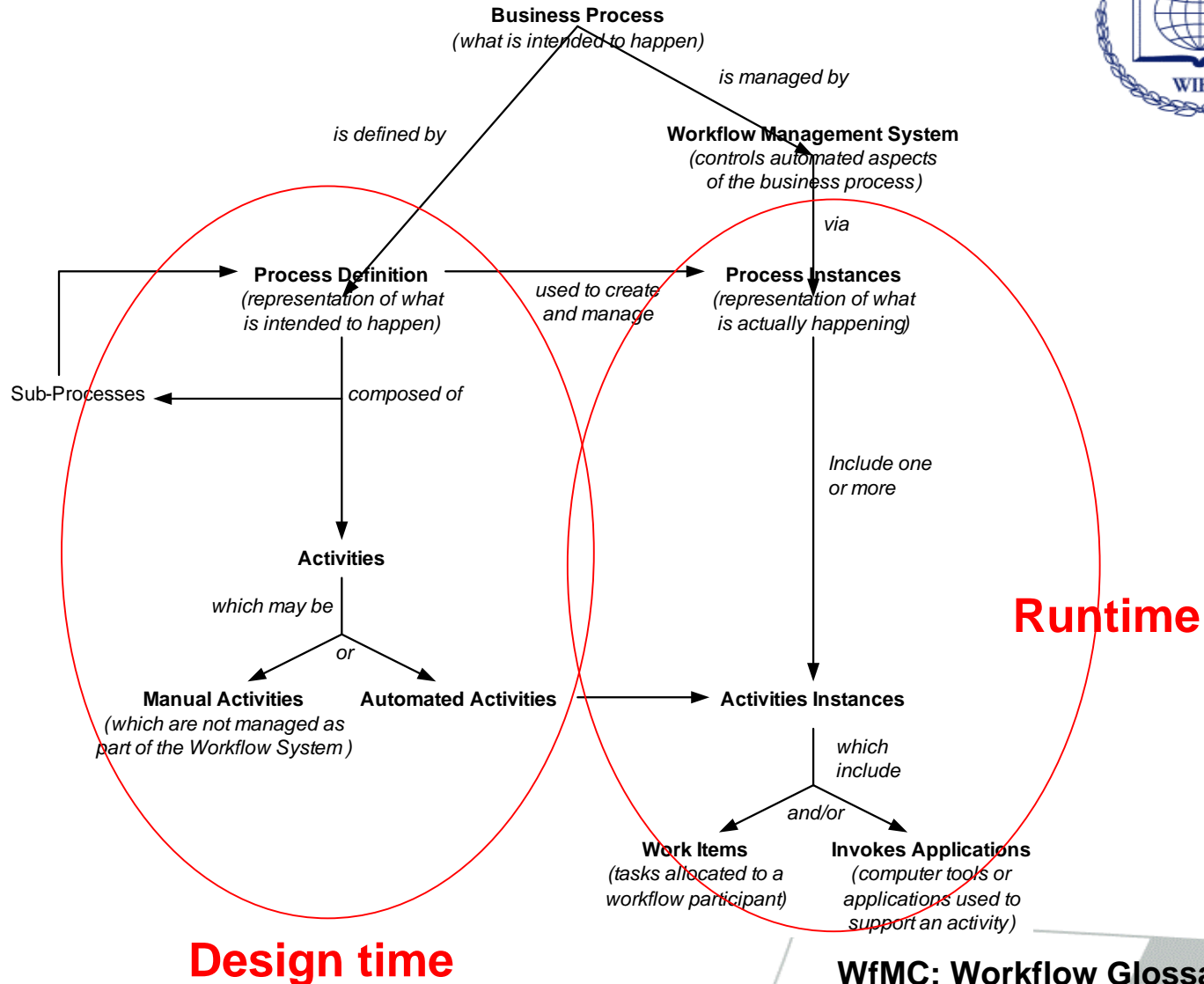
BPEL4WS

Petri Nets

XPDL

Process
Implementation

Workflow Terminology and Glossary





Business Process Design

Modelling Processes with Event-Driven Process Chains

Process Design and Implementation



Event-Driven Process
Chains (EPC)

