32. Web Services, Workflows and Service-Oriented Architectures

Lecturer: Dr. Sebastian Götz

Prof. Dr. Uwe Aßmann

Technische Universität Dresden

Institut für Software- und Multimediatechnik

http://st.inf.tu-dresden.de

7. Juni 2018

1) Web Services as a specific form of service-oriented architectures
2) WSDL
3) BPEL
4) BPMN
5) Evaluation

Obligatory Reading

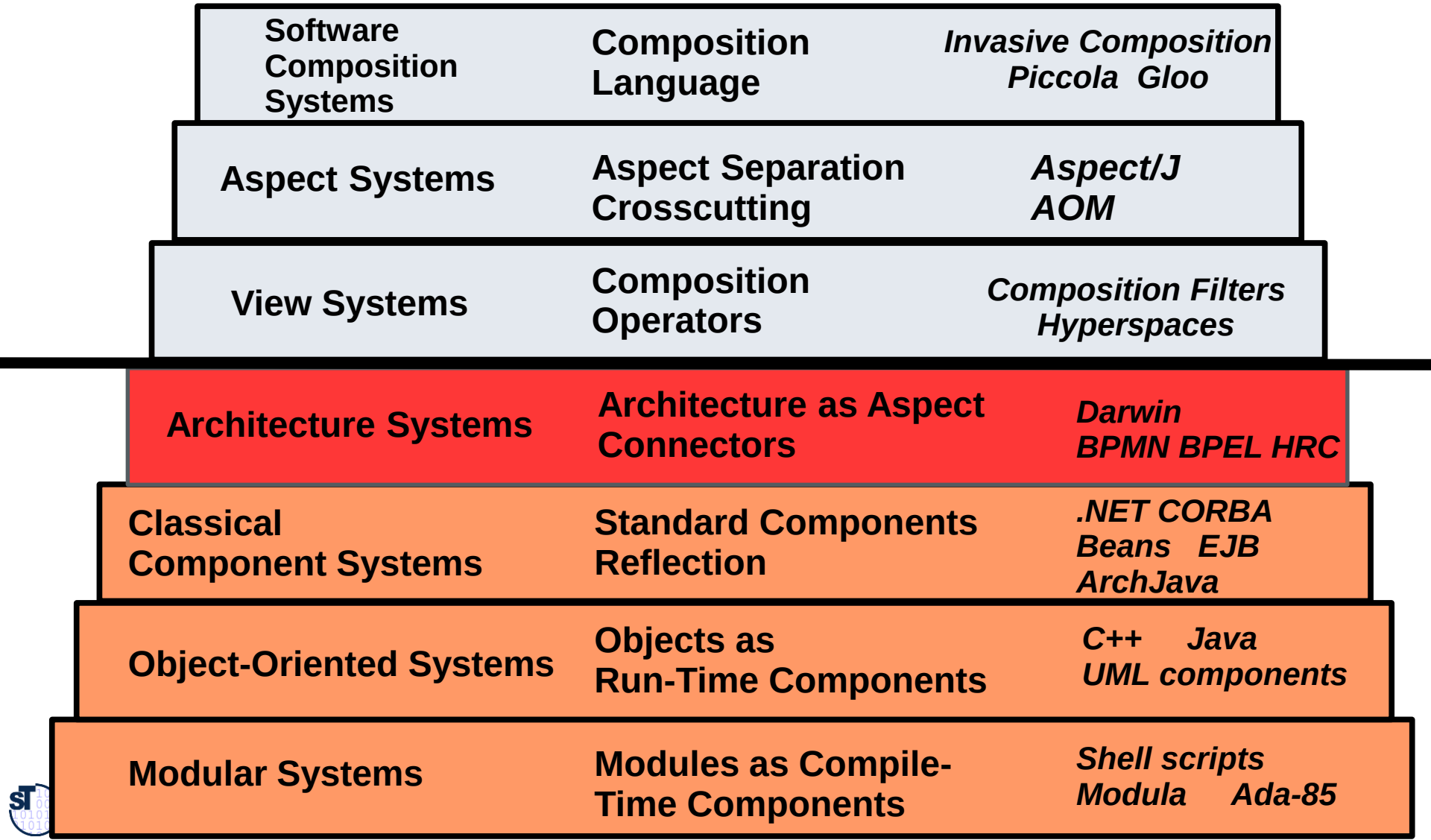
- ISC, Chapter 2.4
- Lohmann, Niels, Verbeek, Eric, Dijkman, Remco. Petri Net Transformations for Business Processes – A Survey. In : Transactions on Petri Nets and Other Models of Concurrency II, Editor: Jensen, Kurt, van der Aalst, Wil, Lecture Notes in Computer Science 5460, 2009, Springer Berlin / Heidelberg http://www.springerlink.com/content/n7464131r6751453/
- W.M.P. Van der Aalst. Don't go with the flow: Web services composition standards exposed. IEEE Intelligent Systems, Jan/Feb 2003.

  http://www.martinfowler.workflowpatterns.com/documentation/documents/ieeewebflow.pdf
- P. Wohed, W.M.P. Van der Aalst, M. Dumas, A. ter Hofstede. Analysis of Web Service Composition Languages: The Case of BPEL.
- http://www.bpmn.org/ BPMN home page at OMG

Other Literature

- Matthias Weske. Business Process Management – Concepts, Languages, Architectures. Springer. 2007
- YAWL http://www.yawlfoundation.org/
- H. P. Alesso, C. F. Smith. Developing Semantic Web Services. A K Peters Ltd, Natick, Massachusetts, 2004.
- BPMN 2.0 language specification
  - http://www.omg.org/spec/BPMN/2.0/
- Scheer, A.-W. ARIS - Business Process Frameworks. Springer, Berlin, 1998, ISBN 3-540-64439-3
- Michael C. Jaeger. Modelling of Service Compositions: Relations to Business Process and Workflow Modelling. ICSOC 2007, LNCS 4652.

