Semantic Web Technologies II

SS 2008

11.06.2008

Semantic Web Services

Dr. Peter Haase
PD Dr. Pascal Hitzler
Dr. Steffen Lamparter
Denny Vrandecic

Prüfungstermine

- Prüfungstermine
  - 17.07.2008
  - 17.09.2008

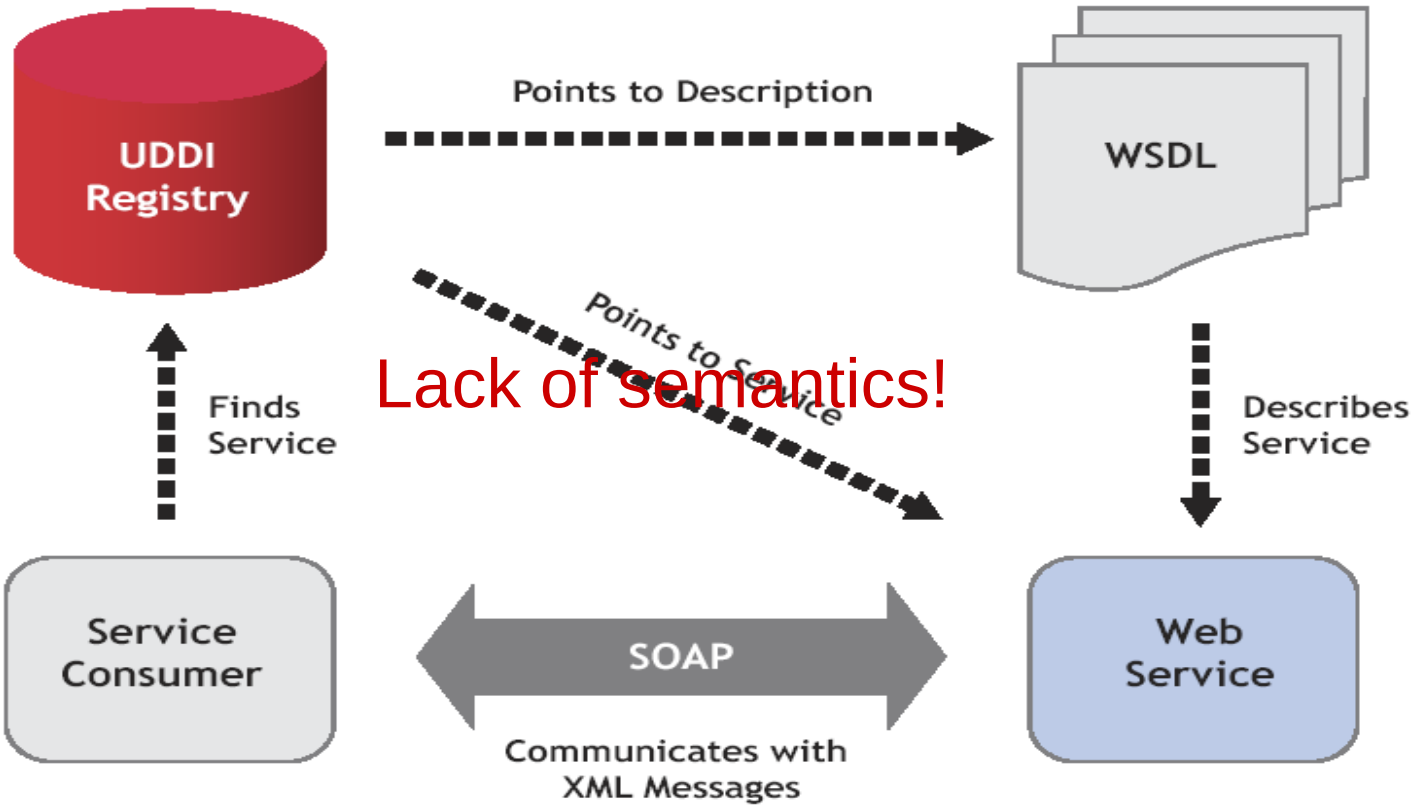
- Anmeldung:
  - Anmeldung zuerst wie gewohnt über System
  - Zusätzlich Mail an Pascal Hitzler (phi@aifb.uni-karlsruhe.de), welcher Termin gewünscht wird

Agenda

- Web Services – Foundations
- Semantic Web and Web Services
- Semantic Mash-ups
- Semantic Management of Web Services

SEMANTIC WEB AND WEB SERVICES

WS standards



Problem: Interoperability by using the same format, but still only syntax-based … no way to describe functionality

Current State: Web Services Standards I

SOAP

- XML-based web services communication protocol
- *Limitations:*
  - *No description of format and role of message in WS interaction*

WSDL

- Structured mechanism to describe a WS interface
  - Abstract operations that a Web Service can perform
  - Format of messages it can process
  - Protocols it can support
  - Physical bindings to URIs and protocols
- *Limitations: no semantics for message sequencing, correlation and content*

Current State: Web Services Standards II

UDDI

- Directory service for Web Services
- *Limitations: only keyword searches, limited functionality search*

BPEL

- Description of how Web Services are composed together
  - structure of the WS in terms of individual process steps, data flow links and control flow links (Flow Model)
  - interaction between provider and requester and mappings between internal operations and WSDL port types (Global Model)
- *Limitations:*
  - *No description of inputs, outputs, preconditions and effects of WS*
  - *Only manually constructed compositions*
    - → *Static binding to Web services*
    - → *Predefined process definition*

The Integration Challenge

Integrate multiple independent and heterogeneous

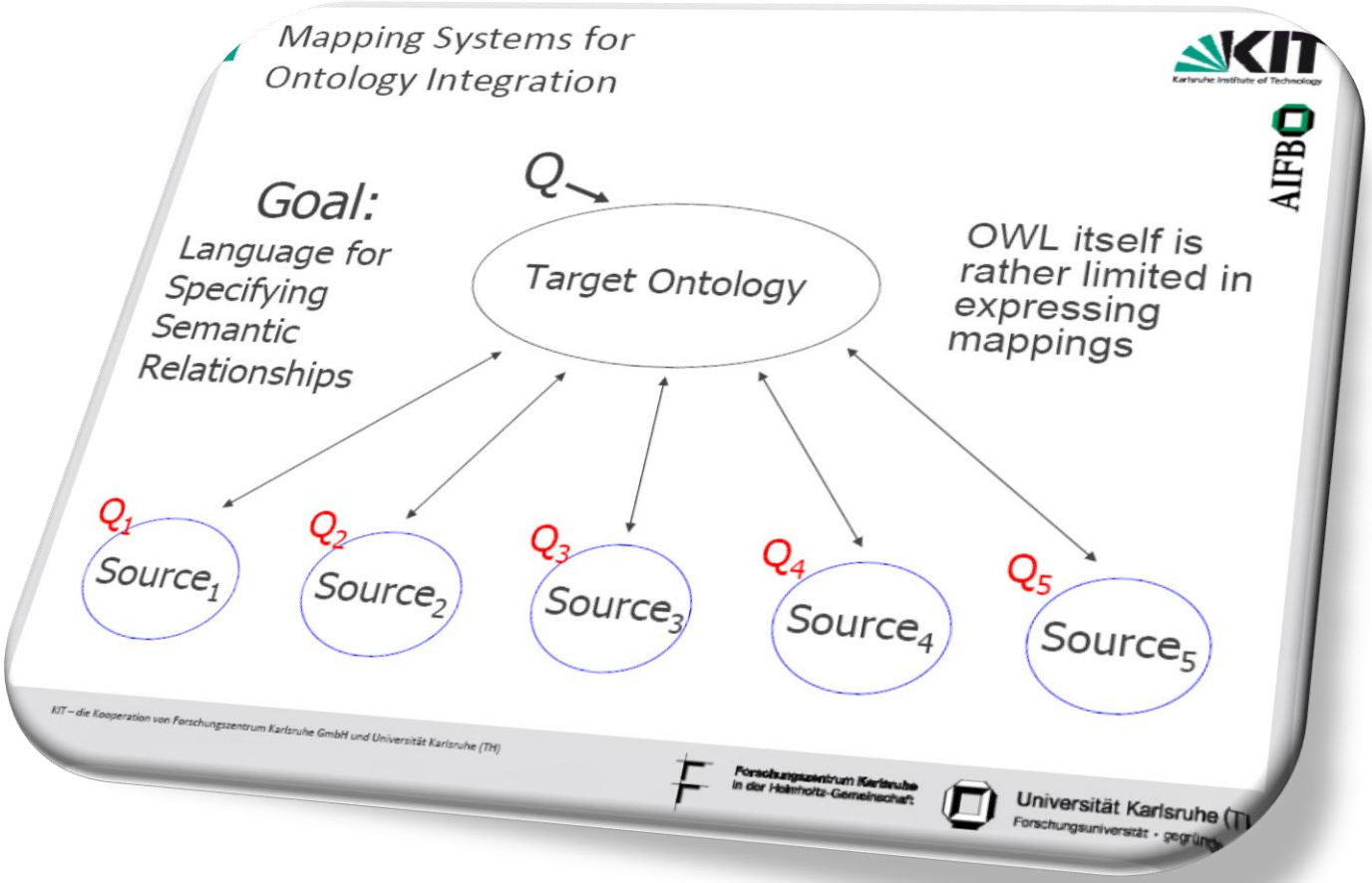
- Data repositories
- Processes
- Applications

Ensure

- Semantic equivalence of equivalent concepts
- Flexibility: systems enter and leave integration
- Performance

Information Integration using Ontologies

- Lecture “Information Integration”



