

Obligatory Reading

2

- ▶ 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://tmitwww.tm.tue.nl/research/patterns/download/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
 - ▶ <http://www.omg.org/cgi-bin/doc?dtc/10-06-02> BPMN by example
 - ▶ <http://www.bpmnforum.com/FAQ.htm> FAQ of BPMN

Prof. U. Alsmann, CBSE



32. Web Services, Workflows and Service-Oriented Architectures

Prof. Dr. Uwe Alsmann
 Technische Universität Dresden
 Institut für Software- und
 Multimediale Technik
<http://st.inf.tu-dresden.de>
 Version 15-0.1, Mai 27, 2015

- 1) Web Services as a specific form of service-oriented architectures
 - 1) SOAP
 - 2) WSDL
- 2) Trading of Web Services
- 3) BPEL
- 4) BPMN
- 5) Web Workflows for CPS
- 6) Evaluation



CBSE, © Prof. Uwe Alsmann

1

The Ladder of Composition Systems

4

Software Composition Systems	Composition Language	<i>Invasive Composition Piccola Gloop</i>
Aspect Systems	Aspect Separation Crosscutting	<i>Aspect/J AOM</i>
View Systems	Composition Operators	<i>Composition Filters Hyperspaces</i>
Architecture Systems	Architecture as Aspect Connectors	<i>Darwin BPMN BPEL HRC</i>
Classical Component Systems	Standard Components Reflection	<i>.NET CORBA Beans EJB ArchJava</i>
Object-Oriented Systems	Objects as Run-Time Components	<i>C++ Java UML components</i>
Modular Systems	Modules as Compile-Time Components	<i>Shell scripts Modula Ada-85</i>

Prof. U. Alsmann, CBSE



Other Literature

3

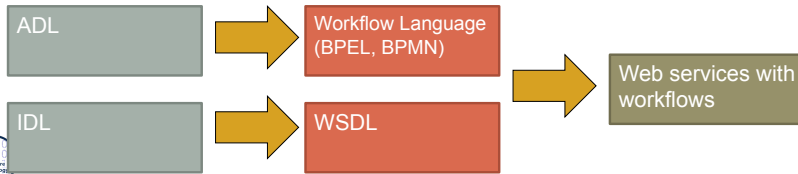
- ▶ Matthias Weske. Business Process Management – Concepts, Languages, Architectures. Springer. 2007
- ▶ YAWL <http://sourceforge.net/projects/yawl/>
- ▶ H. P. Alesso, C. F. Smith. Developing Semantic Web Services. A K Peters Ltd, Natick, Massachusetts, 2004.
- ▶ BPMN 2.0 language specification
 - ▶ <http://www.omg.org/spec/BPMN/2.0/PDF/>
- ▶ <http://www.bpmb.de/index.php/BPMNPoster>
- ▶ List of BPMN tool companies http://www.bpmn.org/BPMN_Supporters.htm
- 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. ICSSOC 2007, LNCS 4652.

Prof. U. Alsmann, CBSE



Web Services and Architecture Systems

- 6
- ▶ 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)



Prof. U. Aßmann, CBSE



32.1 Web Services as Architecture Systems

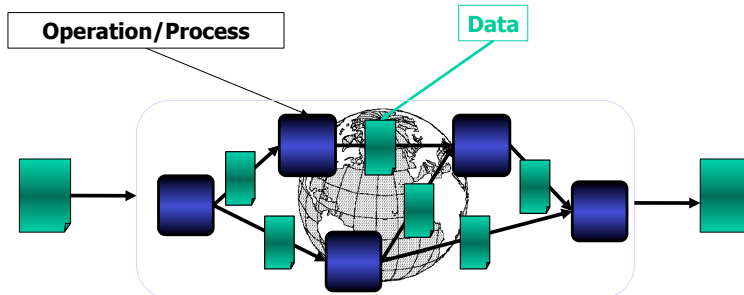


CBSE, © Prof. Uwe Aßmann

5

Web Service Architectures are Described by Workflows

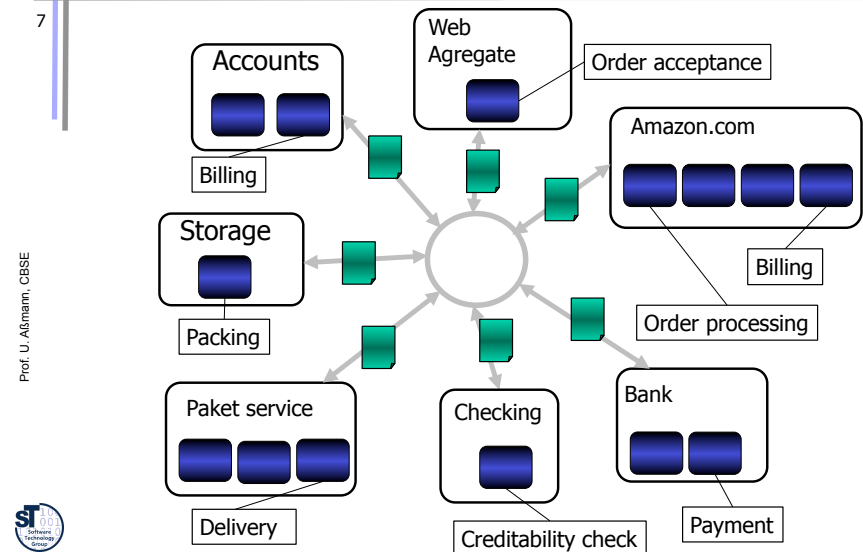
- 8
- ▶ Web service architectures are the first step to service-oriented architectures (SOA), based on traders
 - Services are offered, searched and discovered, downloaded, executed
 - Workflow specifications combine control and data flow
 - ▶ *Enterprise services* transfers web services to business systems
 - ▶ *Customer services* serve the end-user of the web