Process
Design

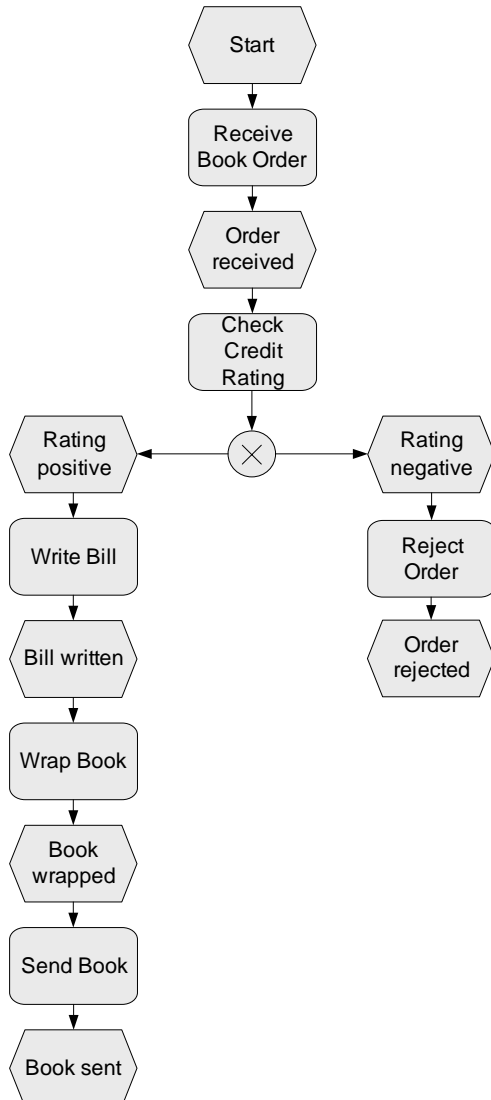
BPEL4WS

Petri Nets

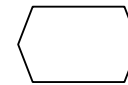
XPDL

Process
Implementation

Event-Driven Process Chains (EPC)



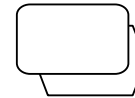
EPC Symbols



Event



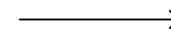
Function



Process Interface



Connectors



Control Flow Arc

Purpose of EPCs

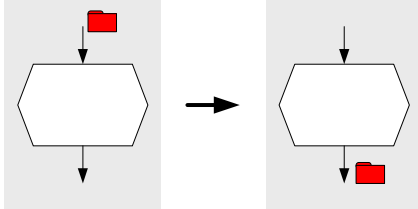
- **Documentation of business processes**
- **SAP implementation projects**
- **Business process re-engineering**