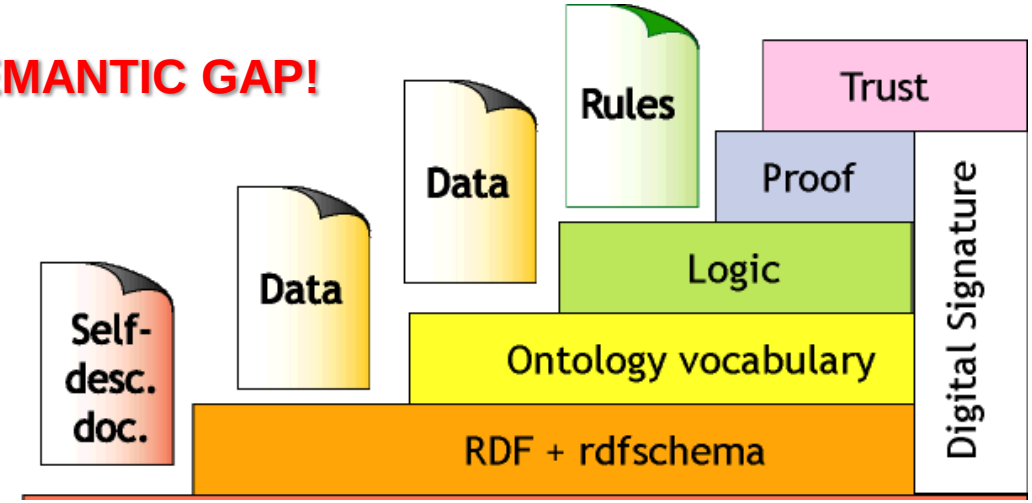
Semantic Web Services Stack Diagram

Semantic Web Services

Eg Contracts; Information Integration; Business Process Automation

Web Services Stack

- Choreography eg BPEL
- Process Specifications
- Directories, e.g. UDDI
- WSDL
- SOAP



Semantic Web Stack

XML | Namespaces

URI | Unicode | Encryption, Signature

Lack of Semantic Interoperability …

A problem for …

- **Discovery**
  - Different terms used for offers and requests
- **Invocation**
  - Different specifications for messages, protocols and WS interface
- **Understanding**
  - Interpreting the results returned by the Web service
- **Composing Services**
  - Reconciling private goals with goals of the WS

SWS as Web services that receive and send semantic data
→ Semantic Mash-ups

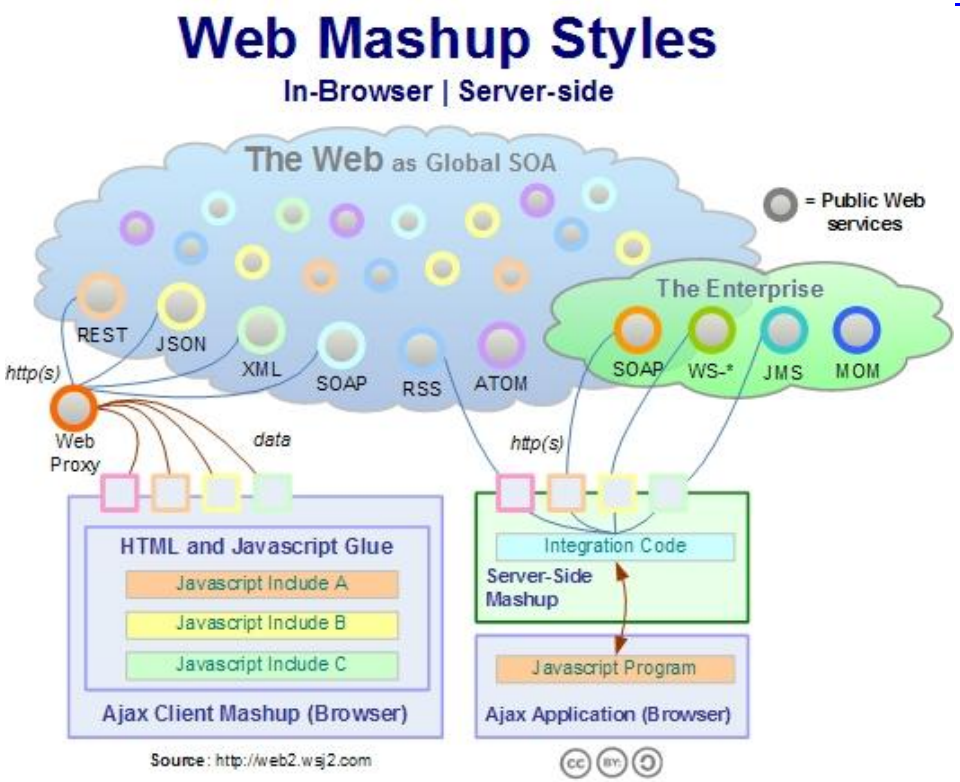
SWS as Web services that are managed by means of semantic descriptions
→ Semantic Management of WS

SEMANTIC MASH-UPS

What are Mash-ups?

- "A Mashup is a new service, that combines functionality or content from existing sources. These existing sources can be Web Services (through the use of API's), RSS feeds or even just other Websites (by screen-scraping)"

http://blog.sherifmansour.com/?p=187



Mashups = (REST + 'Traditional SOA') * Web 2.0

Traditional enterprise software (ERP, SCM, etc.) → **Orchestration via e.g. BPEL**

Situational Applications → **Mash-ups**

Mashups

http://www.programmableweb.com/matrix

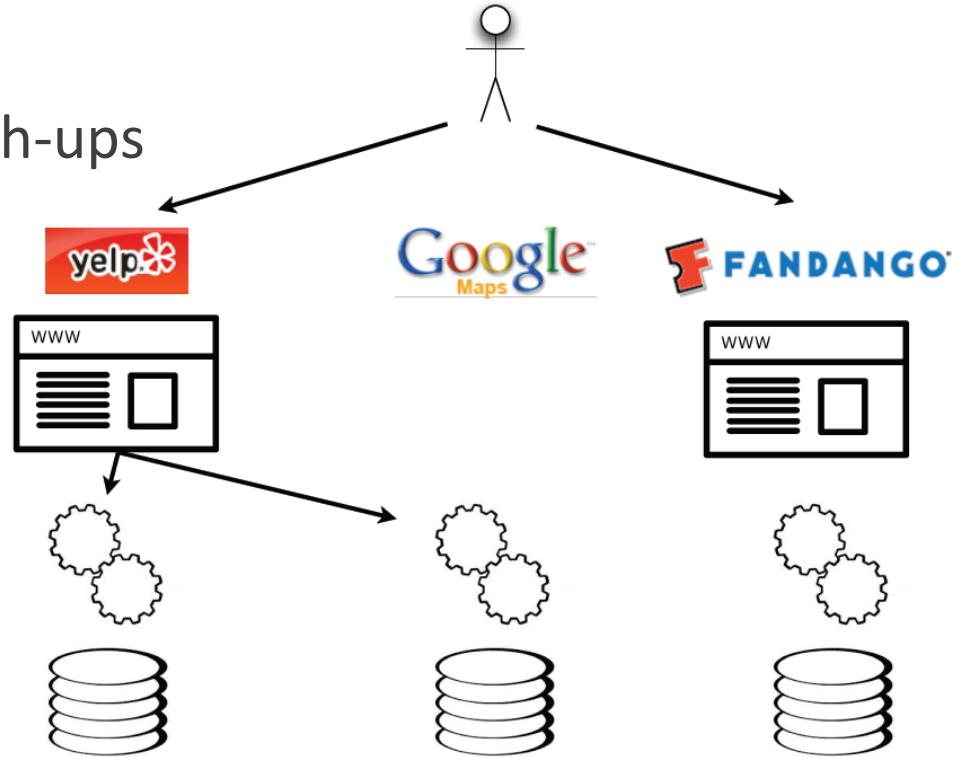
Mash-up Tools

- Yahoo! Pipes (http://pipes.yahoo.com/pipes/)
- Microsoft Popfly (http://www.popfly.com/)
- Google Mashup Editor (http://editor.googlemashups.com)
- IBM MashupHub (http://services.alphaworks.ibm.com/mashuphub/)
- …

- The battle of mashup editors:
  - http://usatoday.com.com/8301-10784_3-9747138-7.html

Web 2.0 Mash-ups

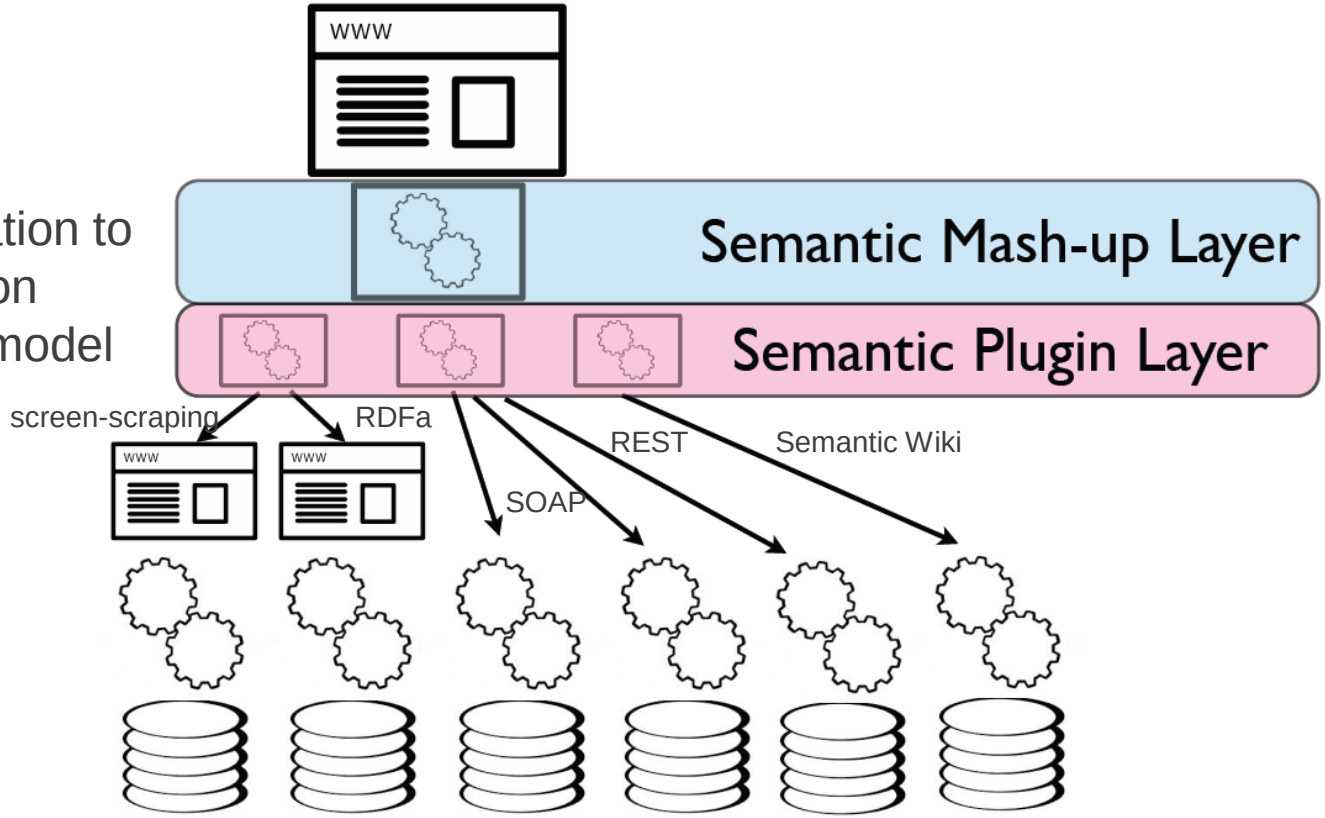
- Pure UI Mash-ups:
  - Leverage silos of content
  - User-generated content
  - Open APIs facilitate mash-ups
  - The “Social Web”
- Problems:
  - Mash-ups only allow shallow integration at the UI level
  - Data is still in silos
  - User-generated content is also in silos