The Ladder of Composition Systems

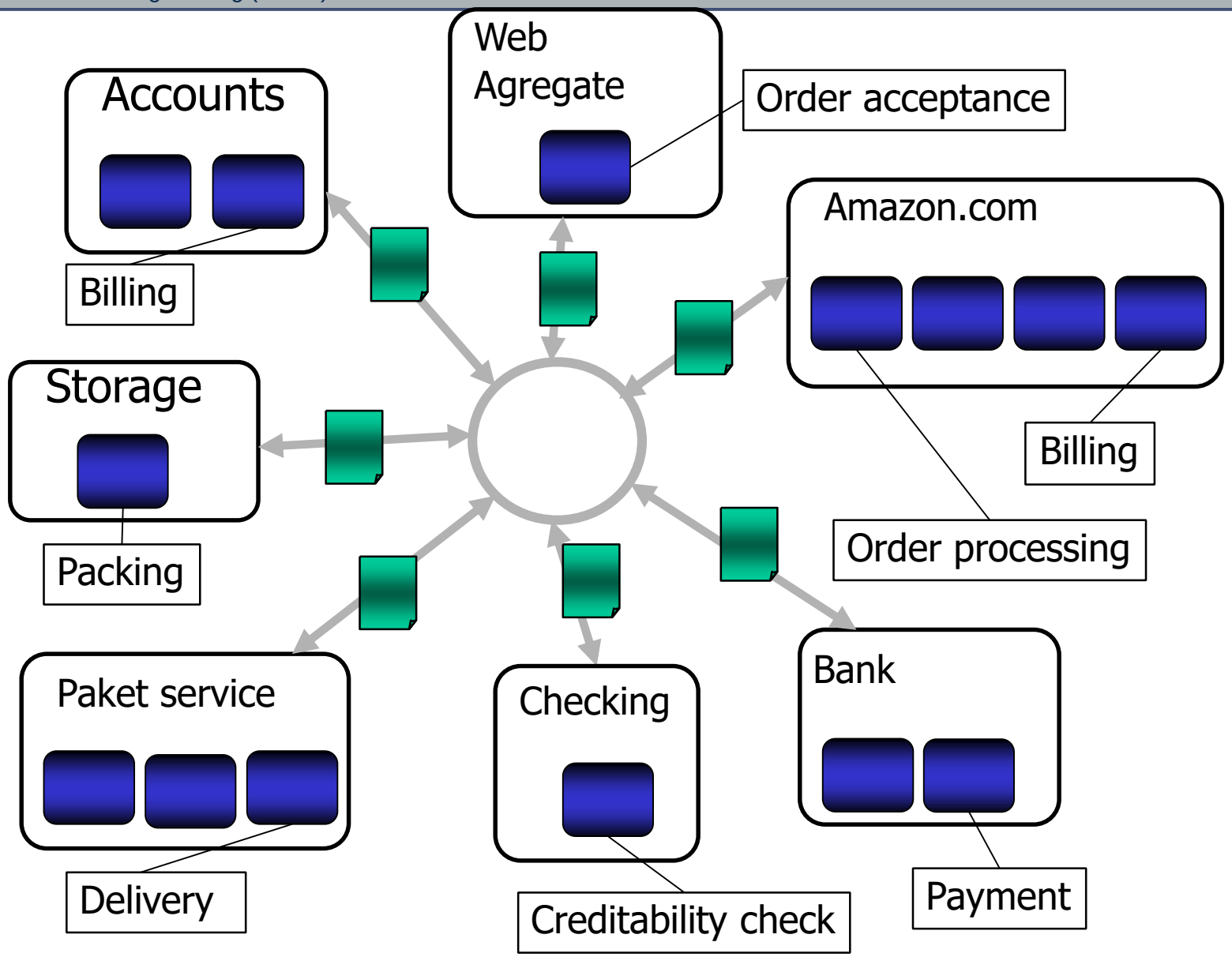


32.1 Web Services as Architecture Systems

Web Services and Architecture Systems

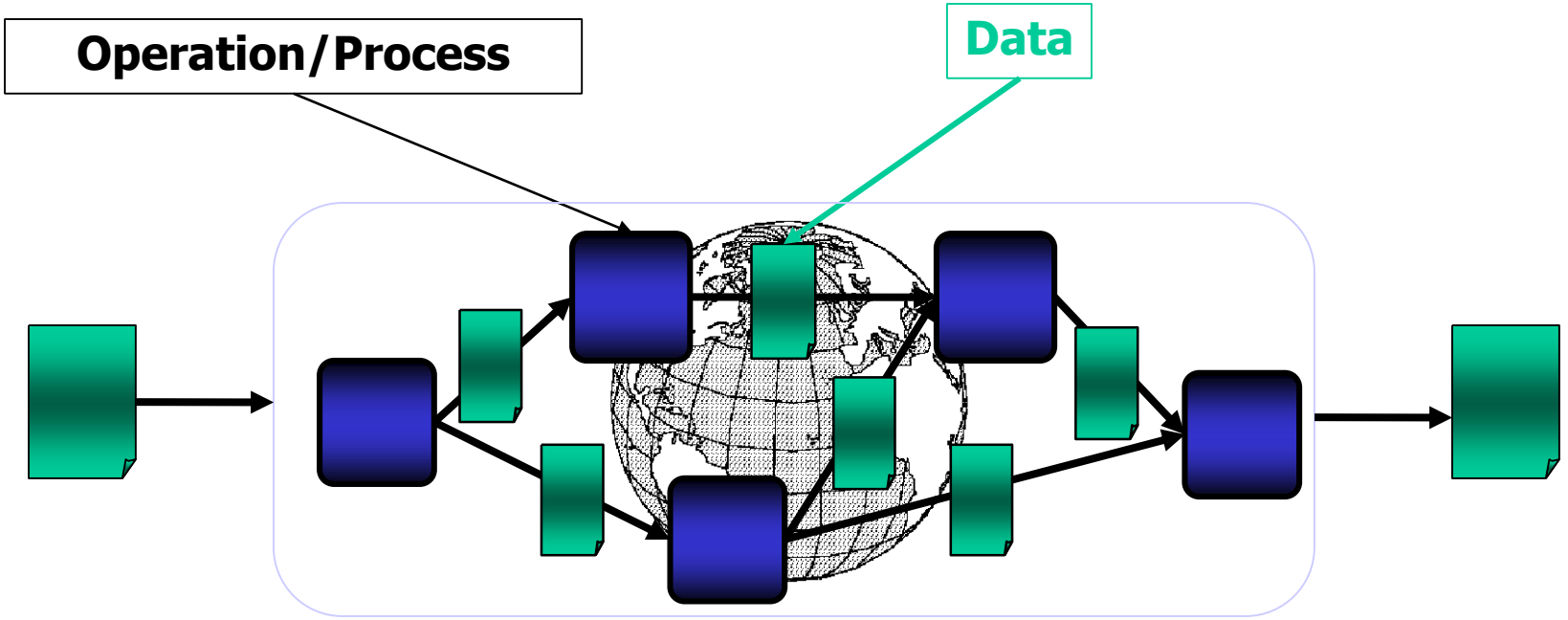
- Architecture systems may have different forms of architectural languages:
  - Topology-based (Unicon, ACME, Darwin)
  - Coordination schemes (CoSy)
  - Imperative scripts (Darwin)
- Web Service Systems and Languages (WSS) are a form of architectural system
  - They separate programming-in-the-small from programming-in-the-large (2-level programming)
    - Components encapsulate the service knowledge
    - The architectural level (orchestration, aggregation, composition) treats the big picture
- However, WSS have an imperative architectural language
  - They are based on XML standards (SOAP, WSDL, BPEL)

Web Services are Black-Box Components



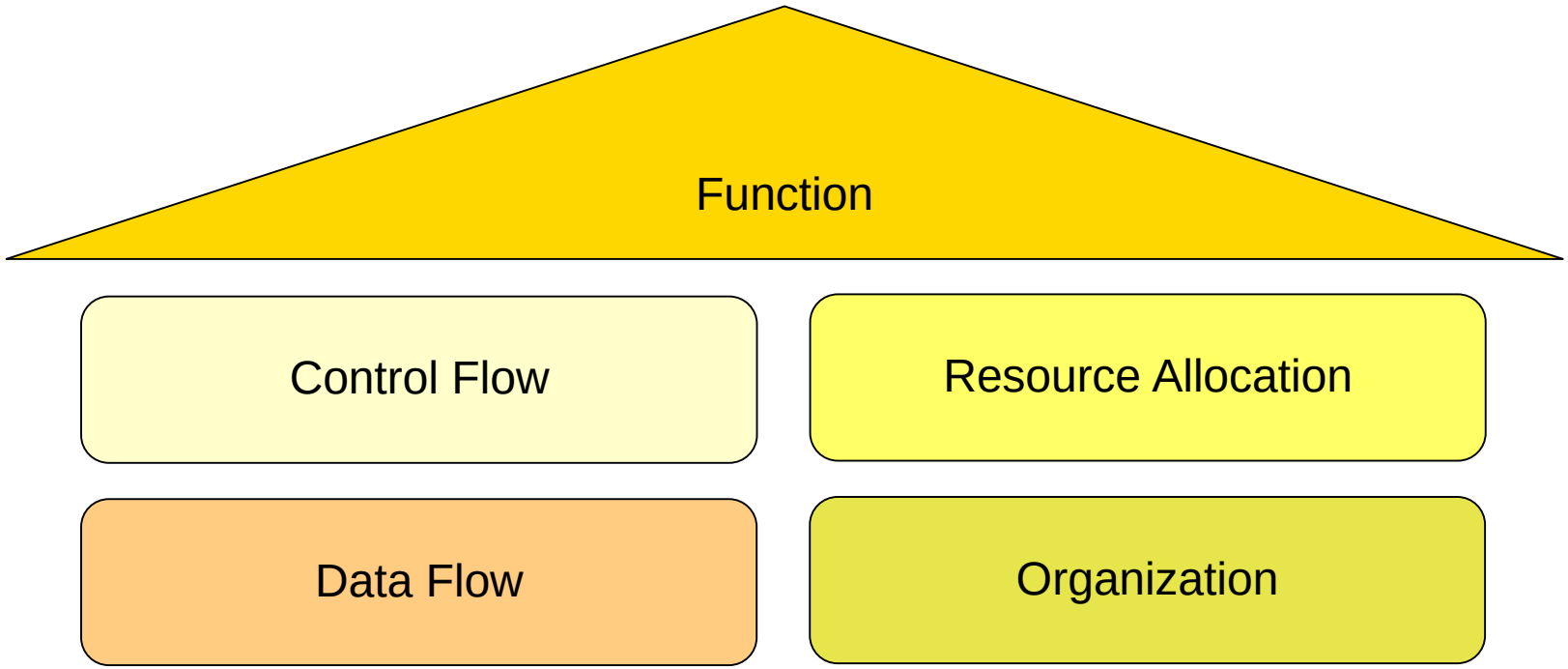
Web Service Architectures are Described by Workflows