Prof. U. Aßmann, CBSE



Web Services are Black-Box Components



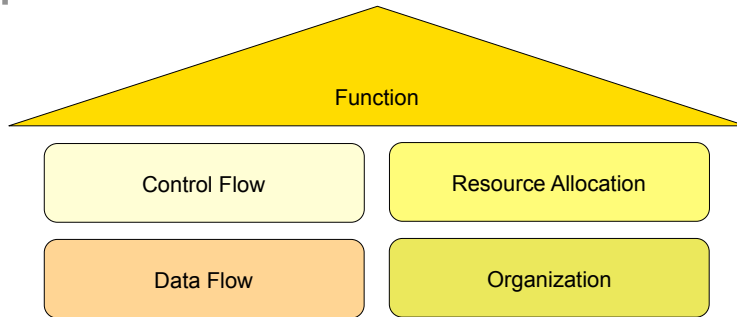
Prof. U. Aßmann, CBSE



Workflows Have Aspects

10

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



Prof. U. Alsmann, CBSE

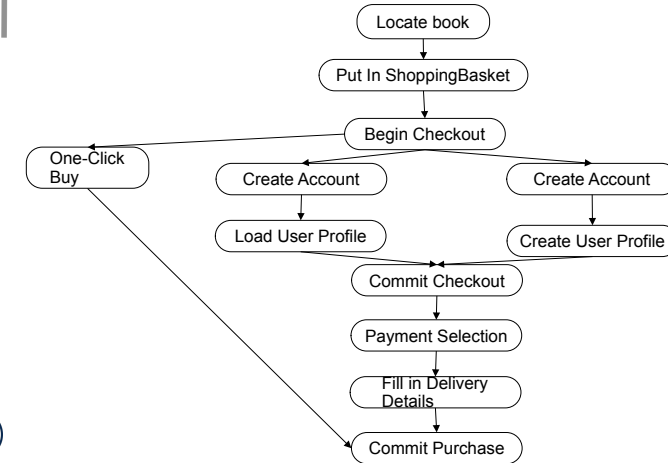


[Scheer]

Ex. Buying a Book from Amazon

9

- ▶ Workflows can be specified graphically as *activity diagrams*
 - ▶ E.g., with UML activity diagrams [Alesso/Smith]



Prof. U. Alsmann, CBSE



Workflow Languages

12

- ▶ 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, Eindhoven)
 - 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

Prof. U. Alsmann, CBSE



Which Types of Operational Specifications Exist for Workflows?

11

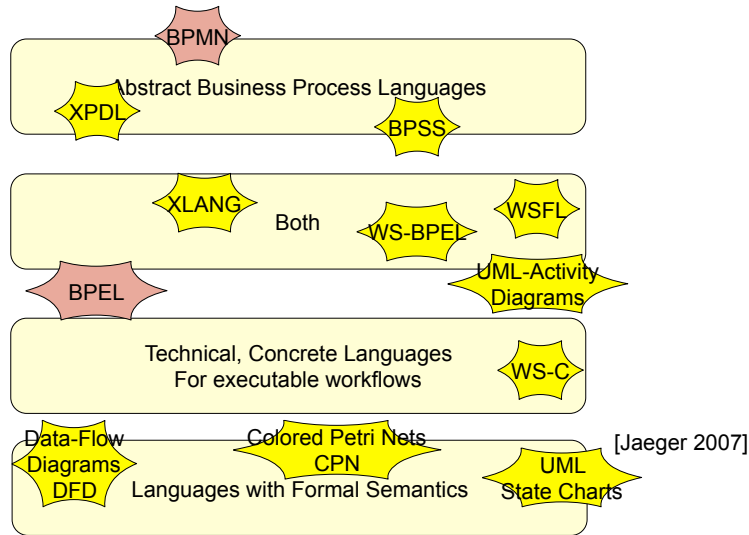
- ▶ **Data-flow graphs** (data flow diagrams, DFD) focus on data flowing through operations
 - Activity diagrams: data flows through actions
 - See courses Softwaretechnologie II, Software-Entwicklungswerkzeuge
- ▶ **Control-flow graphs** (CFG) focus on control dependencies
 - Nodes are control-flow operations that start other operations on a state
 - The standard representation for imperative programs
- ▶ **State systems** focus on transitions between states
 - Finite State Machines (FSM): events trigger state transitions
 - Statecharts: Hierarchical FSM
- ▶ **Mixed approaches**
 - **Colored Petri nets:** tokens mark control and data-flow, see course Softwaretechnologie II
 - **Cyclic data-flow graphs** (also called static-single assignment graphs, SSA)
 - Cycles are marked by phi-nodes that contain control-flow guards
 - **Workflow languages** mix control and data-flow
 - Provide specific split and join operators for control and data flow

Prof. U. Alsmann, CBSE



Languages Serve Different Abstraction Levels

14



Prof. U. Alsmann, CBSE



What is a Business Process? Business Processes are Abstract Workflows

13

- **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.
- An executable business process is called a **workflow** (*executable business process*).

"A collection of related, structured activities--a chain of events--that produce a specific service or product for a particular customer or customers."

www.gao.gov/policy/itguide/glossary.htm

"A business process is a recipe for achieving a commercial result. Each business process has inputs, method and outputs. The inputs are a pre-requisite that must be in place before the method can be put into practice. When the method is applied to the inputs then certain outputs will be created."

en.wikipedia.org/wiki/Business_process

Prof. U. Alsmann, CBSE



32.3 WSDL for the Definition of Interfaces of Web Services



What are Workflow Engines?