http://en.oreilly.com/webexsf2008/public/asset/attachment/2386

Semantic Mash-Ups

Transformation to common semantic model

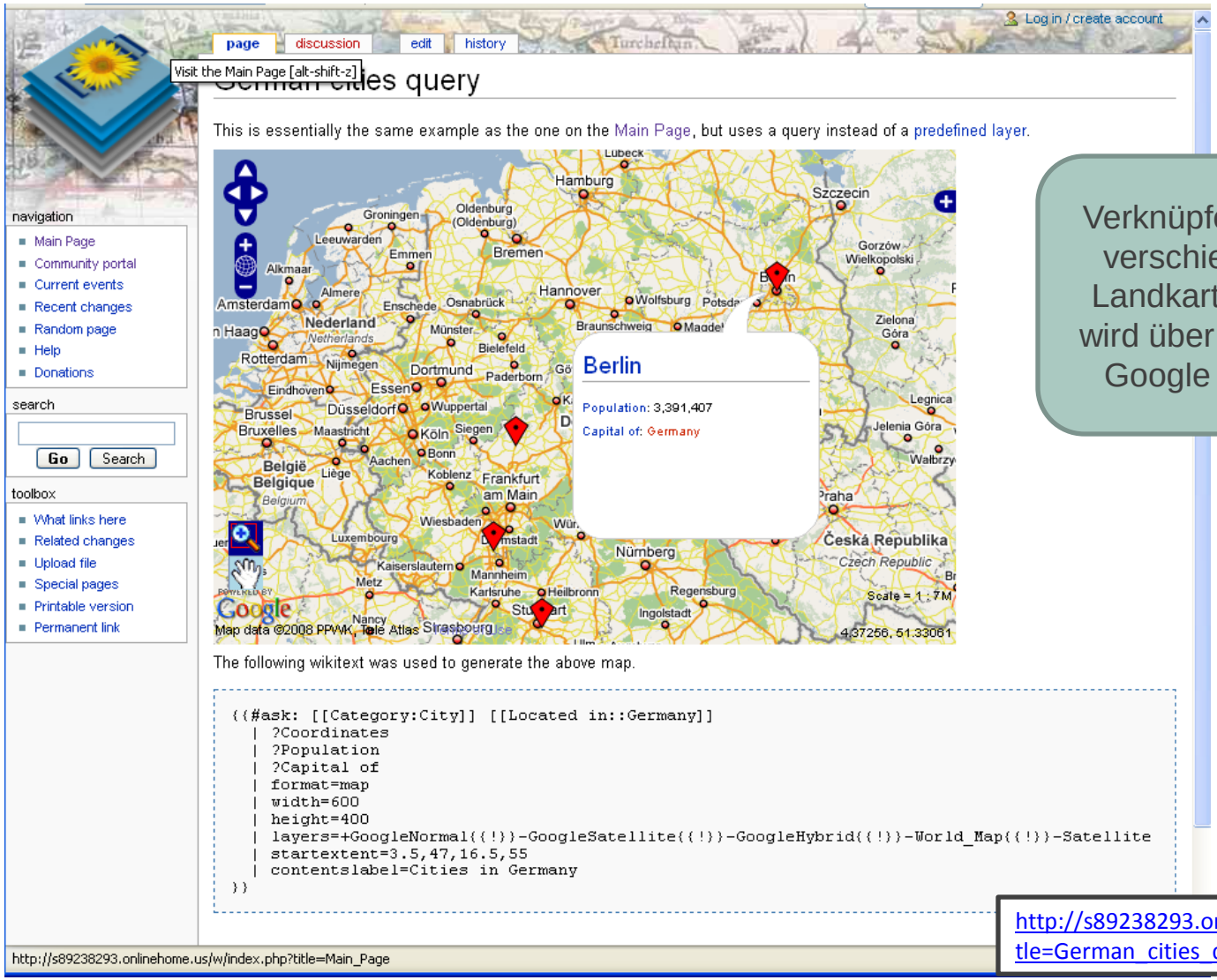


http://en.oreilly.com/webexsf2008/public/asset/attachment/2386

Anwendungsbeispiel: Service Mashups

- Semantische Erweiterung von MediaWiki die Typisierung von Links und die Annotation einer Seite mit Fakten.
  - Karlsruhe is a city in [[country::Germany]] with an area of [[area::173.46 km²]].

Anwendungsbeispiel: Service Mashups



Verknüpfen von Daten aus verschiedenen Quellen: Landkarte mit Wiki-Daten wird über den Web Service Google Maps generiert.

http://s89238293.onlinehome.us/w/index.php?title=German_cities_query

Semantic Mash-ups

- More details on this topic in the lecture “Semantic Web 2.0“
  - 25. Juni 2008
  - 2. Juli 2008

Karlsruhe Institute of Technology

AIFB

SEMANTIC MANAGEMENT OF WEB SERVICES

Semantic Web Services

Objective: Semi-Automation of the Web Service Usage/Management Process:

- Publication: Make available the description of the functionality of a service
- Discovery: Locate different services suitable for a given task
- Selection: Choose the most appropriate services among the available ones
- Composition: Combine services to achieve a goal
- Mediation: Solve mismatches (data, protocol, process) among the combined
- Execution: Invoke services
- Monitoring: Control the execution process
- Compensation: Provide transactional support and undo unwanted effects
- Replacement: Facilitate the substitution of services by equivalent ones

Problem of Discovery

- Requester and provider have different views
  - Provider sells Mutual Funds
  - Requester wants Mutual Funds of European Companies
- There is no exact match
- Matchmaker or broker needs to exploit the semantics of advertisements and requests to recognize partial matches

Problem of Composition

- No single Web service may achieve all goals of an agent
  - Composition is the process of chaining results from different Web services automatically
- Planning problem
  - How do the Web services fit together?
- Interoperation problem
  - How does the information returned fit together?

Overview

- Service Description Ontologies
  - SAWSDL
  - OWL-S
  - WSMO
- Service Discovery
- Service Selection

Further readings:
• Chapter 3 & 7, Studer, Grimm, Abecker (Eds), Semantic Web Services - Concepts, Technologies, and Applications, Springer, 2007.

Semantic Service Description Languages

- Should describe all information necessary to enable
  - automating discovery, composition, execution, etc.
  - semantically enhanced repositories

- Main efforts (W3C Submissions, http://www.w3.org/Submission/):
  - SAWSDL (former WSDL-S, 2005)
  - OWL-S (2004)
  - WSMO (2005)
  - SWSF (2005)

Common Features and Differences

- Ontologies for describing properties of Web Services
  - Different logical foundations
  - Different expressivity wrt functional/non-functional service properties
- Provide some grounding with current WS standards (particularly WSDL)
  - Different usage models: pointer from WSDL to semantic description or vice versa
- Common goals: improve/automate discovery, composition, selection, mediation, etc.

The SAWSDL Approach

- Semantic Annotations for WSDL and XML Schema (SAWSDL)
  - W3C Recommendation (August 2007)
  - Specification: http://www.w3.org/TR/sawsdl/
  - Originally „WSDL-S“:
    - W3C workshop decided for a “evolutionary approach with limited scope”
    - No precondition and effects
    - No support for choreography and orchestration

The SAWSDL Approach

- Generally, SAWSDL is about adding semantic annotations to XML schema
- It is a means to add semantics inline to WSDL for:
  - Inputs and outputs
  - Operations
  - Service categorization
- It is agnostic to ontology representation language used
  - OWL /RDF ontologies used up to now
  - No barrier to the use of other logical formalisms

Semantic Annotations to WSDL

```
<wsdl11:types>
   <xs:schema
        targetNamespace="http://www.w3.org/2002/ws/sawsdl/spec/wsdl/order#"
        xmlns="http://www.w3.org/2002/ws/sawsdl/spec/wsdl/order#"
        elementFormDefault="qualified">
            <xs:element
                 name="OrderRequest"
                 sawsdl:modelReference="http://www.w3.org/2002/ws/sawsdl/spec/ontology/purchaseorder#OrderRequest"
                 sawsdl:liftingSchemaMapping="http://www.w3.org/2002/ws/sawsdl/spec/mapping/Response2Ont.xslt">
                 <xs:complexType>
                      <xs:sequence>
                           <xs:element name="customerNo" type="xs:integer" />
                           <xs:element name="orderItem" type="item" minOccurs="1" maxOccurs="unbounded" />
                      </xs:sequence>
                 </xs:complexType>
            </xs:element>
            <xs:complexType name="item" sawsdl:modelReference="http://.../spec/ontology/purchaseorder#Item">
                 <xs:all>
                      <xs:element name="UPC" type="xs:string" />
                 </xs:all>
                  <xs:attribute name="quantity" type="xs:integer" />
            </xs:complexType>
    </xs:schema>
</wsdl11:types>
```