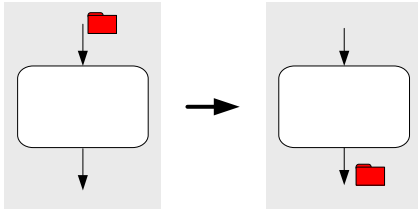
EPC Semantics: Transition Relation



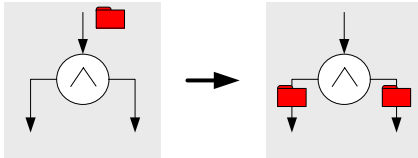
Events



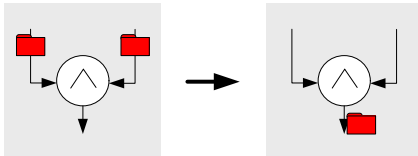
Functions



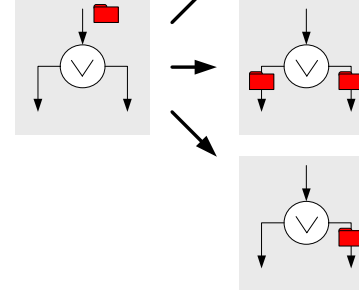
AND Split



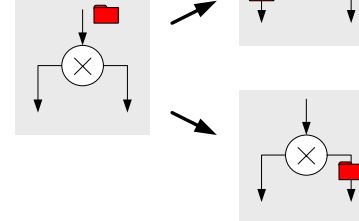
AND Join



OR Split

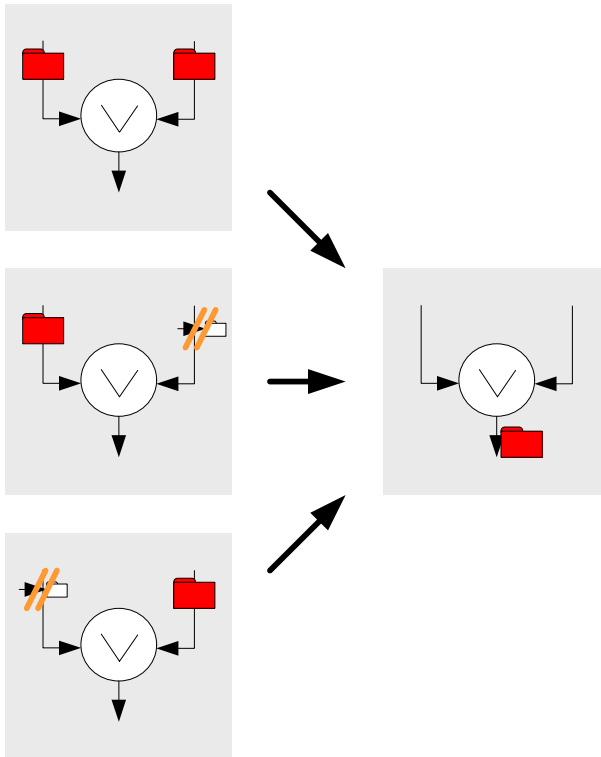


XOR Split

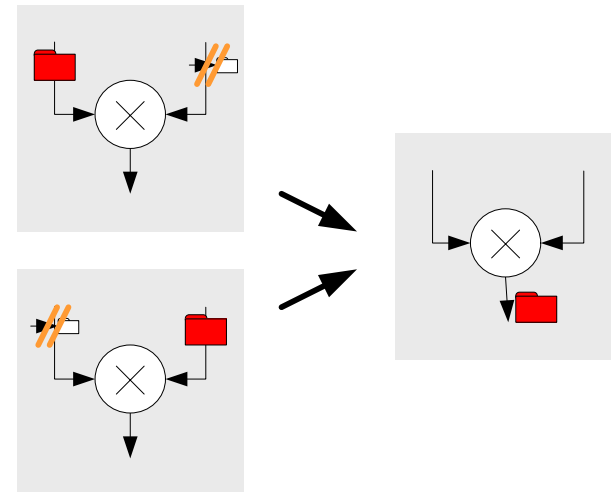


EPC Semantics: Transition Relation II

OR Join

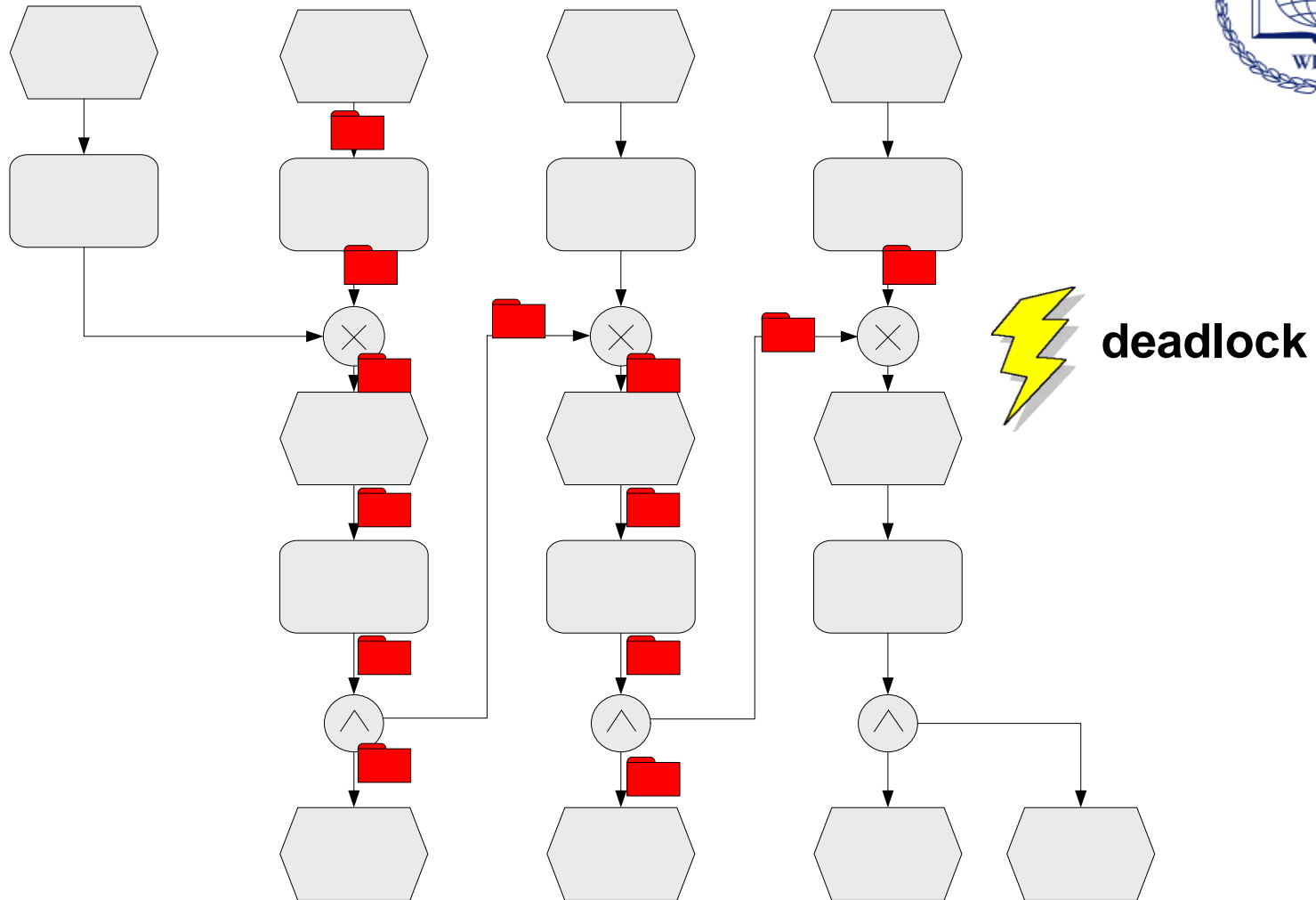


XOR Join

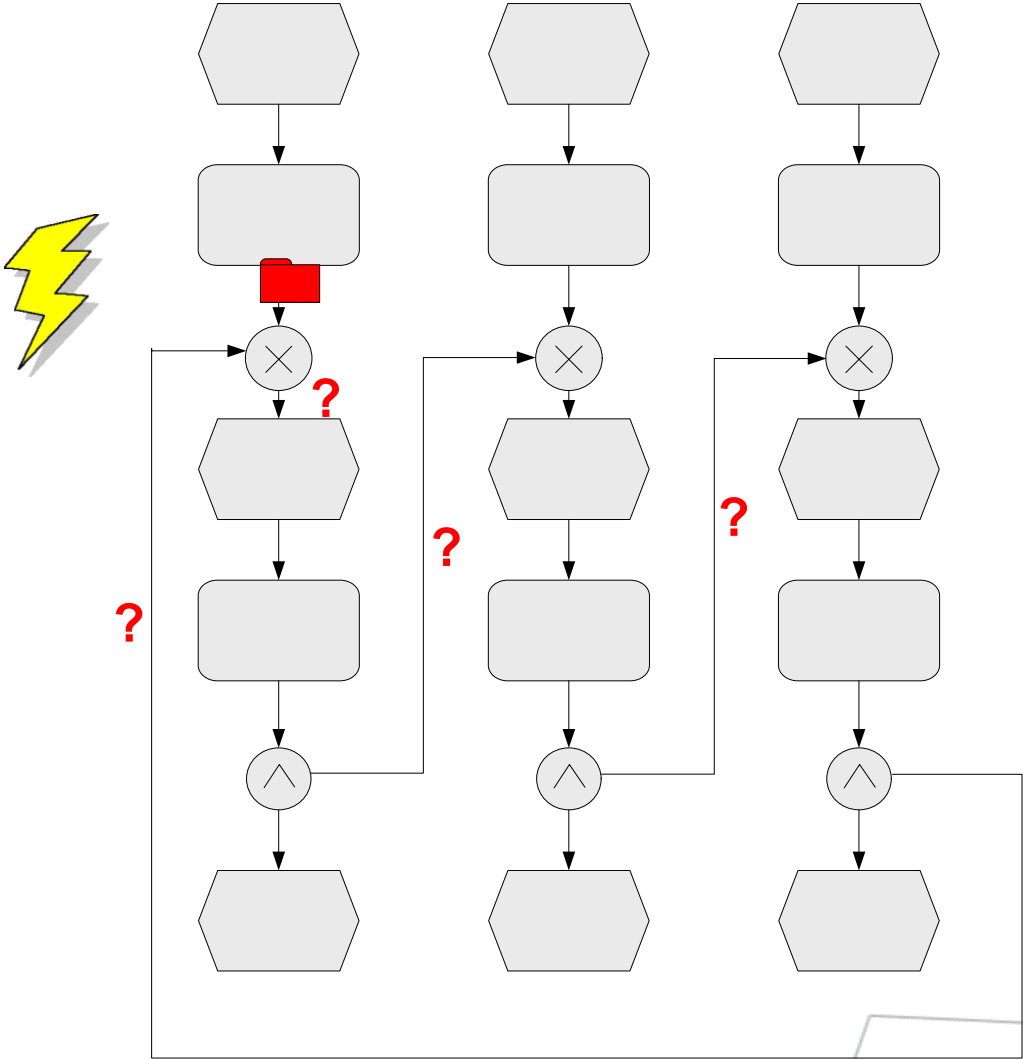


Non-local semantics

Non-local semantics



Vicious Circle



Conclusions



- **Circles may lead to ambiguous situations**