15

- ▶ **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

Prof. U. Alsmann, CBSE



WSDL Components and Their Interfaces

18

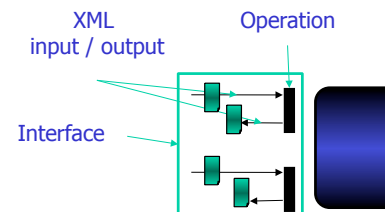
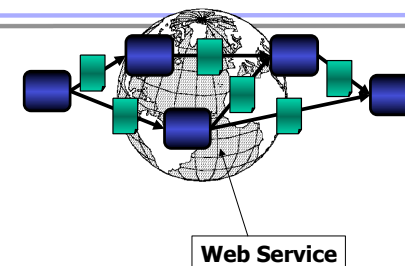
- ▶ A WSDL *Interface* consists of a set of ports
 - Functions with types of parameter and results in XML Schema
 - WSDL unifies call and event ports
 - Plays a similar role as ports of a UML component
- ▶ Advantages
 - WSDL abstracts from underlying protocol (http, SOAP, mime, IIOP)
 - Component model can be mapped to CORBA, EJB, DCOM, .NET
 - WSDL abstracts from the underlying component model, introducing the component model as a *secret*

Prof. U. Alsmann, CBSE



Service Interface

17



Prof. U. Alsmann, CBSE

Web Services Description Language (WSDL) defines a service interface



WSDL Reuses Data Types of XSD Here: Type Definitions <schema> <element> <complextype>

20

```

<wsdl:types>
  <XMLSchema:schema ... [target name space definitions]>
    <XMLSchema:element name="addTreatment">
      <XMLSchema:complextype>
        <XMLSchema:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="parameter"
            nillable="true" type="a:treatment"/>
        </XMLSchema:sequence>
      </XMLSchema:complextype>
    </XMLSchema:element>
    <XMLSchema:element name="addTreatmentResponse">
      <XMLSchema:complextype>
        <XMLSchema:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="result"
            nillable="true" type="XMLSchema:boolean"/>
        </XMLSchema:sequence>
      </XMLSchema:complextype>
    </XMLSchema:element>
    <XMLSchema:complextype name='treatment' ...
      </XMLSchema:complextype>
  </XMLSchema:schema>
</wsdl:types>
    
```

Prof. U. Alsmann, CBSE



WSDL Specification Structure

19

- ▶ **Types**
 - In XML schema or another typing language
- ▶ **Messages**
 - The data that is communicated (not in IDL)
- ▶ **Operation**
 - 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, e.g., http, soap, or mime
- ▶ **Service**
 - A set of related ports (as in UML components)

Prof. U. Alsmann, CBSE





Example: Binding WSDL to SOAP

22

```

<wsdl:binding name="livetoken" type="Token">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/html">
  <operation name="GetLastPrice">
    <soap:operation
      soapAction="http://www.stocktrade.com/GetPrice">
    <input> <soap:body use="literal"> </input>
    <output> <soap:body use="literal"> </output>
  </operation>
</wsdl:binding>

```

Prof. U. Alsmann, CBSE



32.3 Business Process Execution and Web Service Workflows with BPEL

BPEL, a web service composition language



Different Kinds of Port Types

21

Event- or message-based ports

- Notification: data-out port
- One-way: data-in port

Call ports:

- Request-Response: procedure port (callee port)
- Solicit-Response: send, then receive (caller port)

```

<wsdl:definitions [name space definitions]>
  <wsdl:types> ... </wsdl:types>
  <wsdl:message name="addTreatmentSOAPIn">
    <part name="parameters" element="addTreatment"/>
  </wsdl:message>
  <wsdl:message name="addTreatmentSOAPOut">
    <part name="parameters" element="addTreatmentResponse"/>
  </wsdl:message>
  <wsdl:porttype name="TreatmentAdminSOAP">
    <wsdl:operation name="addTreatment">
      <wsdl:input message="addTreatmentSOAPIn"/>
      <wsdl:output message="addTreatmentSOAPOut"/>
    </wsdl:operation>
  </wsdl:porttype>
  <binding [binding to SOAP / HTTP Protocols] ...
</wsdl:definitions>

```

Prof. U. Alsmann, CBSE



WSDL Service Interface

23

WSDL is a Interface Definition Language (IDL)

- Part of BPEL (see later)
- ▶ W3C Recommendation (standard)

Further development:

- ▶ USDL <http://theseus.pt-dlr.de/de/1309.php> with quality specifications
- ▶ Linked USDL <http://www.linked-usdl.org/> for connection to Linked Data

Prof. U. Alsmann, CBSE



Approach

Components of Web Services

WSDL	Service Description
SOAP	Messaging
XML	Message Description
HTTP, FTP, IIOP, MQ, ...	Network



Ingredients of BPEL

26

- ▶ 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*)

Prof. U. Alsmann, CBSE



BPEL Specification Structure

28

- ▶ **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

Prof. U. Alsmann, CBSE

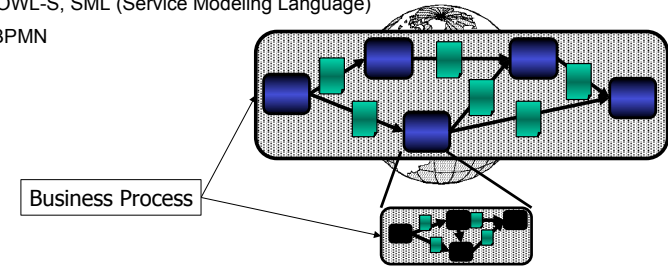


Business Process Definition

25

- ▶ BPEL is the major language for execution of business processes today
 - ▶ Business process on the web
 - ▶ IBM & Microsoft
- ▶ There are many languages proposed today:
 - WSFL, WSCL, WSCI, XLANG, WSEL, UML, WSUI, WSXL, BPML, BPMN ...
 - ▶ OASIS: WS BPEL
 - ▶ W3C: OWL-S, SML (Service Modeling Language)
 - ▶ SAP: BPMN