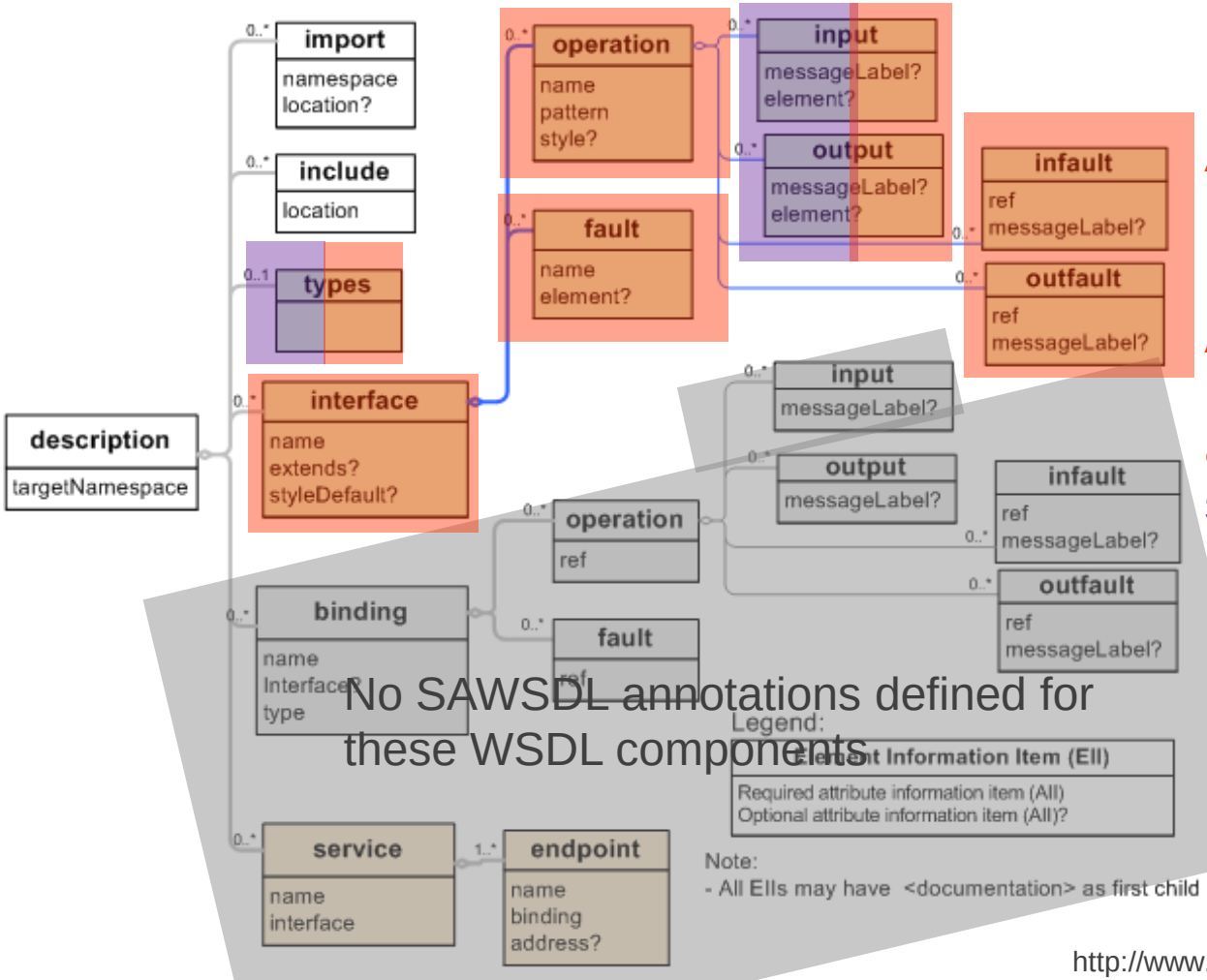
Annotate with Model Reference

Annotate with Schema Mapping

Annotate complex type with Model Reference

http://www.w3.org/TR/sawsdl/

Semantic Annotations to WSDL



Annotated using modelReference

Annotated using modelReference and schemaMapping

http://www.w3.org/TR/wsdl20-primer/

The OWL-S Appraoch

- OWL-S is an OWL ontology to describe Web services
- http://www.w3.org/Submission/OWL-S/
- Relation to WS*-standards

	OWL-S	***Web Services Infrastructure***
Discovery ***What it does***	Profile	*UDDI API*
Orchestration ***How is done***	Process Model	*BPEL4WS*
Invocation ***How to invoke***	Grounding+ WSDL/SOAP	*WSDL/SOAP*

OWL-S Upper Ontology

WS functionality specification to assist discovery
- Used for advertisement of service functionality
- Used for requesting WS with desired capabilities

Mapping to WSDL
- communication protocol (RPC, HTTP, …)
- transformation to and from XSD to OWL

- Control flow of the service
  - Black/Grey/Glass Box view
- Protocol Specification
- Abstract Messages

Service Profile Description

- **Preconditions**
  - Set of conditions that should hold prior to service invocation
- **Inputs**
  - Set of necessary inputs that the requester should provide to invoke the service
- **Outputs**
  - Results that the requester should expect after interaction with the service provider is completed
- **Effects**
  - Set of statements that should hold true if the service is invoked successfully.
- **Service type**
  - What kind of service is provided (eg selling vs distribution)
- **Product**
  - Product associated with the service (eg travel vs books vs auto parts)
- **Service Parameters** (e.g. security, QoS etc)

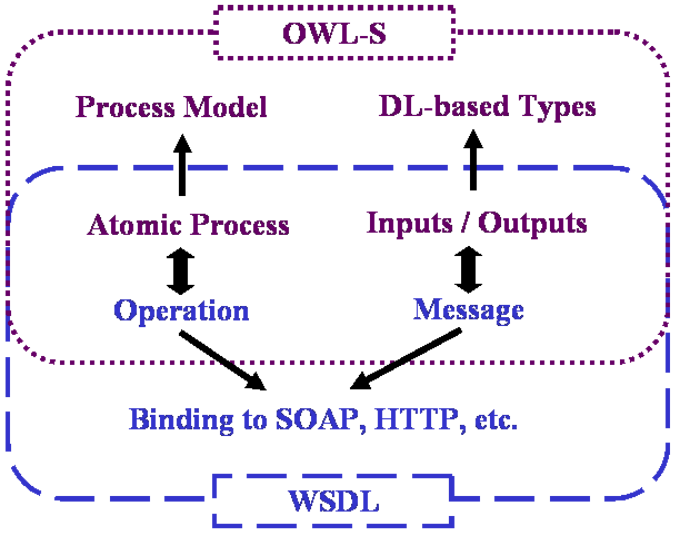
Example of an atomic Process

Inputs / Outputs
Precondition
Result
Condition
Output Constraints
Effect

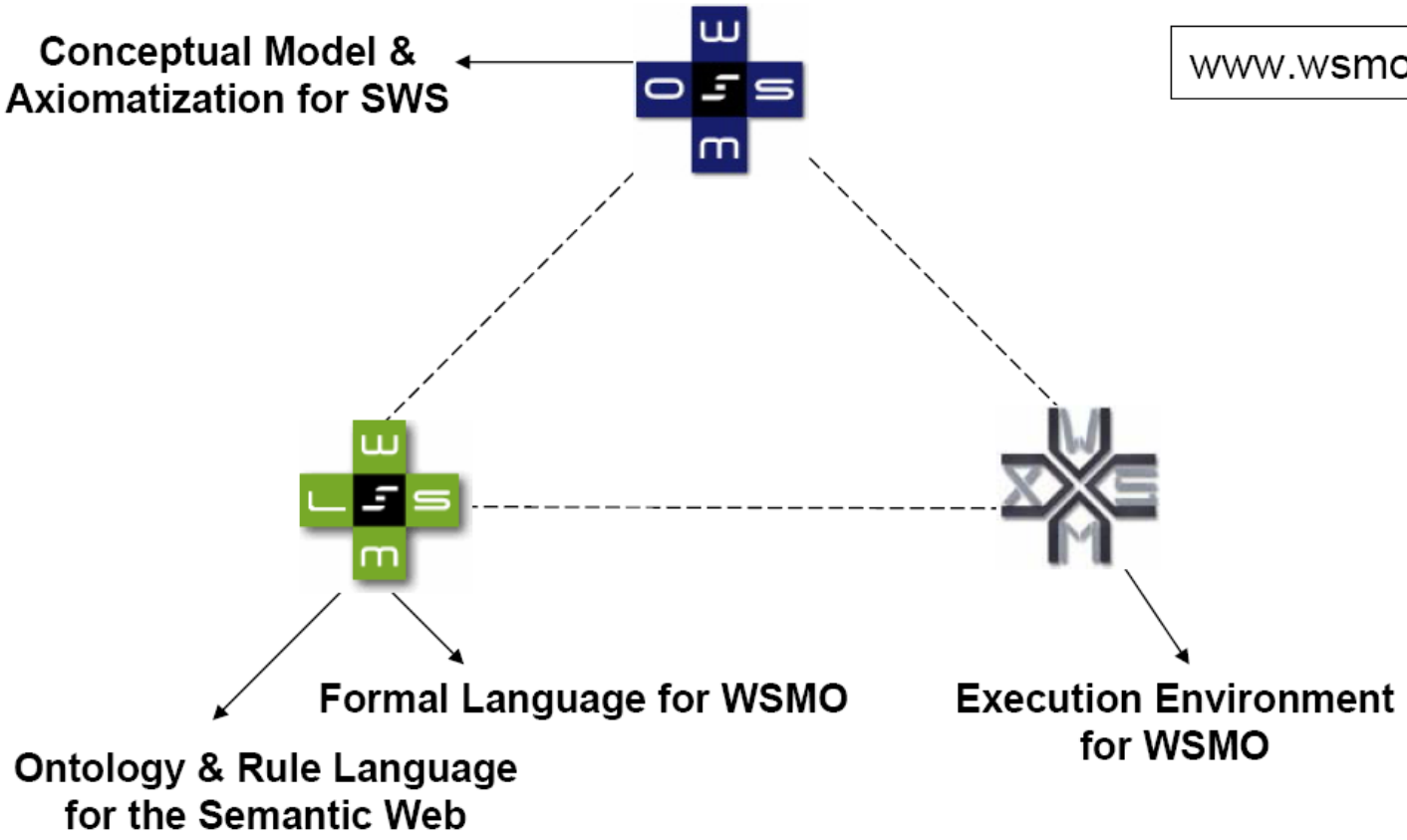
```
<process:AtomicProcess rdf:ID="LogIn">
  <process:hasInput rdf:resource="#AcctName"/>
  <process:hasInput rdf:resource="#Password"/>
  <process:hasOutput rdf:resource="#Ack"/>
  <process:hasPrecondition isMember(AccName)/>
  <process:hasResult>
     <process:Result>
        <process:inCondition>
            <expr:SWRL-Condition>
                correctLoginInfo(AccName,Password)
              </expr:SWRL-Condition>
        </process:inCondition>
        <process:withOutput rdf:resource="#Ack">
            <valueType rdr:resource="#LoginAcceptMsg">
        </process:withOutput>
        <process:hasEffect>
            <expr:SWRL-Condition>
              loggedIn(AccName,Password)
            </expr:SWRL-Condition>
        </process:hasEffect>
     </process:Result>
  </process:hasResult>
</process:AtomicProcess>
```