- Web service architectures are the first step to service-oriented architectures (SOA), based on traders
  - Services are offered, searched and discovered, downloaded, executed
  - Workflow specifications combine control and data flow
- *Enterprise services* transfers web services to business systems
- *Customer services* serve the end-user



Workflows Languages Have Aspects

- Standard workflow modeling discerns about 5 aspects
  - ex. ARIS house [Scheer's company IDS, now Software AG]



[Scheer]

Workflow Languages

- A **workflow language** specifies control and data flow over a set of operations
  - The workflow is executable with an interpreter, the *workflow engine*
  - A single operation need not be executed automatically, but can be performed by humans (… for people)
  - The workflow runs in parallel
- Workflows are usually compiled to Colored Petri Nets, to Statecharts, or to data-flow diagrams
  - YAWL (van der Aalst, Einhoven)
  - Workflow Nets
- Industrial Examples:
  - Lotus Domino (IBM)
  - Business Process Execution Language (BPEL)
  - ARIS system for SAP, based on EPC (event process chains)
  - Business Process Modeling Notation (BPMN), also in use at SAP

What is a Business Process?
Business Processes are Abstract Workflows

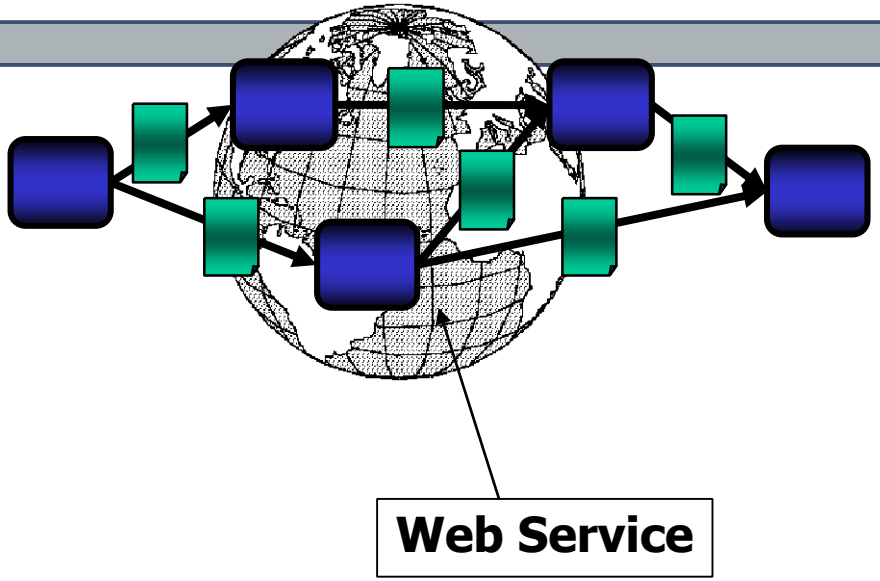
- **Business processes** are *partial* or *abstract* workflows describing processes in enterprises
  - A business process is described on the modeling level, can be abstract, underspecified and need not be executable
  - A business process can be refined iteratively to become executable.

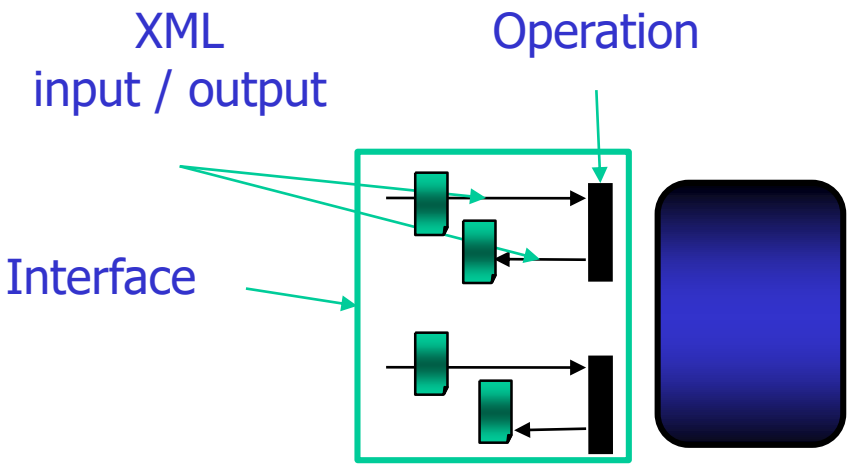
What are Workflow Engines?

- **Workflow engines** are interpreters of workflows
  - They maintain the parallelism in a workflow and synchronize all processes
- Usually, they also support for interactive applications
  - Undo
  - Transactions with rollback and commit
  - Compensation (in case of error)
- They are, for web services and component systems, *composition engines* that execute a composition program, the workflow

32.2 WSDL for the Definition of Interfaces of Web Services

Service Interface

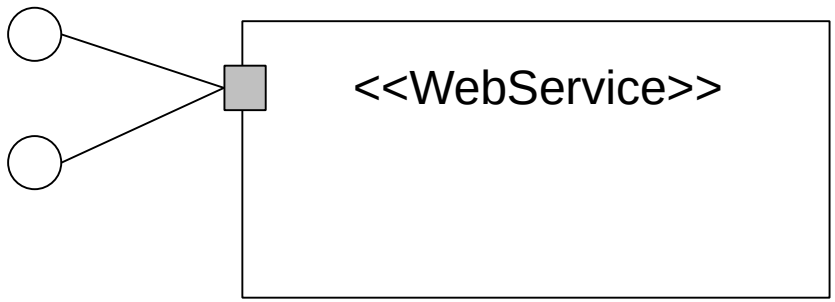




⇨ Web Services Description Language (WSDL) defines a service interface

WSDL Components and Their Interfaces

- A WSDL *Interface* consists of a set of ports
  - Functions with types of parameters and results in XML Schema
  - Plays a similar role as ports of a UML component
- Advantages
  - Component model can be mapped to CORBA, EJB, DCOM, .NET
  - WSDL abstracts from the underlying component model, introducing the component model as a *secret*