Prof. U. Alsmann, CBSE



BPEL Made Simple

27

- ▶ BPEL is a activity-diagram like language,
 - with parallelism and transactions
 - with different kind of join and split operators
 - with ports and connections
 - BPEL can be edited graphically, and has an XML abstract syntax
- ▶ To create a web service, becomes a similar activity as editing an UML activity diagram or Petri Net
- ▶ BPEL uses XML syntax
 - ▶ WSDL definitions to define types, message types, and port types
 - WSDL definitions can be without binding
 - Bindings can be added when the BPEL process is deployed
 - That increases reuse of the process
 - This achieves *component model transparency* (independence of the underlying component model)
 - ▶ *Partner link types* (connector types) describing typed connections

Prof. U. Alsmann, CBSE

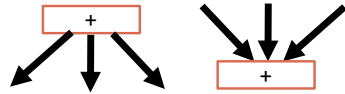


Typical Control-Flow Operators in BP and Workflow Languages (Gateways)

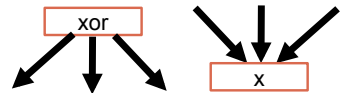
30

- Control-flow operators in Workflow languages are more complex than simple transitions in Petri Nets, which support only AND-split and -join

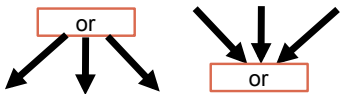
- AND-split: all
- AND-join: all of n



- XOR-split: 1 of n
- XOR-join: 1 of n



- OR-split: m of n
- OR-join: m of n



Prof. U. Alsmann, CBSE



BPEL Tools

32

- <http://en.wikipedia.org/wiki/BPEL>
- Eclipse BPEL project
 - <http://www.eclipse.org/bpel/>
- Orchestra tool
 - <http://orchestra.ow2.org/xwiki/bin/view/Main/WebHome>
- People work on the translation of Colored Petri Nets and UML activity diagrams from and to BPEL
 - CPN have good formal features (see ST-2)
 - Can be used for deadlock checking, resource control, etc.
 - YAWL is such a nice language, see the work of [van der Aalst]

Prof. U. Alsmann, CBSE



A Simple Pizza Order

29

```
<!-- 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

Prof. U. Alsmann, CBSE



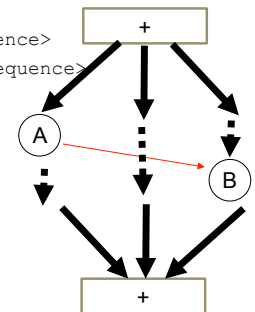
Flow Operations are Workflow Graphs

31

- The <flow> operation is structured as a workflow graphs
 - The names of messages, ports, partner links help to span up the graph
 - <flow> executes its sequences in parallel
 - <links> can synchronize parallel tasks