- **It is a good choice to avoid circles and loops (if possible)**
- **Many Workflow Systems do not allow circles**
- **For more on EPCs, see www.epk-community.de**



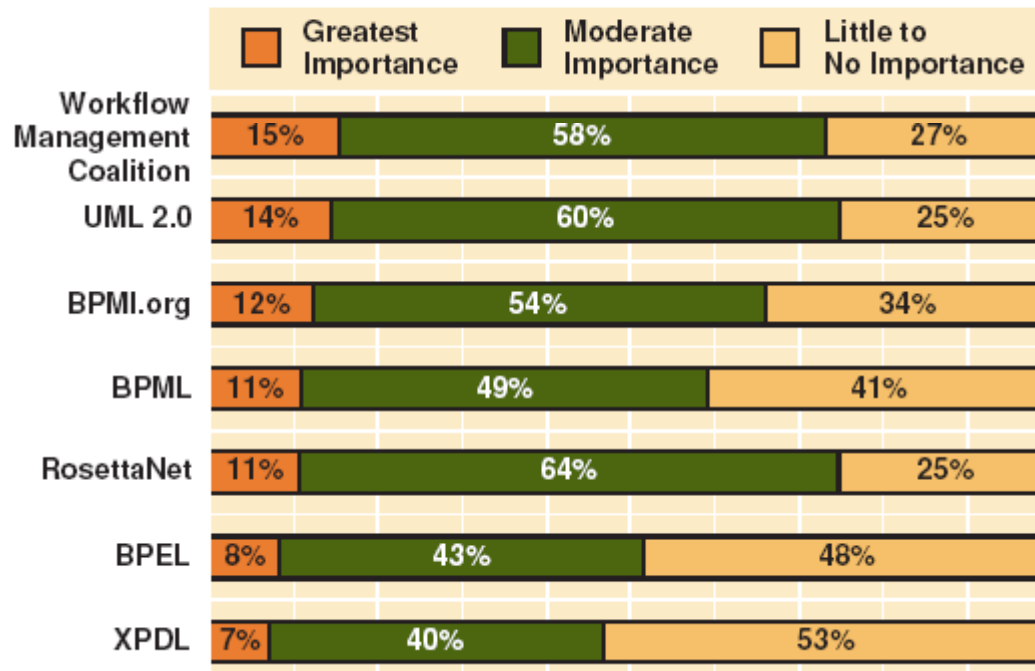
Business Process Implementation

Standards and Languages for Modelling Workflows

Various Standardization Efforts



Recognition of Existing BPM Standard Initiatives



©2003 Delphi Group

Standardization Bodies



- **OMG:** Object Management Group
- **WfMC:** Workflow Management Coalition
- **BPMI:** Business Process Management Initiative
- **OASIS:** Organization for the Advancement of Structured Information Standards
(+ UN/CEFACT)
- **W3C:** World Wide Web Consortium