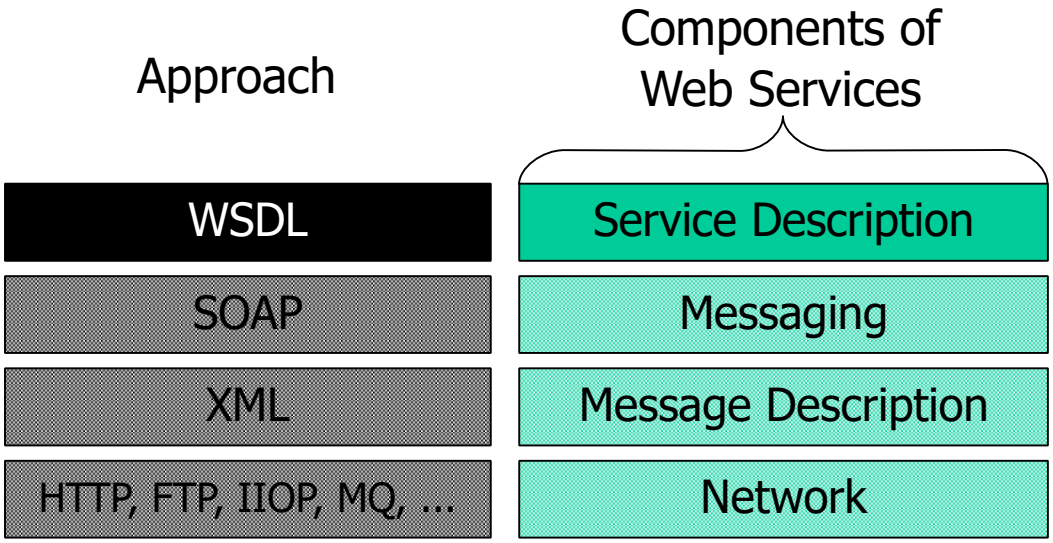


WDSL Specification Structure

- **Types**
  - In XML schema or another typing language
- **Messages**
  - The data that is communicated (not in IDL)
- **Operations**
  - An interface of the service, with input and output, fault parameters
- **Port type**
  - A named set of operations (as in UML components or IDL)
- **Binding**
  - A mapping of the port to underlying component models
- **Service**
  - A set of related ports (as in UML components)

WSDL Service Interface

- WSDL is an Interface Definition Language (IDL)
  - Part of BPEL (see later)
  - W3C Recommendation (standard)



32.3 Business Process Execution and Web Service Workflows with BPEL

- BPEL, a web service composition language

Ingredients of BPEL

- BPEL is an *executable language for workflows*, executable business processes
  - An architectural language for web services
  - Based on workflow languages
  - Mixing control and data flow operators
- BPEL is a composition language composing web services at their ports
  - BPEL uses WSDL for service interface descriptions, as IDL
  - BPEL adds connection types (*partner link types*)

BPEL Made Simple

- BPEL is an activity-diagram like language,
  - with parallelism and transactions
  - with different kinds of join and split operators
  - with ports and connections
  - BPEL can be edited graphically, and has an XML abstract syntax
- BPEL uses XML syntax
  - WSDL definitions to define types, message types, and port types
  - *Partner link types* (connector types) describing typed connections

BPEL Specification Structure

- **Process definition**: Header with namespace declarations
- **Variables**: global variables of the process
- **PartnerLink declarations**: interface declaration
  - with whom is the process connected?
- **Partners**: actual partners of the communication
- **Correlation sets**: Which instance of a process is talking to which other instance?
- **Fault handler**: What happens in the case of an exception?
- **Compensation handler** specifies compensation actions for inconsistencies or damages a fault has provoked
  - Optimistic transactions with compensations
- **Event handler**: what happens in case of a certain event?
- A (structured) **main** operation
  - e.g., sequence or flow

A Simple Pizza Order


```
<!-- Process definition -->
<process name="OrderPizza" suppressJoinFailure="yes"
xmlns="http://schema.xmlsoap.org/ws/2003/03/business-process"
pns="http://www.pizza.org/schema">
<partnerLinks>
   <partnerLink name="PizzaService" partnerLinkType="pns:OrderChannel"
myRole="PizzaOrderer">
</partnerLinks>
<!-- Global Variables -->
<variables>
  <variable name="input" messageType="PizzaOrder"/>
  <variable name="output" messageType="PizzaDelivery"/>
</variables>
<faultHandlers> ... </faultHandlers>
<sequence name="body">
  <invoke name="order" partnerLink="PizzaService" portType="PizzaOrder"
operation="body" variable="output">
  <receive name="acknowledgement" partnerLink="PizzaService" portType="Pizza"
operation="body" variable="input">
</sequence>
</process>
```