Mapping OWL-S / WSDL 1.1

- Operations correspond to Atomic Processes
- Input/Output messages correspond to Inputs/Outputs of processes



Web Service Modeling Ontology (WSMO)

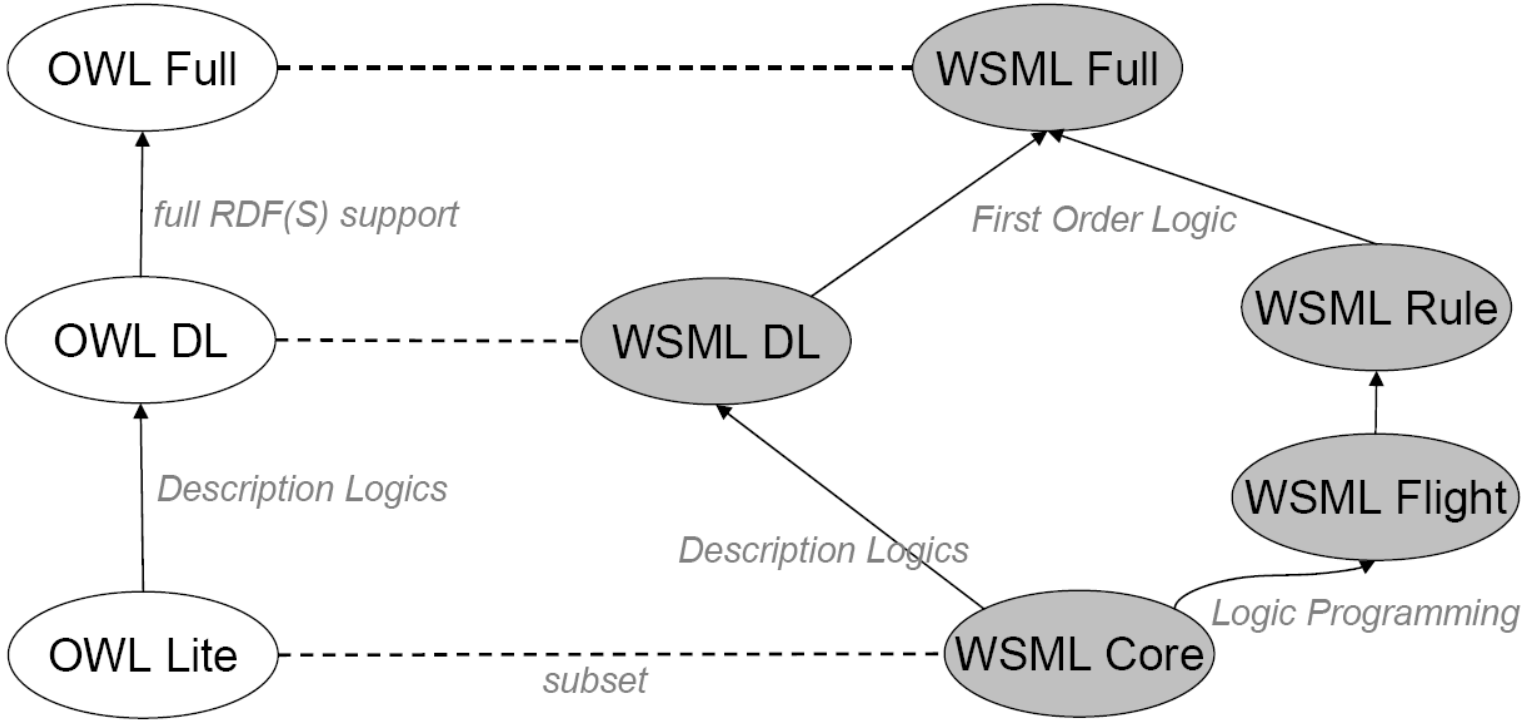


WSMO Conceptual Model

- WSMO is a meta-ontology for defining the components required for describing a Web service
- WSMO contains four elements…
  - …ontologies that provide the terminology used by other elements,
  - …goals that define the problems that should be solved by Web Services,
  - …Web Services descriptions that define various aspects of a Web Service,
  - …mediators which bypass interoperability problems.

Web Service Modeling Language

- WSML provides support for different logical fragments (description logics as well as logic programming)

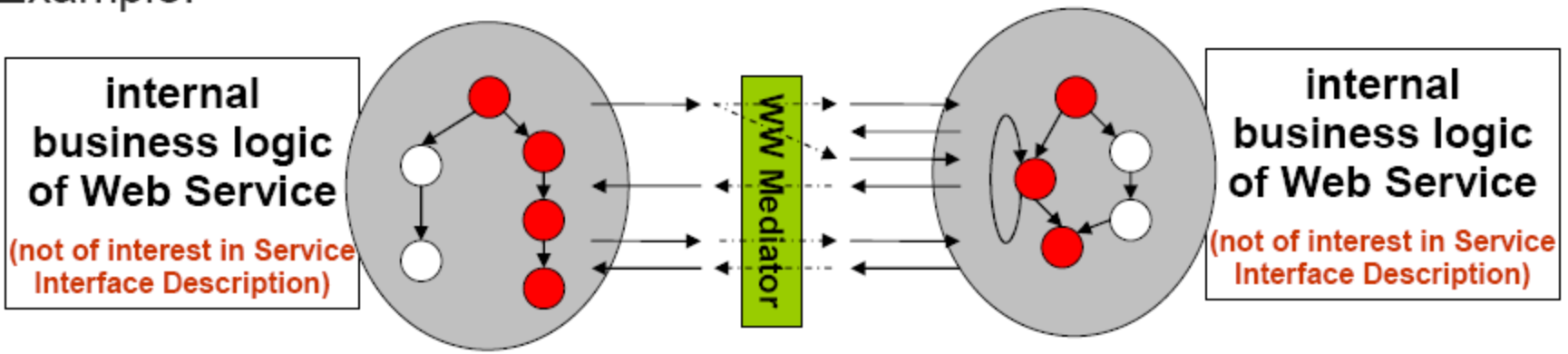


http://www.wsmo.org/TR/d17/sesa-tutorial/SESA-tutorial-day1.pdf

WSMO Mediators

- **oo**Mediator
  - Data Level Mediation
- **gg**Mediator
  - Data Level Mediation
  - Functional Level Mediation
- **wg**Mediator
  - Data Level Mediation
  - Functional Level Mediation
  - Process Level Mediation
- **ww**Mediator
  - Data Level Mediation
  - Functional Level Mediation
  - Process Level Mediation

Example:



http://www.sti-innsbruck.at/fileadmin/documents/SW07-08/SWE9large.pdf

- To some extend OWL-S and WSMO complement SAWSDL
  - OWL-S and WSMO can be used as semantic model in SAWSDL
  - SAWSDL can be used (partly) to ground OWL-S and WSMO descriptions (by connecting them to WSDL)

Overview

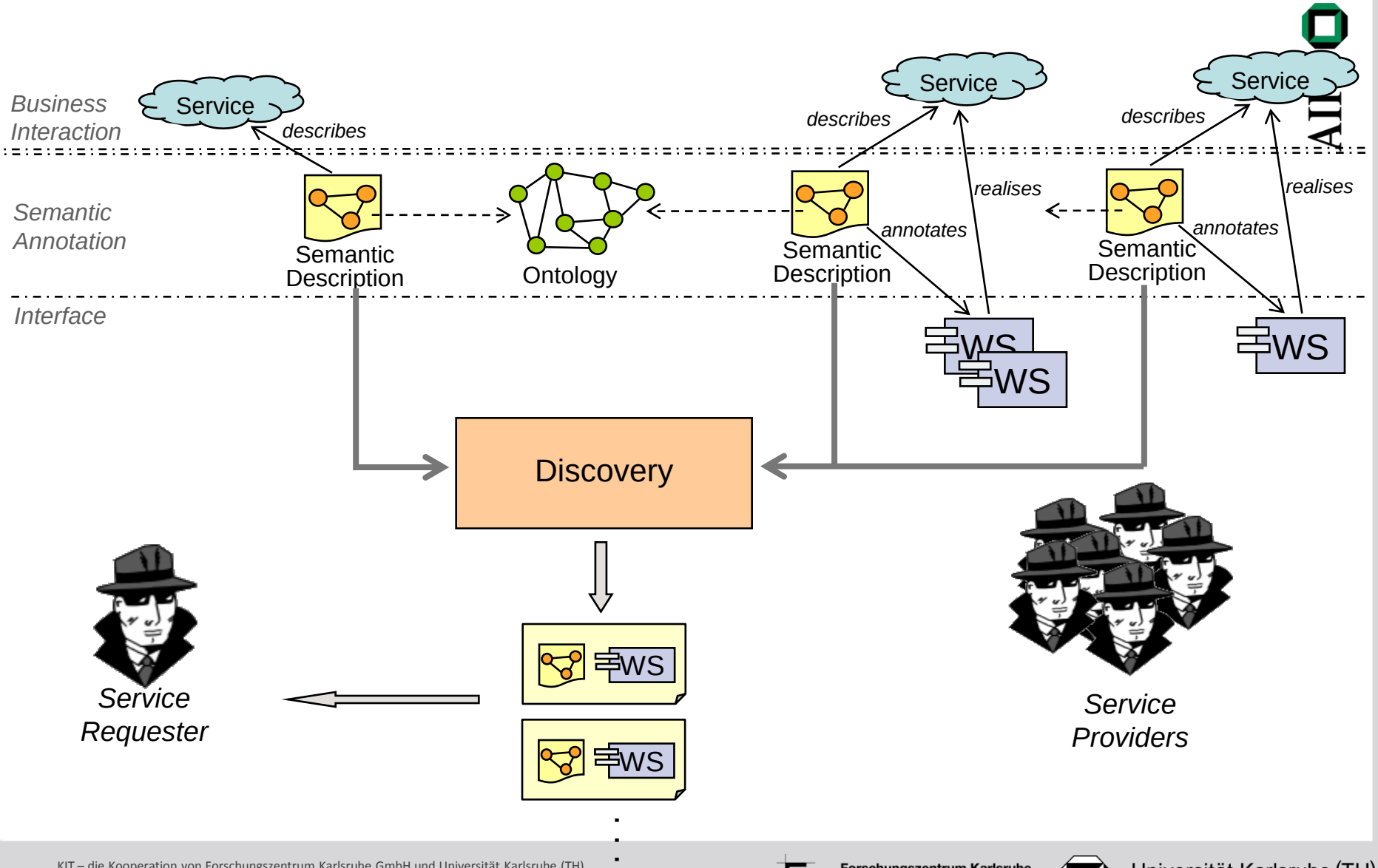
- Service Description Ontologies
- Service Discovery
  - Matching Semantic Service Descriptions in DL
  - Matching Behaviour under Open-World Assumption
- Service Selection