- **academic initiatives**

BPM Specifications Overview



- **BPDM:** Business Process Definition Metamodel by OMG
- **BPEL4WS:** Business Process Execution Language for Web Services by OASIS
- **BPML:** Business Process Modeling Language by BPMI
- **BPMN:** Business Process Modeling Notation by BPMI
- **BPSS:** Business Process Specification Schema by OASIS + UN/CEFACT
- **EPML:** EPC Markup Language by academia
- **OWL-S** by academia
- **PNML:** Petri Net Markup Language by academia
- **UML ActD** by OMG
- **WS-CDL:** WS-Choreography Description Language by W3C
- **WSCI:** WS Choreography Interface by W3C
- **WSCL:** WS Choreography Language by Hewlett-Packard
- **WSFL:** WS Flow Language by IBM
- **XLANG** by Microsoft
- **XPDL:** XML Process Definition Language by WfMC

Comparison of BPM Specifications



	BPDM	BPEL4WS	BPML	BPMN	BPSS	EPML	OWL-S	PNML	UML Act.D.	WS-CDL	WSCI	WSCL	WSFL	XLANG	XPDL
Task I/O	?	+	+	+	+	-	+	-	+	+	+	+	+	+	+
Task Address	?	+	+	+	-	-	+	-	-	+	+	+	+	+	+
Quality Attributes	?	-	-	-	+	-	+	-	-	-	-	-	+	-	-
Protocol	?	+	-	+	-	-	+	-	-	+	+	+	+	+	-
Control Flow	?	+	+	+	+	+	+	+	+	+	+	+	+	+	+
Data Handling	?	+	+	+	-	-	-	-	+	+	-	-	+	-	+
Instance Identity	?	+	+	-	-	-	-	-	-	-	+	-	+	+	-
Roles	?	+	+	+	+	-	+	-	+	+	+	-	+	+	+
Events	?	+	+	+	-	+	-	-	-	-	-	-	+	+	+
Exceptions	?	+	+	+	+	-	-	-	+	+	+	-	+	+	+
Transactions	?	+	+	+	+	-	-	-	-	+	+	-	-	+	-
Graphic Position	?	-	-	+	-	+	-	+	+	-	-	-	-	-	-
Statistical Data	?	-	-	-	-	-	-	-	-	-	-	-	-	-	+

Process Design and Implementation



Event-Driven Process
Chains (EPC)

Process
Design

BPEL4WS

Petri Nets

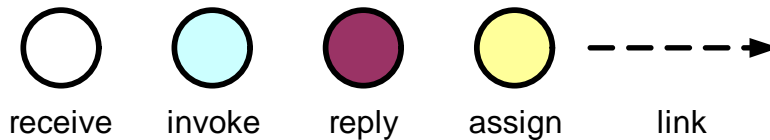
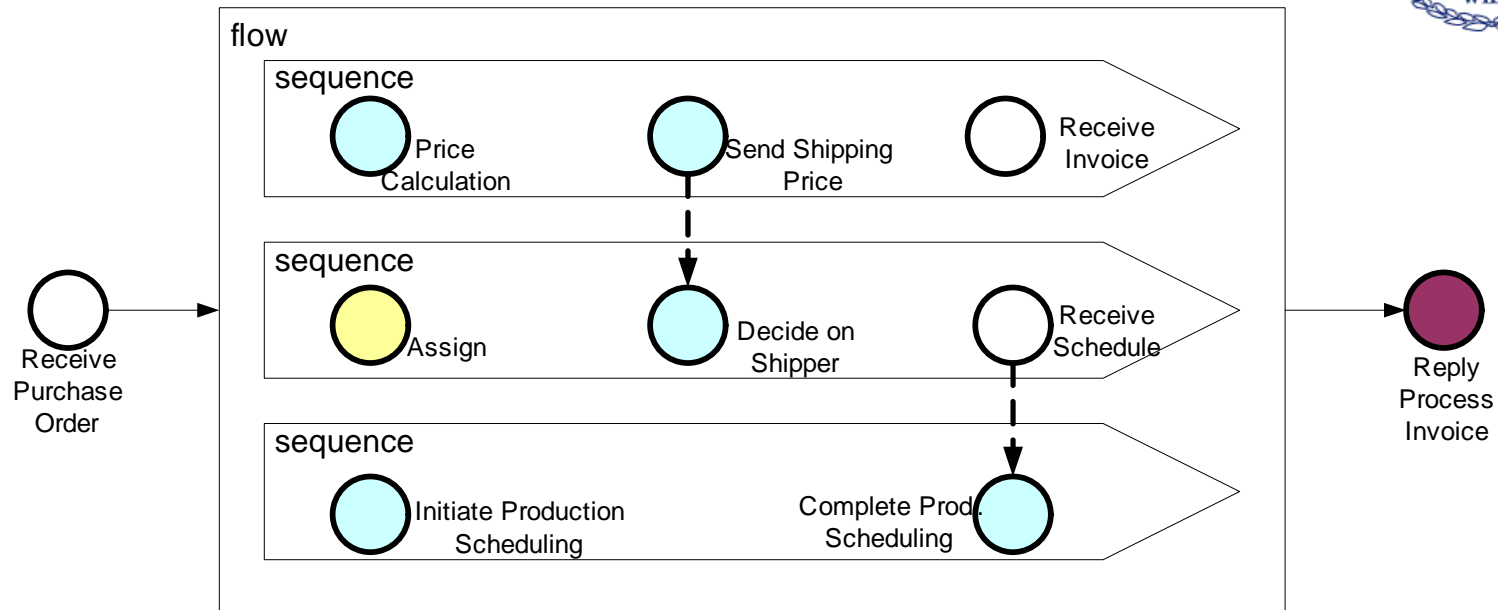
XPDL