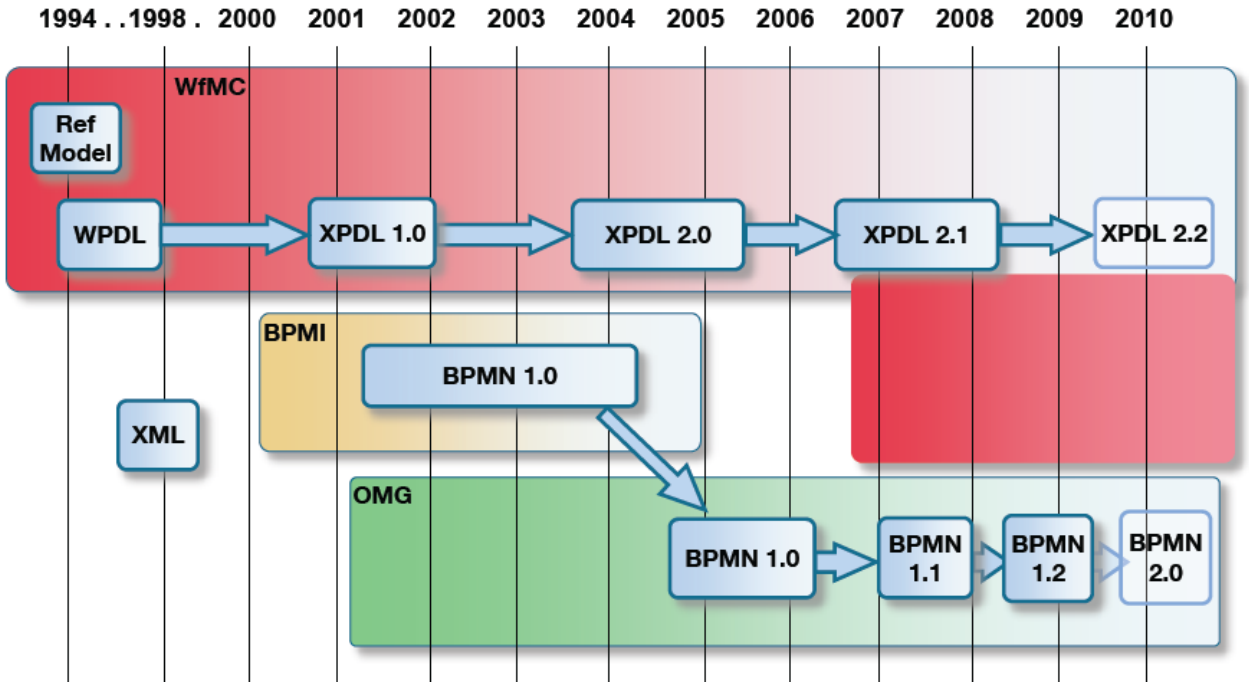
Connector

32.4 Business Process Modeling Notation (BPMN)

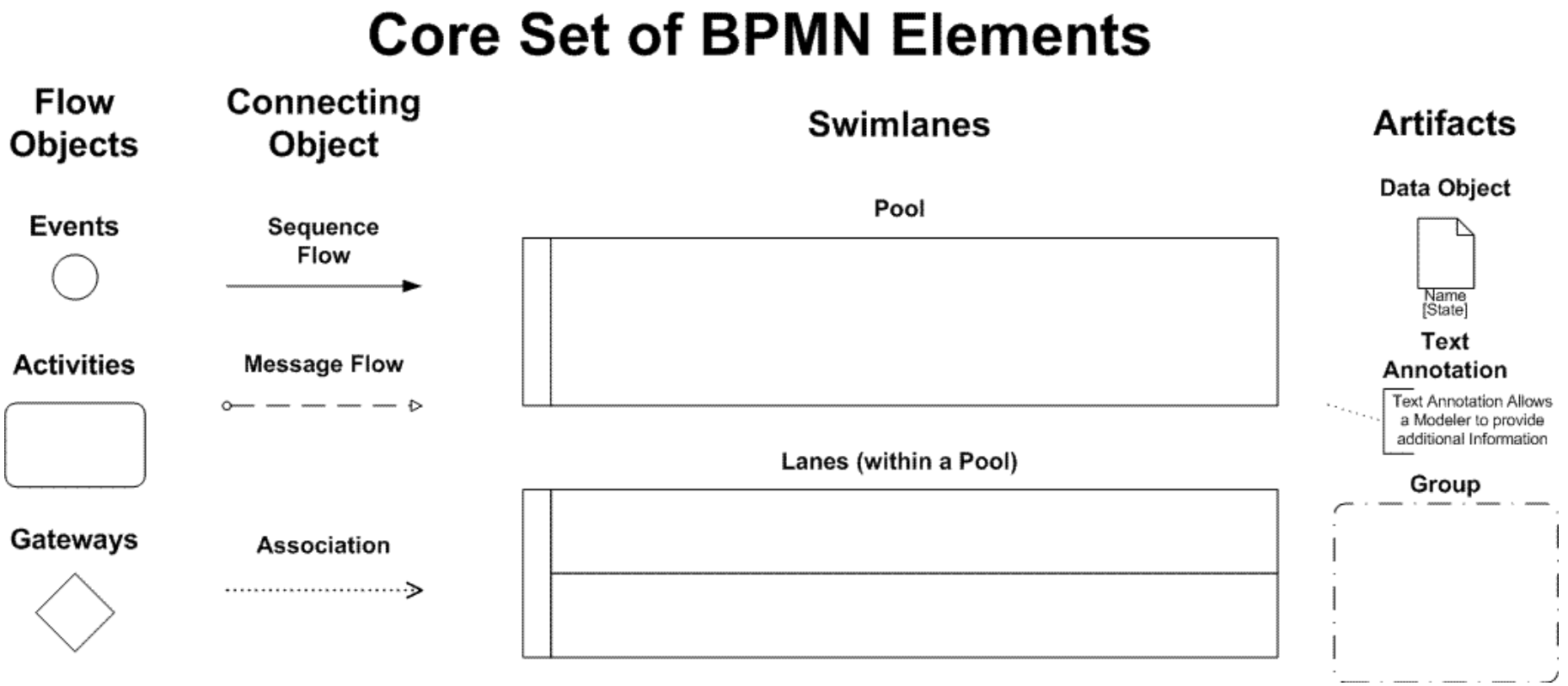
- Another composition language

History

- The Business Process Modelling Notation (BPMN)
- Graphical notation for conceptual business processes
- Covers control, data, authorization, exception
- Standardized by OMG

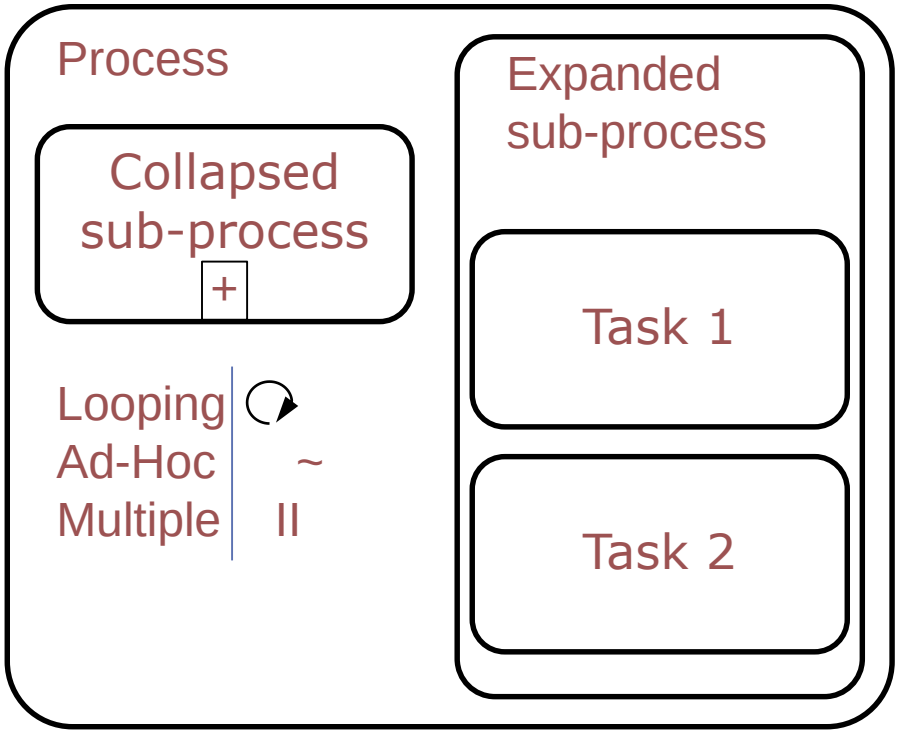


Core Elements



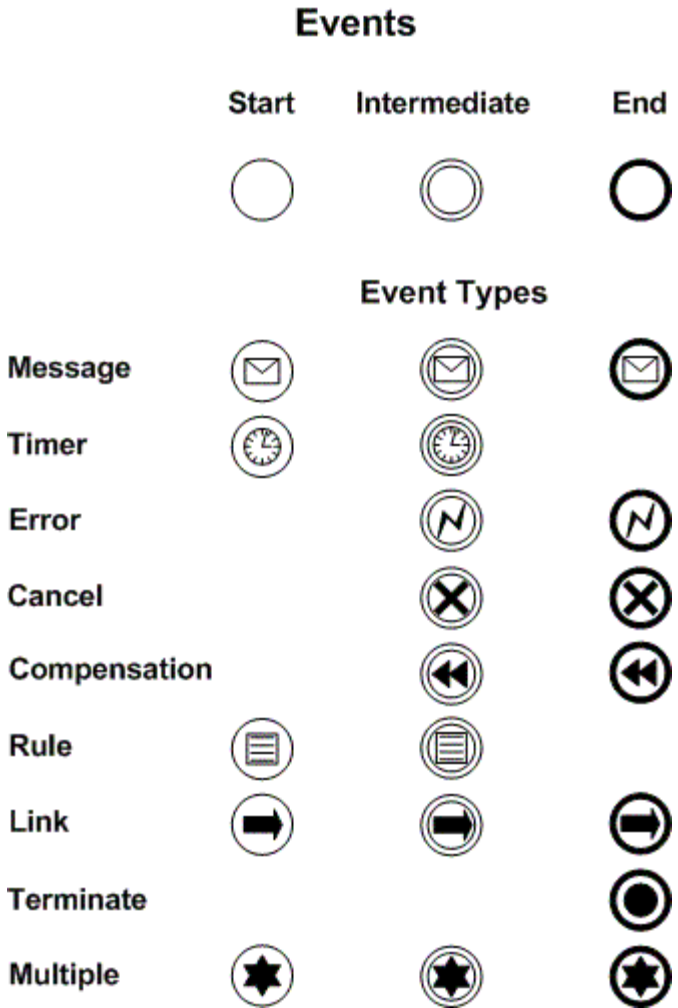
Activities and Processes

- An **activity** in BPMN is a generic type of work that a company performs.
- An activity can be *atomic* (task) or *compound* (process, sub-process).



Events and Activities

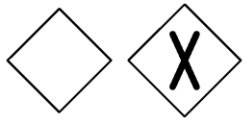
- Events affect the flow of the process and usually have a cause (trigger) or an impact (result): 'Email received', 'Warehouse empty'



Gateways and Connections

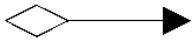
A gateway is used to split or merge multiple process flows. It will determine branching, forking, merging and joining of paths.

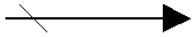
Gateway control types

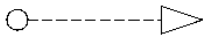
XOR (DATA)		Data based exclusive decision or merging. Both symbols have equal meaning. See also Conditional flow.
XOR (EVENT)		Event based exclusive decision only.
OR		Data based inclusive decision or merging.
COM-PLEX		Complex condition (a combination of basic conditions)
AND		Parallel forking and joining (synchronization).