```
<flow>
<links><link> name="A"</link>
  <link>name="B"</link>
</links>
<sequence>..<invoke> <target name="A">..</sequence>
<sequence>...<target name ="B">.....</sequence>
</flow>
```

- Structured control-flow operators
 - Sequence, switch, while, flow
 - pick (XOR join), terminate
- Compensate (Error compensation)
- scope
- assign



Prof. U. Alsmann, CBSE

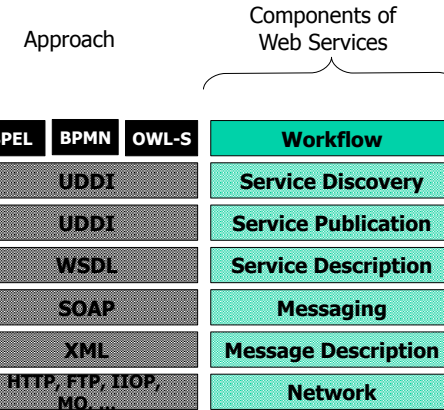


32.4 Business Process Modeling Notation (BPMN)

Another composition language

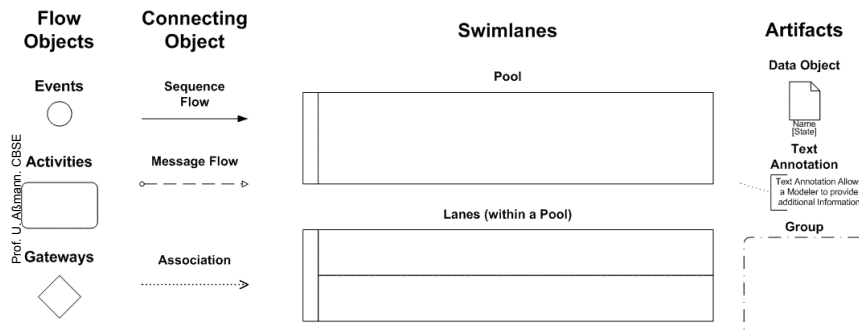


Business Process



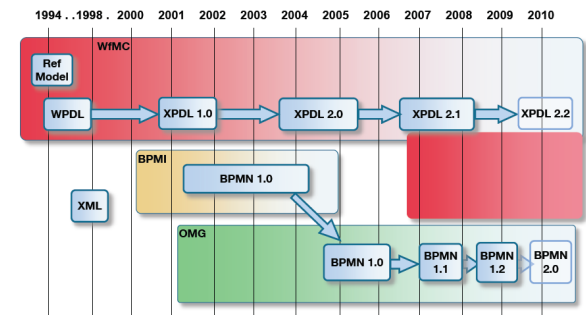
Core Elements

Core Set of BPMN Elements



History

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



Events and Activities

38

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

	Events		
	Start	Intermediate	End
	Event Types		
Message			
Timer			
Error			
Cancel			
Compensation			
Rule			
Link			
Terminate			
Multiple			

Prof. U. Alsmann, CBSE

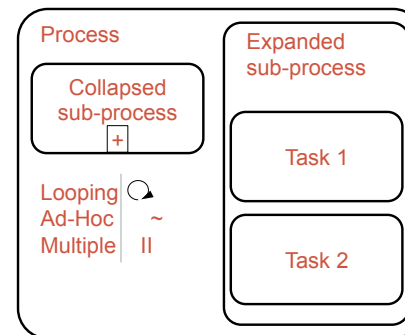


Activities and Processes

37

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).



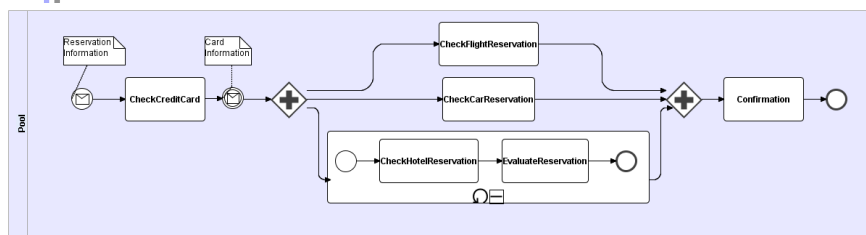
Prof. U. Alsmann, CBSE



Example: Travel Process Control Flow

40

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



Prof.



Gateways and Connections

39

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

Graphical connectors

	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.	
COMPLEX		Complex condition (a combination of basic conditions)	
AND		Parallel forking and joining (synchronization).	

Prof. U. Alsmann, CBSE



Normal sequence flow	
Conditional sequence flow	
Default sequence flow	
Message flow	
Association	

Why BPMN?

42

BPMN v2.x

- Modeling language for business processes: no execution semantics, only a partial mapping to Business Process Execution Language (BPEL)
- Explicit service mapping to web services (as components)
- Engines are available (jBPM for jBoss)

BPMN geared towards business analysts:

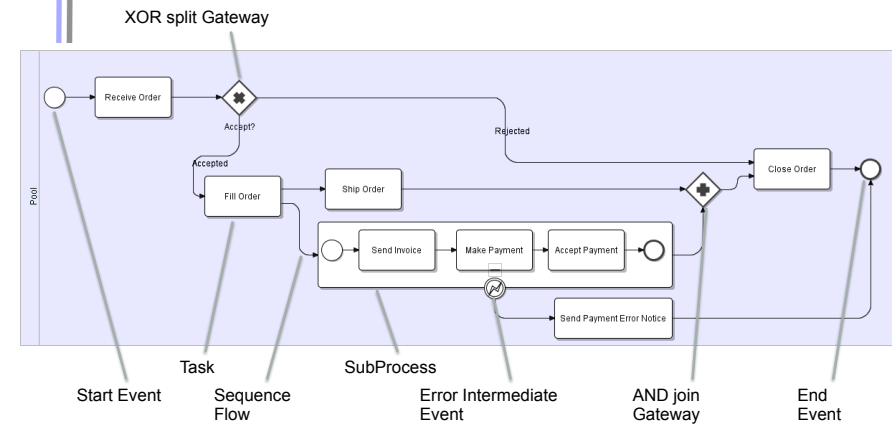
- BPMN constructs are simplified
- UML notation too bloated
- BPMN is on the platform-independent level, BPEL nearer the platform-specific level

Prof. U. Asmann, CBSE



Example

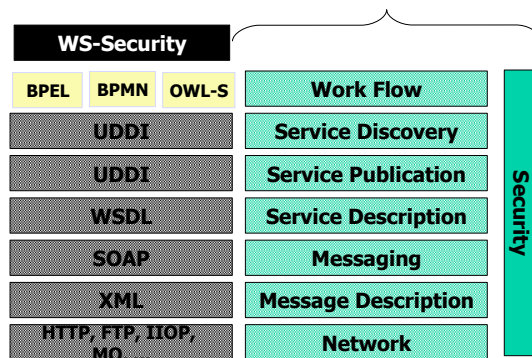
41



The Complete Hierarchy

44

Components of Web Services



Prof. U. Asmann, CBSE



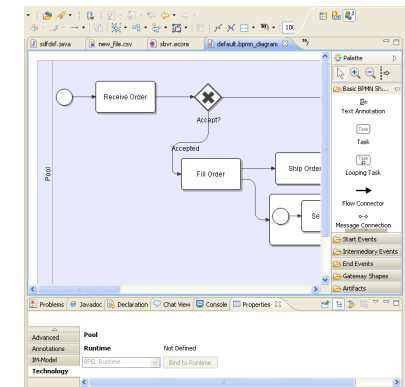
Give BPMN a try

43

Free BPMN Editor from Eclipse
Included in the SOA Tools Project

<http://www.eclipse.org/bpmn/>
<http://www.eclipse.org/bpmn2-modeler/>

SAP has decided to use BPMN in their products



Prof. U. Asmann, CBSE



32.5 Web Workflows for Cyber-Physical Systems