Process
Implementation

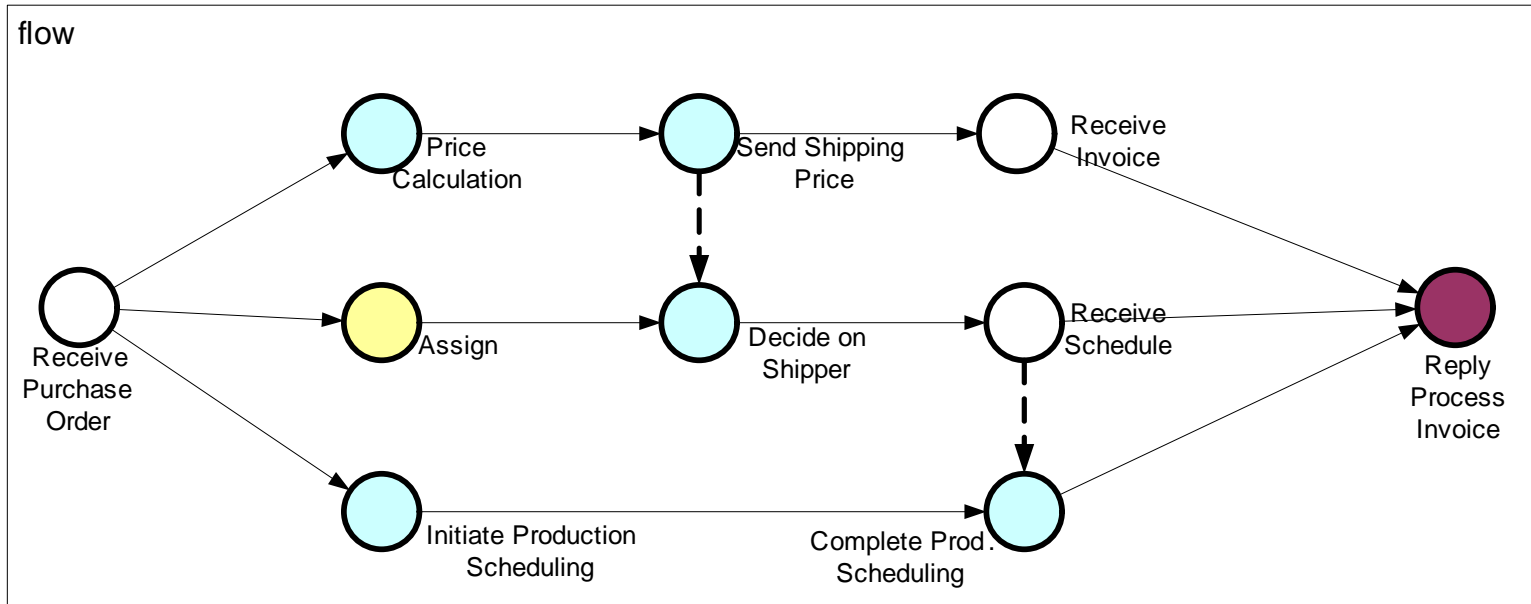


- **Partner Links:**
 - Bilateral conversation (my Role, Partner Role)
 - Interface requirements of partners
- **Data and messages:**
 - Variables
 - Properties
 - Correlation Set
- **Activities:**
 - Basic Activities
 - Structured Activities (control flow)

BPEL4WS Example



BPEL4WS Example





- **Deadlock Freedom**
there is no situation where a process instance has not yet reached a correct final state, but no activity can be finished anymore
- **Termination**
The flow must terminate exactly once without any residual branch being still under execution
- **Reachability**
Each activity should be reachable – starting from a correct initial state there must be a valid sequence of activity executions and outputs that will lead to activation of X.