Graphical connectors

Normal sequence flow 

Conditional sequence flow 

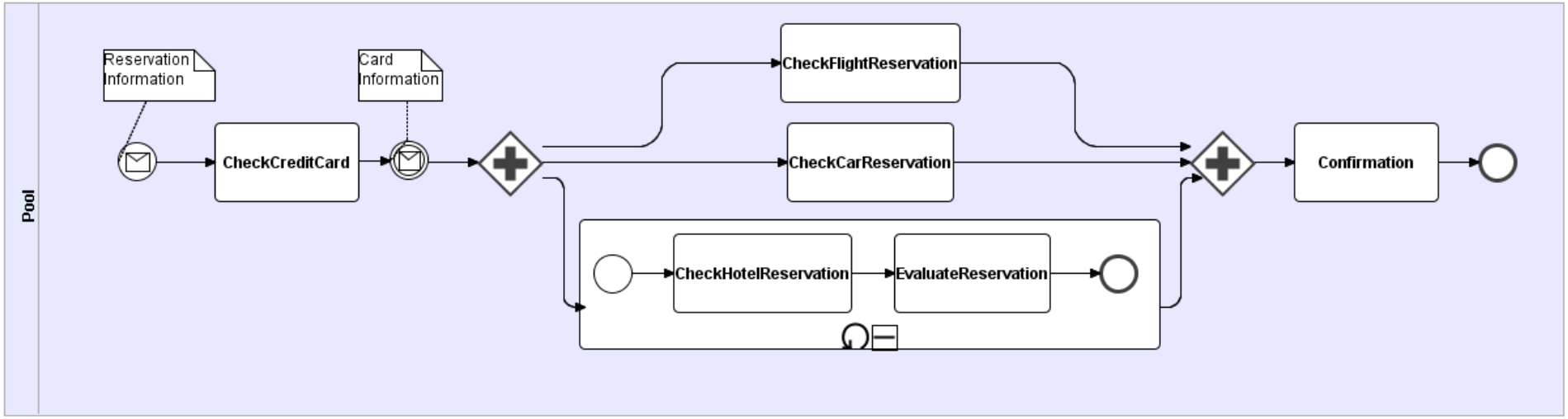
Default sequence flow 

Message flow 

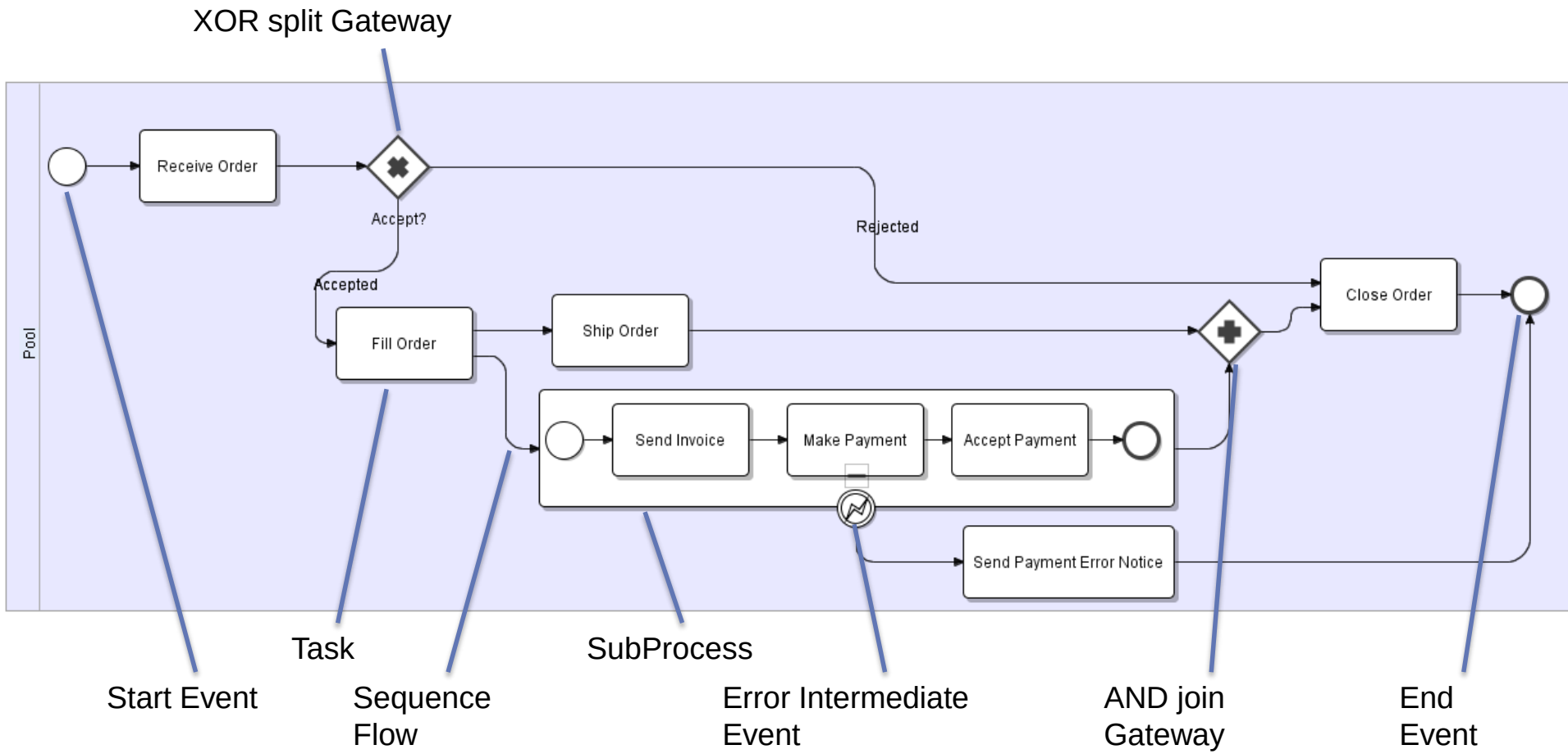
Association 

Example: Travel Process Control Flow

- More refinement leads to business process specifications (with control and data flow)