Further readings:

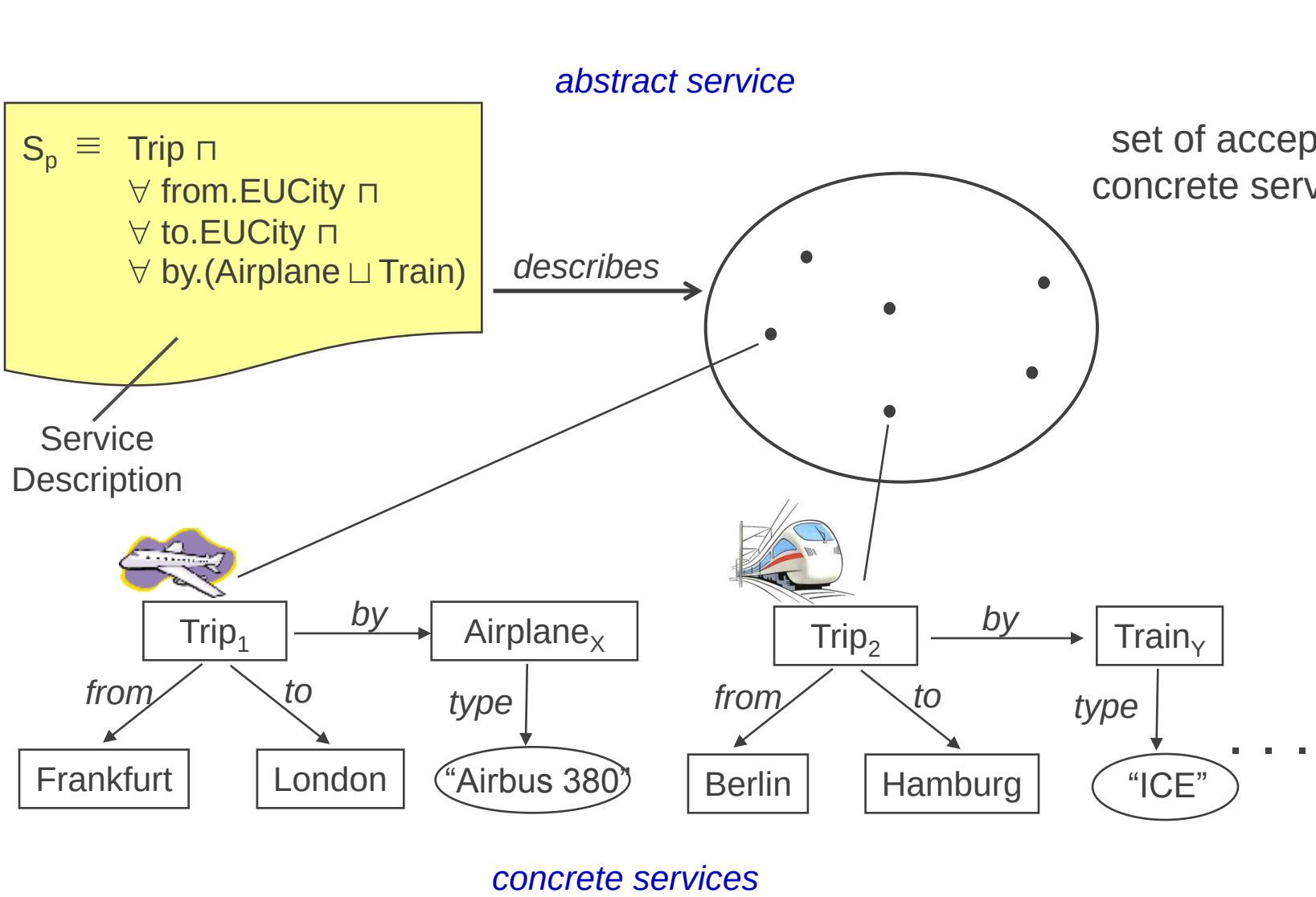
• Stephan Grimm, Intersection-Based Matchmaking for Semantic Web Service Discovery, Proceedings of the International Conference on Internet and Web Applications and Services ICIW'07. May 2007. IEEE Computer Society.

• Chapter 8: Discovery in Studer, Grimm, Abecker (Eds), Semantic Web Services - Concepts, Technologies, and Applications, Springer, 2007.