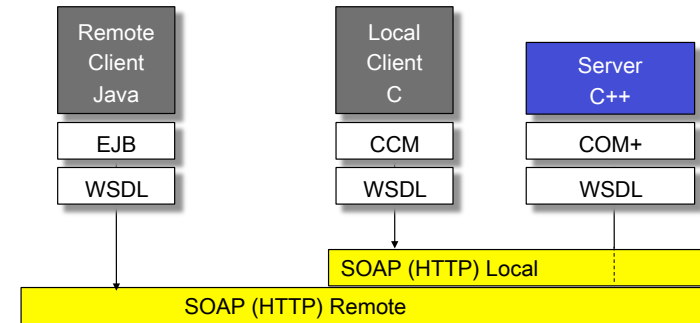


Web Services – Component Model Transparency

45

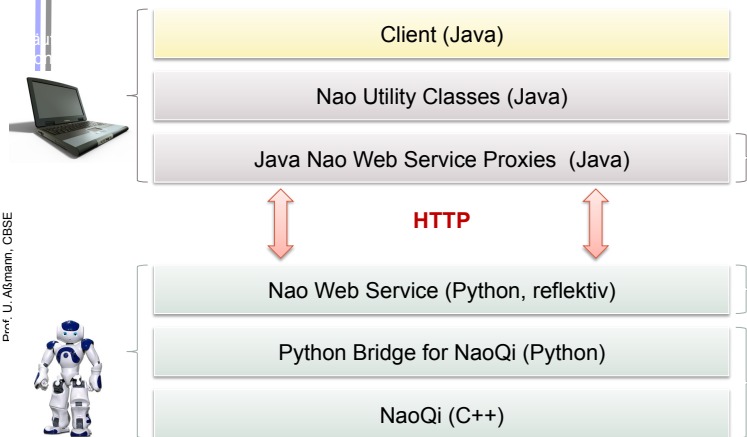
- ▶ Language adaptation: XML Schema + WSDL
- ▶ Remote transparency: SOAP (+ HTTP)
- ▶ Component model transparency (EJB, COM+, CORBA, CCM, Beans, etc...)

Prof. U. Aßmann, CBSE



Communication Layers between a Client and the Web Server on a NAO Robot

48

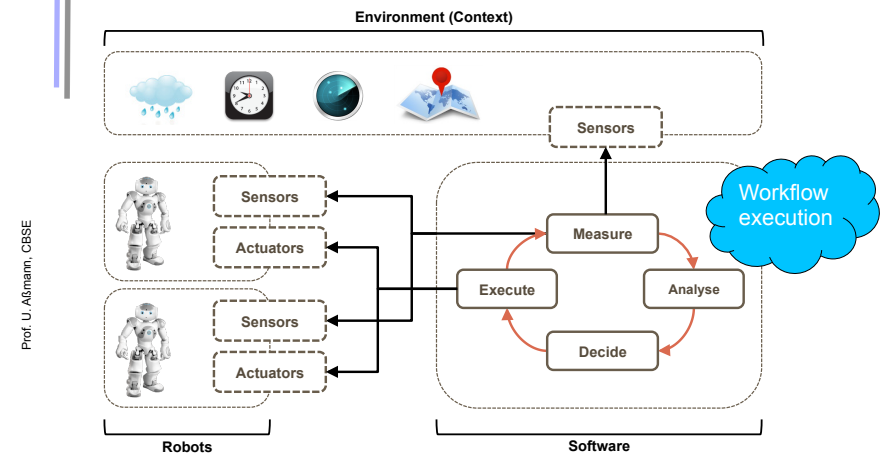


Prof. U. Aßmann, CBSE



Cloud Robots will be Controlled by Workflows on Web Services

47



Prof. U. Aßmann, CBSE



Control by Tablet

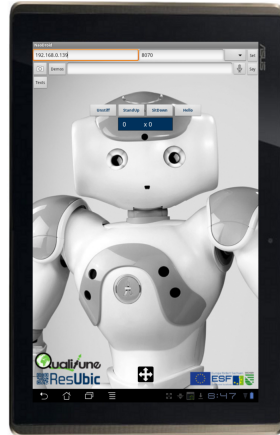
50

NAO Droid

Scripts can be started by buttons and „apps“
Dynamic scripting possible
Commands are transported to robot and executed on the web server

Film about Petri-Net controlled robot on ResUbic site

https://www.youtube.com/watch?feature=player_embedded&v=37DcE9e7phl



Prof. U. Aßmann, CBSE

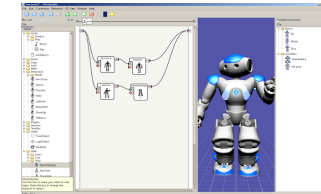


Control Software

49

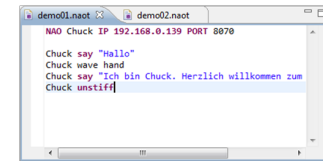
Choregraphe (from NAO-Aldebaran, Paris)

Workflow Language
Visual composition of workflows



NAOText (domain-specific language)

- Good abstractions for users in domain
- Expressive



Prof. U. Aßmann, CBSE



Component Model

52

- ▶ 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
- ▶ Generic BPEL Web Services are possible (without bound WSDL ports)
- ▶ BPMN Web Services can be stepwise refined from abstract to concrete

Prof. U. Aßmann, CBSE



32.6 Evaluation of Web Services

as composition system



Composition Language

54

- ▶ 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 workflow from other specifications
 - Generic workflow architectures will be possible

Prof. U. Alsmann, CBSE



Composition Technique

53

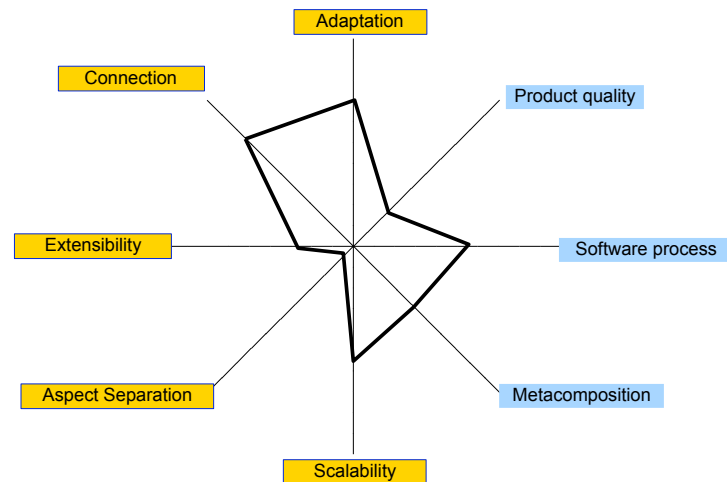
- ▶ Mechanisms for connection
 - Protocol transparency allows for flexible connections
 - WSDL binding is flexible
- ▶ Mechanisms for aspect separation
 - ▶ Separate modeling from execution (abstract business processes from workflows)
- ▶ Scalability: Better
 - Changes of protocol possible
 - Changes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

Prof. U. Alsmann, CBSE



Web Services – Composition Technique and Language

56

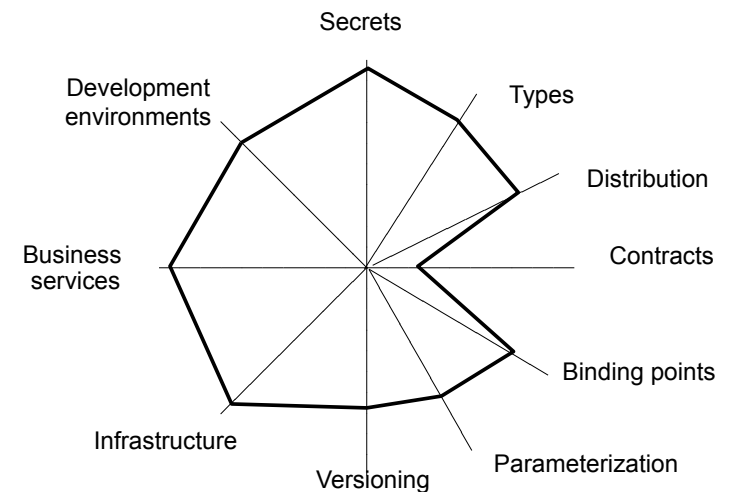


Prof. U. Alsmann, CBSE



Web Services - Component Model

55



Prof. U. Alsmann, CBSE



So Far: Blackbox Composition Systems

58

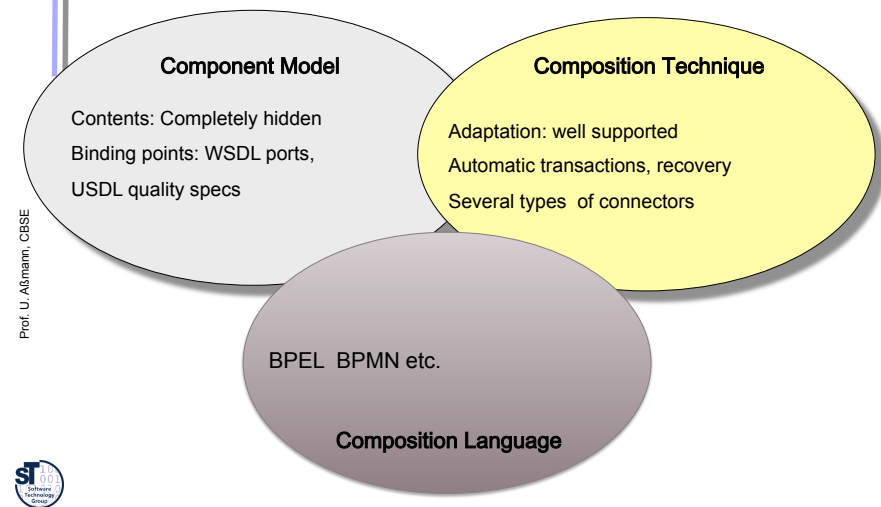
Software Composition Systems	Composition Language	<i>Invasive Composition</i> <i>Piccola Gloop</i>
Aspect Systems	Aspect Separation Crosscutting	<i>Aspect/J</i> <i>AOM</i>