Example

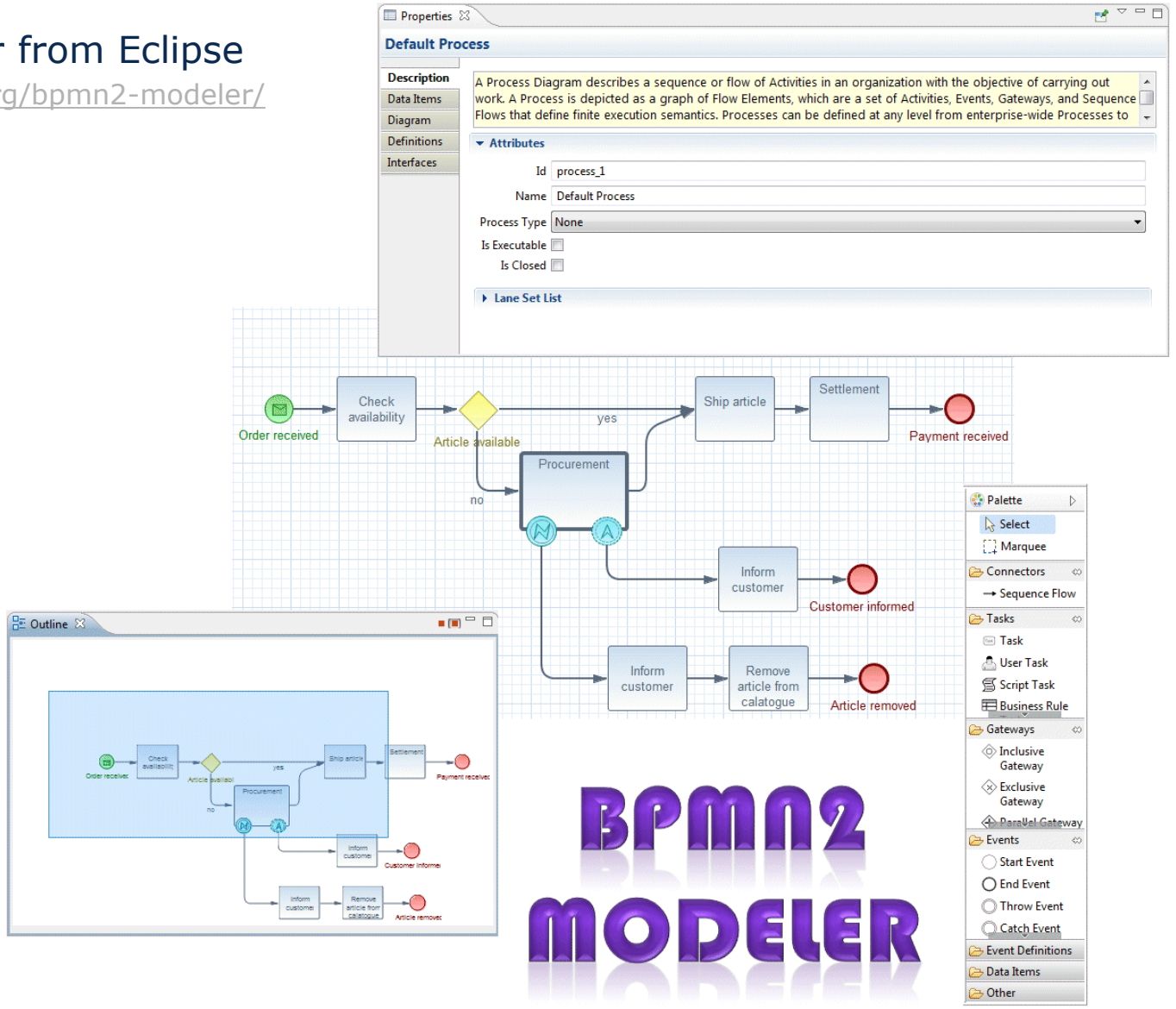

XOR split Gateway
Pool
Receive Order
Accept?
Rejected
Accepted
Fill Order
Ship Order
Close Order
Send Invoice
Make Payment
Accept Payment
Send Payment Error Notice
Task
SubProcess
Start Event
Sequence Flow
Error Intermediate Event
AND join Gateway
End Event

Why BPMN?

- BPMN 2.x
  - Modeling language for business processes: no execution semantics, only a partial mapping to Business Process Execution Language (BPEL)
- BPMN geared towards business analysts:
  - BPMN constructs are simplified
  - UML notation too bloated
  - BPMN is on the platform-independent level, BPEL nearer the platform-specific level

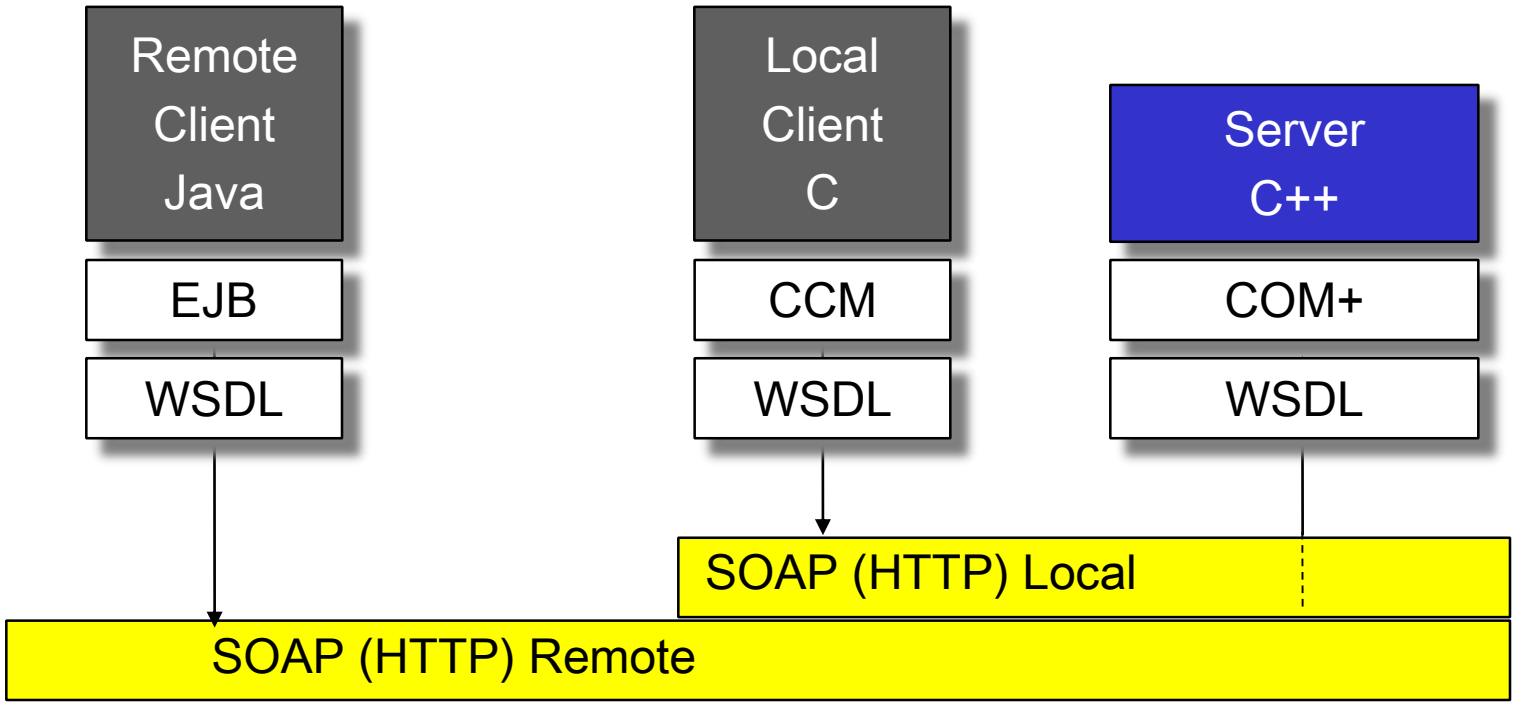
Give BPMN a try

- Free BPMN Editor from Eclipse
- https://www.eclipse.org/bpmn2-modeler/



Web Services – Component Model Transparency

- Language adaptation: XML Schema + WSDL
- Remote transparency: SOAP (+ HTTP)
- Component model transparency (EJB, COM+, CORBA, CCM, Beans, etc...)



32.5 Evaluation of Web Services

- as composition system

Component Model

- Mechanisms for secrets and transparency: very good
  - Location, language, component model transparency
  - Communication protocol transparency
  - Interface specification is flexible with WSDL and USDL
  - Different black-box component models can be hidden under WSDL specifications