Conclusion



- **These control flow problems can only be analyzed for a subset of all BPEL**
- **BPEL4WS processes without links grant good control flow**
- **Restrictions on links allows analysis**

- **For details see Reichert, Rinderle, Dadam: On the modeling of correct service flows with BPEL4WS, 2004.**

Process Design and Implementation



Event-Driven Process
Chains (EPC)

Process
Design

BPEL4WS

Petri Nets

XPDL

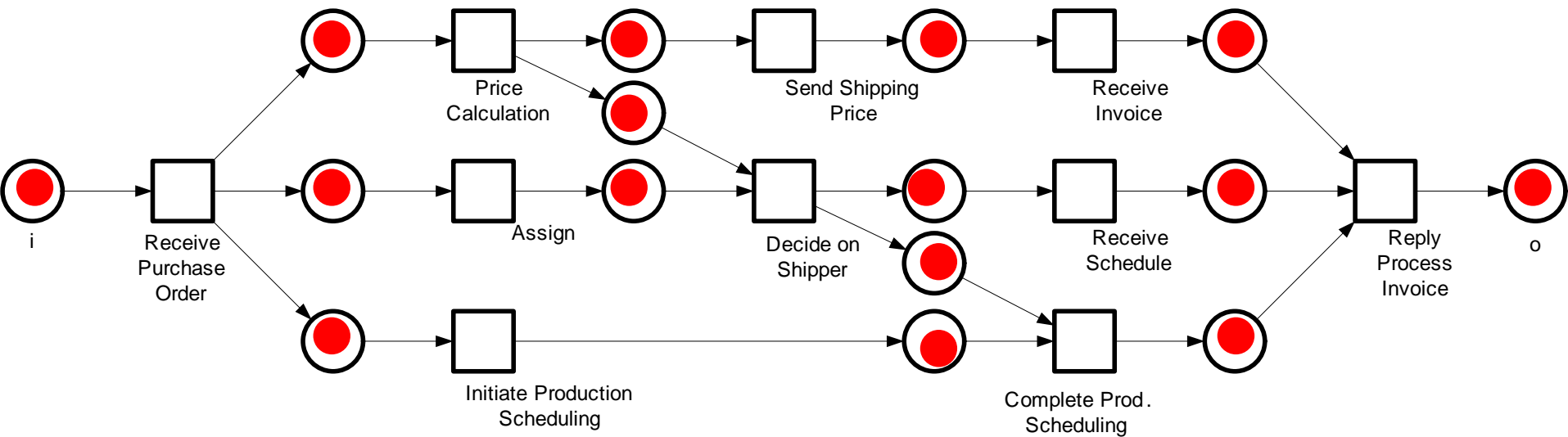
Process
Implementation

Petri Nets



- **Places** to capture states of a process
- **Transitions** to capture state changes
- **Arcs** to capture control flow
- **Tokens** to capture current state

Petri Net Example





Workflow Nets are special Petri Nets

Soundness implies:

- For every state M reachable from state I , there exists a firing sequence leading from state M to state o .
- State o is the only state reachable from state I with at least one token in place o .
- There are no dead transitions.
- For details see e.g. van der Aalst, 2000

Conclusion



- **Rich mathematical foundations permits in-depth analysis**
- **Petri Nets are popular in academia**
- **There are several workflow engines that use Petri Nets**

Process Design and Implementation



Event-Driven Process
Chains (EPC)

Process
Design

BPEL4WS

Petri Nets

XPDL

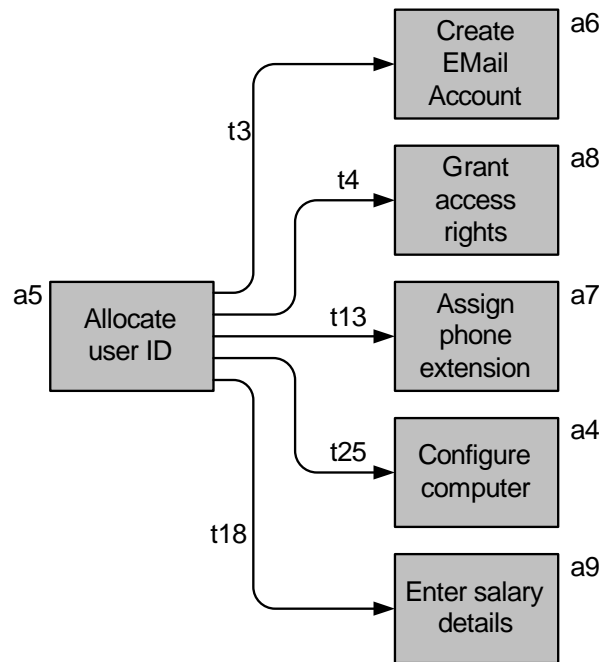
Process
Implementation

XPDL Concepts



- **Standard proposed by Workflow Management Coalition**
- **Used in open source workflow engine OBE**
- **Workflow defined by activities and transitions**
- **Participants, applications, and data fields involved**

XPDL Example



```

<WorkflowProcesses>
  <WorkflowProcess AccessLevel="PUBLIC"
    Id="new-employee" Name="New Employee">
    <Activities>
      <Activity Id="a5" Name="Allocate user ID">
        <Implementation>
          <Tool Id="createNetworkUser" Type="PROCEDURE">
            <ActualParameters>
              ...
            </ActualParameters>
          </Tool>
        </Implementation>
        <Performer>IT</Performer>
        <StartMode><Automatic/></StartMode>
        <FinishMode><Automatic/></FinishMode>
        <TransitionRestrictions>
          <TransitionRestriction
            <Split Type="AND">
              <TransitionRef
                <TransitionRef Id="t3"/>
                <TransitionRef Id="t4"/>
                <TransitionRef Id="t13"/>
                <TransitionRef Id="t25"/>
                <TransitionRef Id="t18"/>
              </TransitionRef>
            </Split>
          </TransitionRestriction>
        </TransitionRestrictions>
      </Activity>
      ...
    </Activities>
    <Transitions>
      <Transition From="a5" Id="t3" Name="Transition" To="a6"/>
      ...
    </Transitions>
  </WorkflowProcess>
</WorkflowProcesses>
  