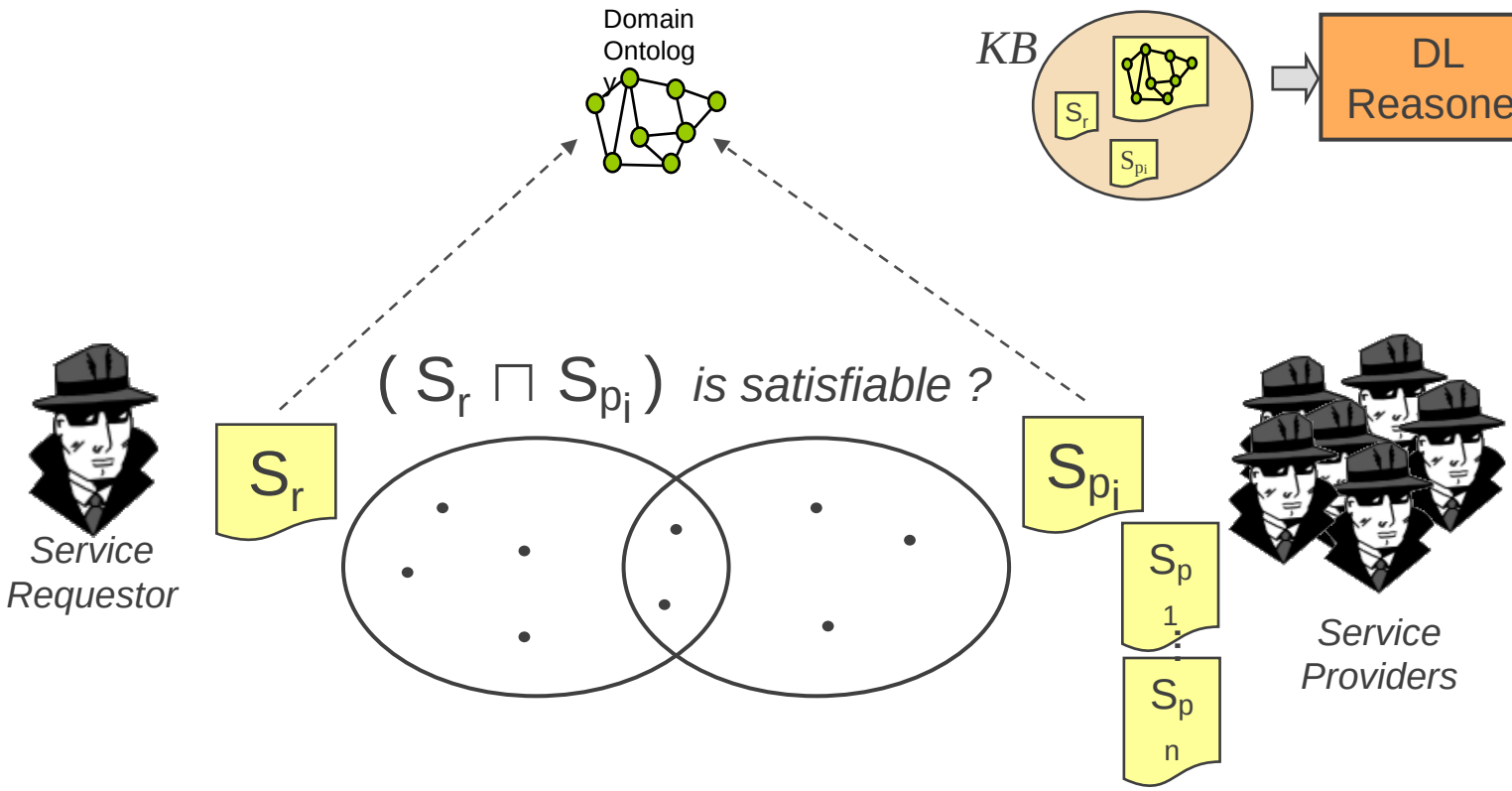
Service Discovery in the SW



Service Descriptions in DL



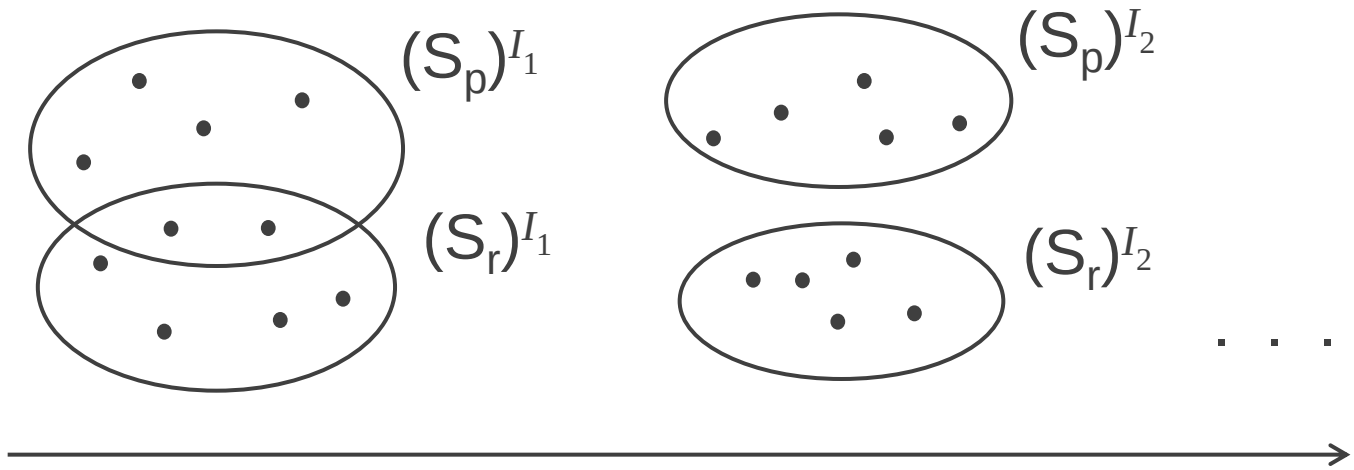
Matching DL Service Descriptions



- Matching Service Descriptions of Requesters and Providers
  - do they specify common concrete services?

DL Inference for Matching

- Satisfiability of Concept Conjunction
  - (Sr ⊓ Sp) is satisfiable w.r.t. KB



- $(S_r)^I \cap (S_p)^I \neq \emptyset$ in some model of KB
- Intuition:
  - incomplete knowledge issues can be resolved such that request and offer overlap

Matching Example (Travelling)

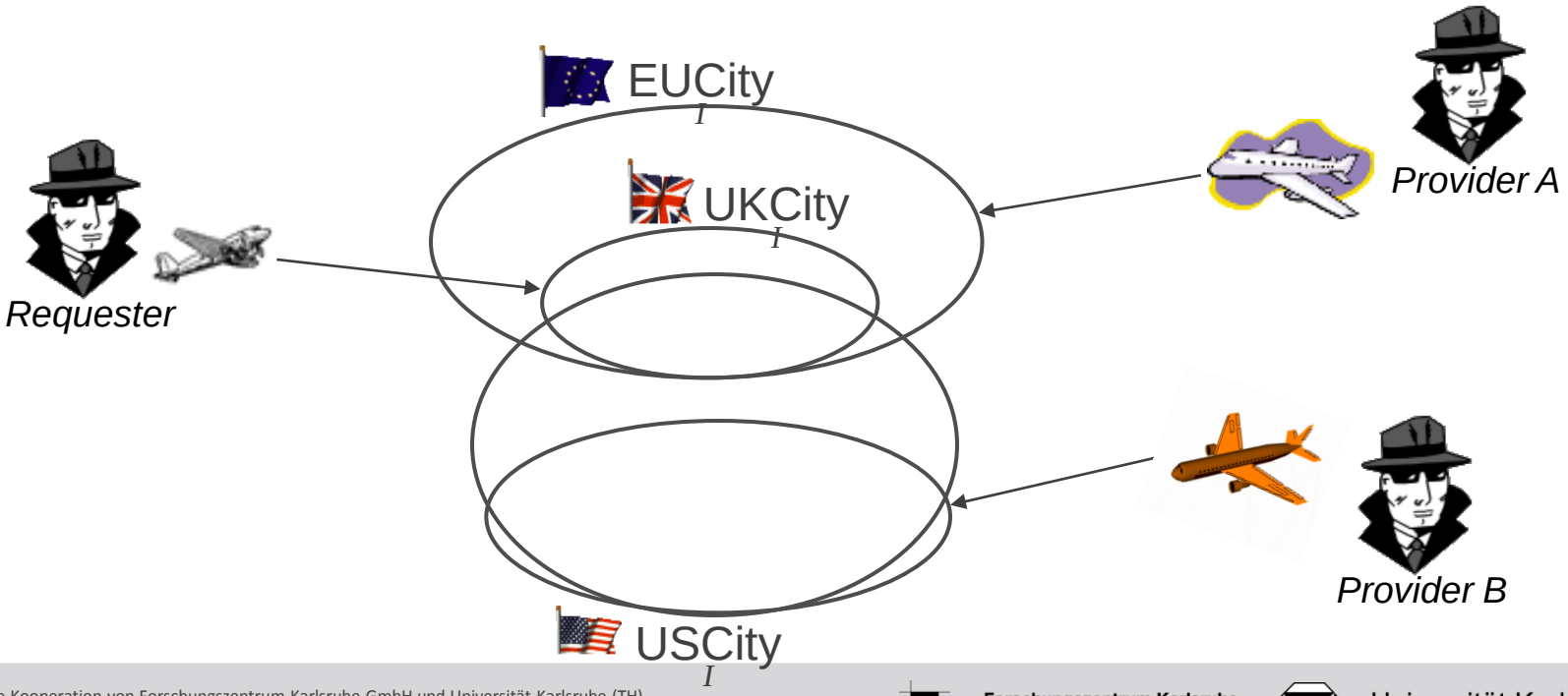
Domain Ontology

$$KB = \{\ UKCity \sqsubseteq EUCity,\ Flight \sqsubseteq \exists\, from\,.\top\ \}$$

request $S_r = Flight \sqcap \forall\, from\,.\,UKCity$

offer A $S_{pA} = Flight \sqcap \forall\, from\,.\,EUCity$

offer B $S_{pB} = Flight \sqcap \forall\, from\,.\,USCity$



Problems with Matching under OWA

- False positive matches
  - incomplete knowledge is likely to produce false positive matches
  - unknown concepts not restricted in the domain model
- Overspecification of the situation
  - the need to add more and more additional information
- Requirements on domain ontologies
  - the need to augment domain ontologies
  - difficult integration of available ontologies

Web Service Matching in WSMO

Exact Match:

$G, WS, O, M \models \forall x.\ (G(x) \Leftrightarrow WS(x))$

PlugIn Match:

$G, WS, O, M \models \forall x.\ (G(x) \Rightarrow WS(x))$

Subsumption Match:

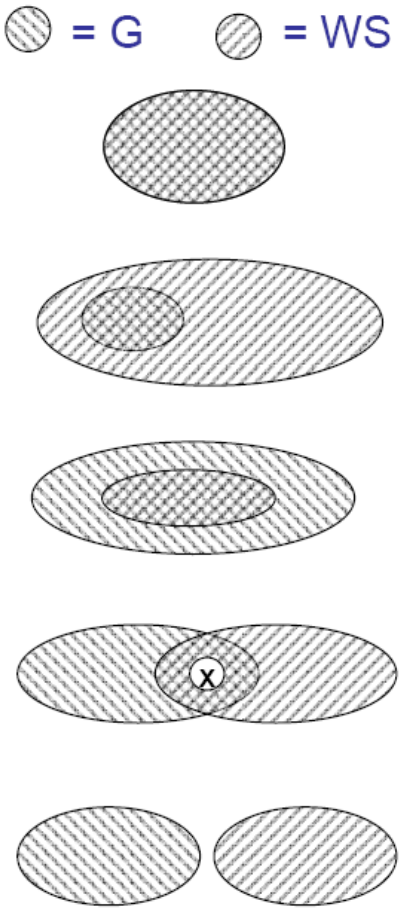
$G, WS, O, M \models \forall x.\ (G(x) \Leftarrow WS(x))$

Intersection Match:

$G, WS, O, M \models \exists x.\ (G(x) \wedge WS(x))$

Non Match:

$G, WS, O, M \models \neg\exists x.\ (G(x) \wedge WS(x))$



Keller, U.; Lara, R.; Polleres, A. (Eds): *WSMO Web Service Discovery*. WSML Working Draft D5.1, 12 Nov 2004.

Overview

- Service Description Ontologies
- Service Discovery
  - Matching Semantic Service Descriptions in DL
  - Matching Behaviour under Open-World Assumption

⇨ Service Selection

Further reading:

• *Steffen Lamparter, Anupriya Ankolekar, Stephan Grimm, Rudi Studer:* **Preference-based Selection of Highly Configurable Web Services,** In Proc. of the 16th Int. World Wide Web Conference (WWW'07). Banff, Canada, May 2007.