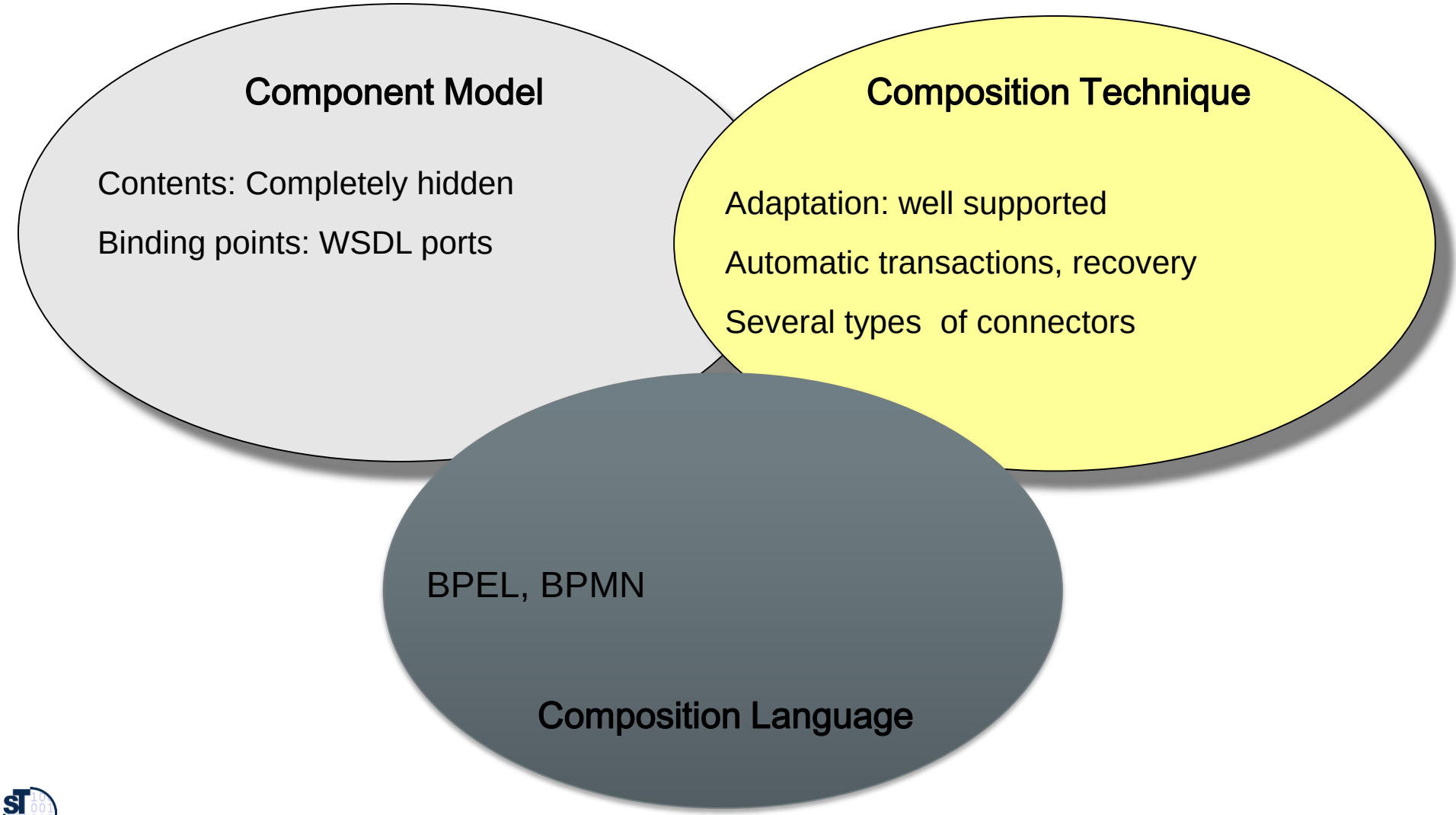
Composition Technique

- Mechanisms for connection
  - Protocol transparency allows for flexible connections
  - WSDL binding is flexible
- Mechanisms for aspect separation
  - Separate modeling from execution (abstract business processes from workflows)
- Scalability: Better
  - Changes of protocol possible
  - Changes of distribution easy
  - Changes of workflow easy

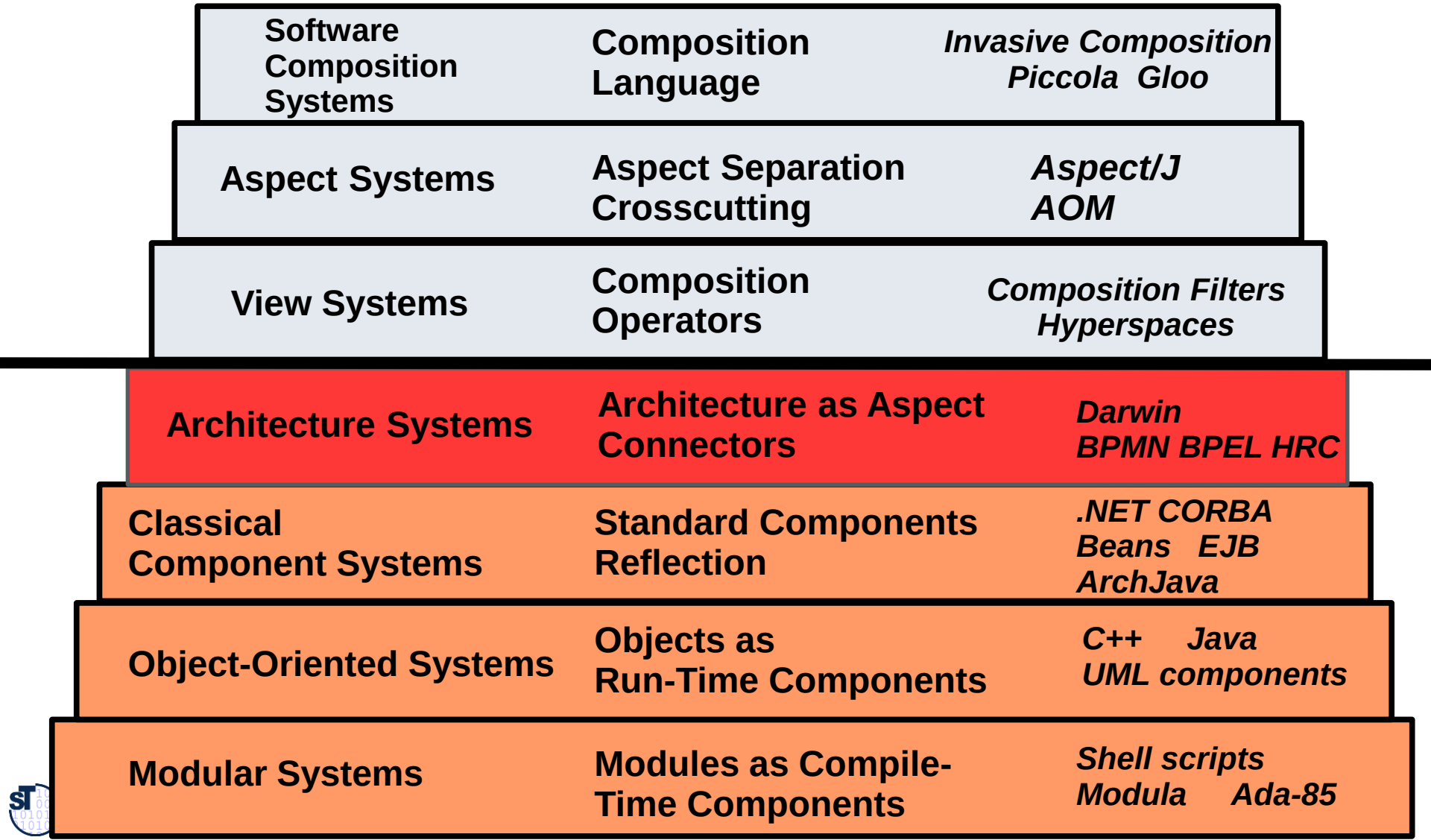
Composition Language

- BPEL, BPMN are flexible composition languages for web services
  - Based on ADL
  - Not yet full exchangeability of connector types
  - But graphic support for workflow specifications
  - Sophisticated control- and data-flow operators (gateways)
  - Parallel execution semantics
  - Abstract (business processes) and executable level (workflows)
- Metacomposition fully supported
  - The generation and composition of a BPEL or BPMN script is easy
    - because it is XML based
  - Development environments generate workflows from other specifications
  - Generic workflow architectures will be possible

Web Services as Composition Systems



So Far: Blackbox Composition Systems



The End

- Many slides inherited from
  - Stig Berild's talk on the Nordic Conference on Web Services, Nov. 2002
  - Prof. Welf Löwe, Web Service Competence Center (WSCC), Växjö Linnaeus University

Some Abbreviations

- ebXML: Electronic Business XML
- UDDI: Universal Description, Discovery and Integration
- OAG: Open Applications Group
- OASIS: Organization for the Advancement of Structured Information Standards
- SOAP: Simple Object Access Protocol
- HTTP: Hypertext Transfer Protocol
- tpaML: Trading Partner Agreement Markup Language
- UML: Unified Modeling Language
- UN/CEFACT: United Nations Centre for the Facilitation of Procedures and Practices in Administration, Commerce and Transport
- WSFL: Web Services Flow Language
- WSDL: Web Services Description Language
- WSIL: Web Services Inspection Language
- WSXL: Web Services Experience Language
- WSCL: Web Services Conversation Language
- WSUI: Web Services User Interface
- WSML: Web Services Meta Language
- WSCM: (Web Services Component Model) Numer omdöpt till WSIA
- WSIA: Web Services for Interactive Applications
- WSEL: Web Services Endpoint Language
- WSRP: Web Services for Remote Portals