```

XPDL Schema Problems



- **Missing Default values**
 - **Undefined semantics**
 - **Schema errors and ambiguities**
 - **Schema omissions and inconsistencies**
-
- **For details, see Mendling, zur Muehlen, Price: Standards for Workflow Definition and Execution, in: Process-Aware Information Systems, 2005.**

Conclusion



- **XPDL in its current version needs rework**
- **Formal analysis is difficult, because of transition conditions**
- **Yet, some open source workflow engines use XPDL**

Modelling Languages for Business Processes and Workflow

Event-Driven Process
Chains (EPC)

Process
Design

BPEL4WS

Petri Nets

XPDL

Process
Implementation

Workflow Patterns



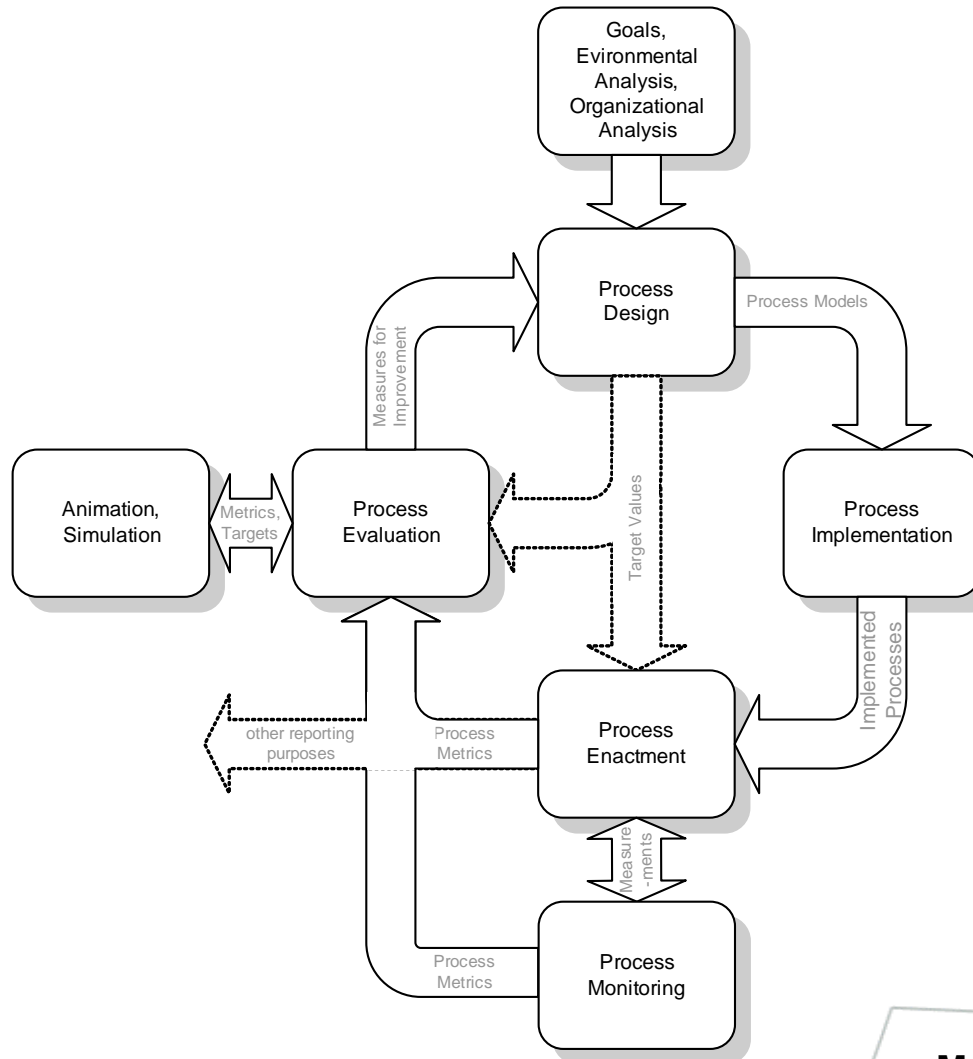
- **Identification of control flow concepts**
- **List of 20 Workflow Patterns**
- **Generalization from EPCs, Petri Nets, etc.**
- **For details see van der Aalst et al., 2003**



Conclusion

Process Design and Implementation

Business Process Lifecycle



Overall Conclusion



- **Heterogeneity is still a problem**
- **Standard proposals are often vendor driven**
- **Analysis of semantics is an important issue**
- **Workflow Patterns will hopefully be reflected in future standards**



Thank you for your attention!

Jan Mendling

Dept. Of IS and New Media
WU Wien, Austria
{jan.mendling}@wu-wien.ac.at