View Systems	Composition Operators	<i>Composition Filters</i> <i>Hyperspaces</i>
Architecture Systems	Architecture as Aspect Connectors	<i>Darwin</i> <i>BPMN BPEL HRC</i>
Classical Component Systems	Standard Components Reflection	<i>.NET CORBA</i> <i>Beans EJB</i> <i>ArchJava</i>
Object-Oriented Systems	Objects as Run-Time Components	<i>C++ Java</i> <i>UML components</i>
Modular Systems	Modules as Compile- Time Components	<i>Shell scripts</i> <i>Modula Ada-85</i>

Prof. U. Alsmann, CBSE



Web Services as Composition Systems

57



Prof. U. Alsmann, CBSE



The End

60

- ▶ 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

Prof. U. Alsmann, CBSE



The Second Part of the Course: Greybox Composition

59

Generic programming Generic program elements Home-made connectors	Invasive software composition Slots and hooks
View-based programming Formal foundations (lambda N, pi-calculus) Record calculi, Scala Hyperspace programming	Novel Forms of Composition Uniform composition Active document composition: Architectural styles for active documents and web systems
Aspect-oriented development Aspect-oriented programming Aspect-oriented design	Roundtrip engineering and composition

Prof. U. Alsmann, CBSE





Some URLs

62

- ▶ www.ebxml.org
- ▶ www.uddi.org
- ▶ www.oasis-open.org
- ▶ www.uncefact.org
- ▶ www.w3.org
- ▶ www.omg.org
- ▶ www.biztalk.org
- ▶ www.soapclient.com
- ▶ www.software.org
- ▶ www.xml.com
- ▶ www.xml.org
- ▶ www.webservices.org
- ▶ www.webservicesarchitect.com
- ▶ www.ws-i.org

Prof. U. Alsmann, CBSE



Some Abbreviations

61

- ▶ 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
- ▶ WSM: 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

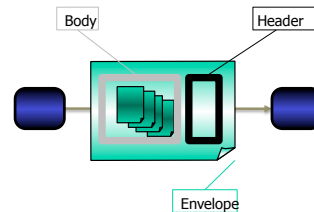
Prof. U. Alsmann, CBSE



32.1.2 Communication with SOAP, an XML-based Interaction Protocol

64

- ▶ Simple Object Access Protocol (SOAP) defines a message format
- ▶ Message contains target address and an envelope
 - with name space, encoding attributes and
 - Header (fixed format) contains
 - Authentication (Sender, Receiver),
 - Transactions,
 - Error handling information,
 - Routing information ...
 - Body contains user data (free format)
- ▶ Transport is transparent, predefined channels:
 - HTTP (with back channel, de facto standard)
 - SMTP, TCP (with back channel)



Prof. U. Alsmann, CBSE

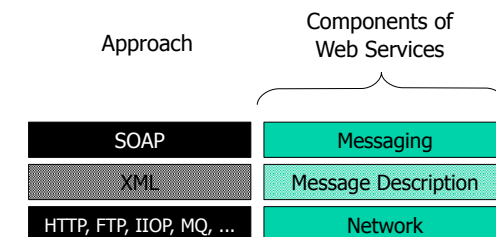


SOAP Interaction Protocol

63

- + W3C Recommendation (standard)
- + Implements RPC
- Untyped user data, types to encode in the message
- Interpretation of SOAP messages required
- High overhead / low performance

Prof. U. Alsmann, CBSE



Example: SOAP Body

66

```
<SOAP-ENV:Body>
<m:AddTreatment xmlns:a=http://localhost/TreatmentAdmin>
<treatment>
  <patient insurer="1577500"nr='0503760072' />
  <doctor city ="HD" nr='4321' />
  <service>
    <mkey>1234-A</mkey>
    <date>2001-01-30</date>
    <diagnosis>No complications.
  </diagnosis>
  </service>
</treatment>
</SOAP-ENV:Body>
```

Prof. U. Alsmann, CBSE



OWL Web Ontology Language

68

- ▶ Classes and relationships
- ▶ Expressions to compute (derive) new classes and relationships (*derived model*)
 - Union, intersection of relations and classes
 - Cardinality restrictions
 - Existential quantifiers
- ▶ Roughly speaking, OWL corresponds to UML-class diagrams without methods + OCL + class expressions
- ▶ Instead of plain XML, OWL can be used to type data
 - Beyond trees and context-free structures, graphs, knowledge webs, semantic nets can be described (context-sensitive structures)

Prof. U. Alsmann, CBSE



Example: SOAP Header

65

```
POST /TreatmentAdmin HTTP/1.1
HOST: www.hospital-admin.com
Content-Type: text/xml
Charset="utf-8"
Content-Length: nnnn
SOAPAction: http://localhost/TreatmentAdmin
```

Message Header
HTTP

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV=http://schemas.xmlsoap.org/soap/envelop/
SOAP-ENV:Encoding="http://.../encoding">
  <SOAP-ENV:Header>
    <a:Authentication
      xmlns:a=http://localhost/TreatmentAdmin ... >
      ...
    </a:Authentication>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body ... </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP Envelop

Prof. U. Alsmann, CBSE



32.9 OWL-S (Web Ontology Language for Services)

Additional material

- ▶ OWL-S definition at <http://www.w3.org/Submission/OWL-S/>
- ▶ <http://daml.semanticweb.org/services/owl-s/1.0>





OWL-S Processes

70

- ▶ **Atomic**
 - Cannot be decomposed
 - Can be called and executed
 - Can be mapped to WSDL process descriptions (*grounding*), and hence, to SOAP
- ▶ **Simple**
 - Cannot be decomposed
 - Can be executed, but not be called from outside
- ▶ **Composite**
 - Build from atomic and simple processes

Prof. U. Alsmann, CBSE



Creating an OWL-S specification

72

- ▶ Describe atomic processes
- ▶ Describe grounding of atomic processes
- ▶ Describe compositions
- ▶ Describe simple processes
- ▶ Describe profile of service

Prof. U. Alsmann, CBSE



OWL-S

69

- ▶ Based on OWL, a language for specification of web services has been developed by the OWL-S coalition
- ▶ Specification has three parts:
 - *Service profile*: semantic service description, service offer, service functionality (*what does the service provide?*)
 - Based on domain ontologies in OWL, i.e., OWL-specified attributes
 - *Service model*: service realization, decomposition of a service (*how does the service work?*)
 - Service is also called a *process*
 - Here, OWL-S provides a process ontology
 - *Service grounding*: service mapping to underlying mechanisms (*how is the service mapped to a component model and transport protocol?*) Similar to WSDL grounding

Prof. U. Alsmann, CBSE



Service Model (Process Model) of OWL-S

71

- ▶ **Process Ontology**
 - Describes a service (process) with an *IOPE* specification
 - Inputs
 - Outputs
 - Parameters
 - Effects
- ▶ **Process control ontology (for composite processes)**
 - Internal realization with state, activation, execution, completion (control-flow specification)

Prof. U. Alsmann, CBSE



32.10 Trading of Web Services



OWL-S Statements of a Composite Process

73

- ▶ Unordered (unspecified order)
- ▶ Sequence
- ▶ Split
- ▶ Split+Join (fork and join)
- ▶ Concurrent
- ▶ Choice
- ▶ If-then-else
- ▶ Repeat-until
- ▶ Repeat-while



UDDI

76

White Page

Registered (and other) names
Service Description
Contact person (name, e-mails, ...)
Telephone/fax number
Web site
...

Yellow Pages

Service category
Type of industry
Type of products/services
Geographic localization
...

Green Pages

Offered service
Documentation, description
Principles cooperation realization
...

WSDL

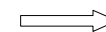
WSDL



Trading on the Web: Offer and Find Services

75

- ▶ Standardized publishing, advertisement ...
- ▶ Extended name server, describing interface and properties
- ▶ XML Descriptor
 - White Page: Address
 - Yellow Page: Semantics (based on standard taxonomy)
 - Green Page: Technical specification of service
- ▶ Logically central, physically distributed data base



Universal Description, Discovery and Integration (UDDI) defines service properties for service trading



UDDI: Offer and Find Services

77

- Required and approached
- **No** search strategies
- **No** no trader or market place

Prof. U. Alsmann, CBSE



Approach	Components of Web Services
UDDI	Service Discovery
UDDI	Service Publication
WSDL	Service Description
SOAP	Messaging
XML	Message Description
HTTP, FTP, IIOP, MQ, ...	Network