Approaches to WS selection

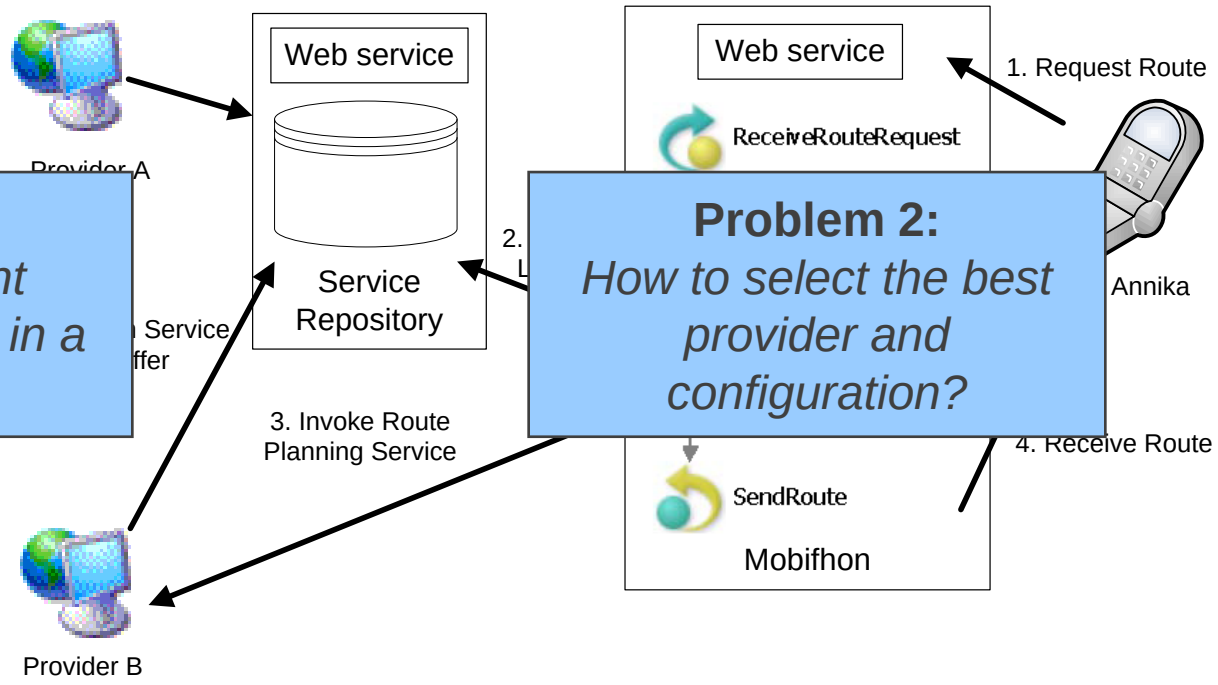
- Information retrieval techniques
  - Similarity/relevance between goal and service descriptions
- Social service selection
  - Based on community ratings (trust, reputation)
  - Collaborative filtering
  - …
- Preference-based service selection
  - Customer defines preferences on non-functional properties
  - Ranking of services according to preferences

Motivation

- Scenario: Dynamic selection of a route planning service

Provider A	C1	C2	C3
Coverage	EU	EU	EU
Response Time	10	20	5
Traffic Info	no	yes	yes
Indicated Attraction	no	Historic	Attractions
Price			

Provide		
Coverage	Ger	Ger
Response Time	10	20
Traffic Info	no	yes
Indicated Attraction	no	Historic Sites
Price	5	15



Problem 1:
How to represent configurable offers in a compact way?

Problem 2:
How to select the best provider and configuration?

Selection Model

- Set of configurations: $C = A_1 \times \cdots \times A_n$
  - Exponential size of configuration space: $|C| = \prod_{l=1}^{n} |A_l|$
  - **Problem:** Enumeration not feasible
- Define prices via Utility Functions: $U : C \rightarrow \mathbb{R}$
  - Request: scoring function (max. price) $F(c,k) = \sum_{l=1}^{n} w_l f_l(a_{le}, k)$
  - Offer: pricing function (min. price) $P(c) = p^{base} + \sum_{l=1}^{n} w_l p_l(a_{le})$
- Web Service Selection: Problem of finding the best provider from a set $\mathcal{P}$ for a given request.

Multi-attribute Matching Problem (MMP)

$$\underbrace{\underset{j \in \mathcal{P}}{\operatorname{argmax}} \left(\overbrace{\max_{c \in C_i \cap C_j} F_i(c,k) - P_j(c)}^{} \right)}_{}$$

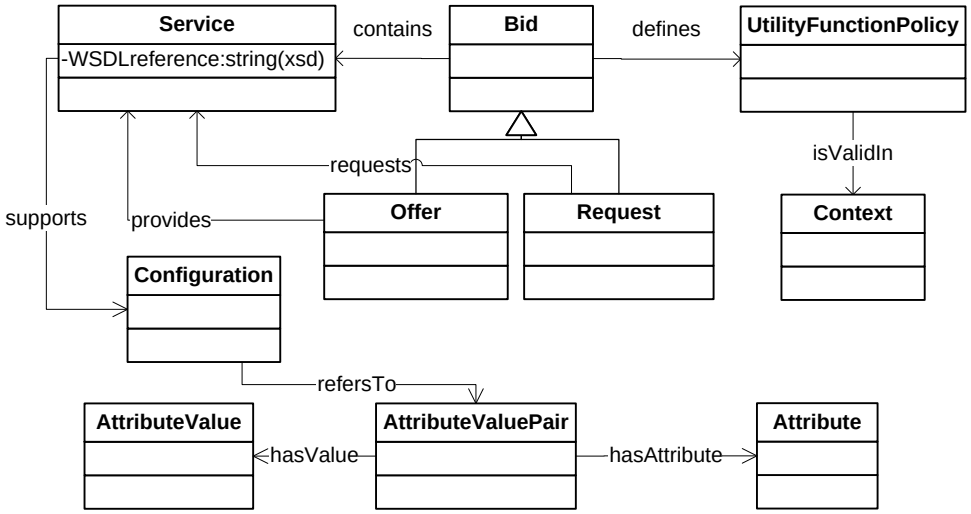
Local Allocation Problem (LAP)

Ontology-based WS Selection

- How to implement the selection model in a heterogeneous Web environment?
  - Support heterogeneous service descriptions
  - Provide flexibility (e.g. changes in the vocabulary)
  - Support declarative representation of utility functions
- Our approach:
  - Use ontologies for expressing offers and requests
    - Formalism: OWL-DL + DL-safe SWRL rules
    - Logical formalism features semantic matching
      - Matching requires reasoning (e.g. subsumption checking)
    - Rules: specification of utility functions and matching rules

Offer/Request Specification

- Ontology for expressing Web service offers and requests



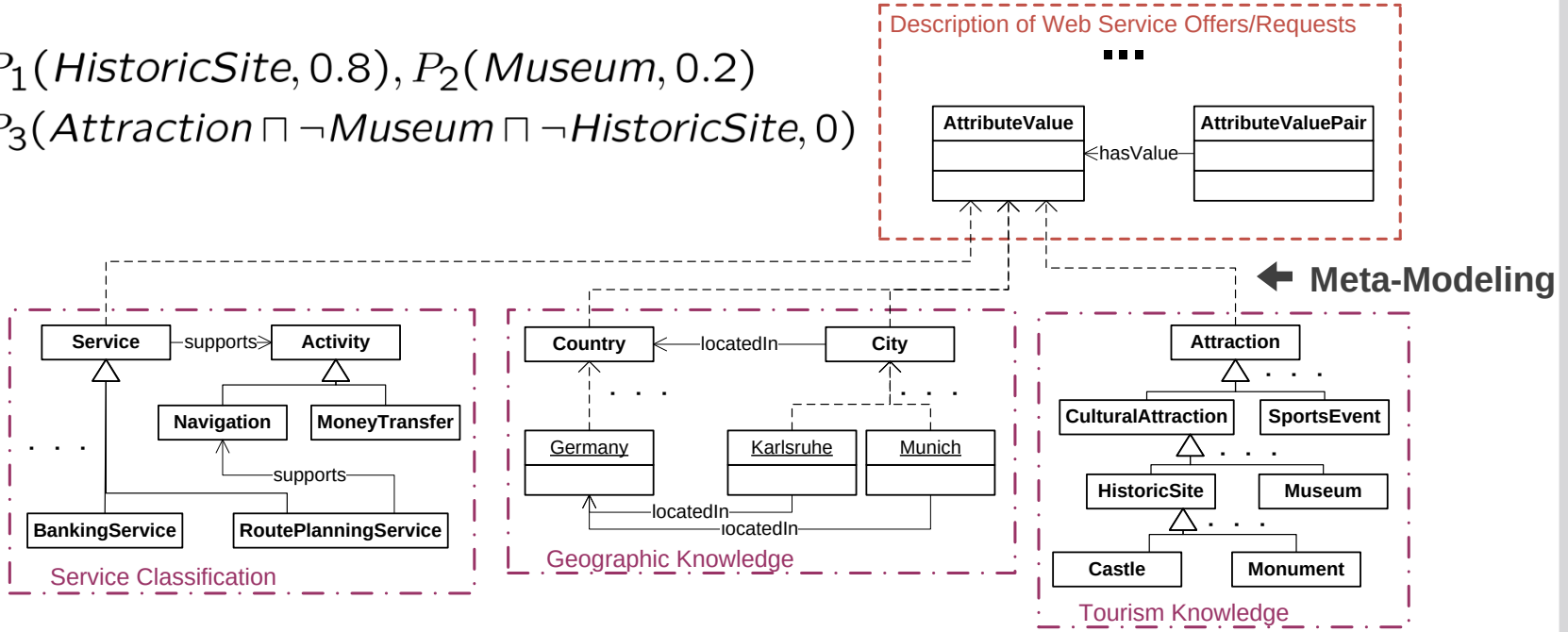
Reuse of service ontologies (OWL-S, WSMO) possible

  - Represent prices with Utility Function Policies
    - Expressed with OWL and DL-safe SWRL rules [Lamparter2006]
    - Reduces space complexity from $O(|O|n^{|A|})$ to $O(|O| + n|A|)$
    - Rule-based preference representation is well-known [Grosof2003], [Oldham2006]
  - Problem: Performance and scalability

Matching of Attribute Values

- **Issue 1:** Defining policies on sets of attribute values
  - Enumeration of all values inefficient
  - Attribute values may change or are not known
- Example:
  - requester scoring function

$$P_1(HistoricSite, 0.8), P_2(Museum, 0.2)$$
$$P_3(Attraction \sqcap \neg Museum \sqcap \neg HistoricSite, 0)$$



Matching of Attribute Values

- **Issue 2:** Matching of attribute values depends on attribute and on the ontology
  - For individuals operators like “=“,”<“,”>”, etc. can be used
  - Subsumes-Match

    $$\mathcal{O}_{tourism} \models \mathit{HistoricSite} \sqsubseteq \mathit{CulturalAttraction}$$

  - Many other DL-based matching approaches, e.g. [Paolucci2002], [diNoia2003]

- Define *matching rules* for attributes:

$$\mathit{match}(?P_1, ?P_2) \leftarrow \mathit{hasAttribute}(?P_1, \mathit{IndicatedAttraction}), \mathit{hasAttribute}(?P_2, \mathit{IndicatedAttraction}), \mathit{hasValue}(?P_1, ?V_1), \mathit{hasValue}(?P_2, ?V_2), \mathit{subsumes}(?V_1, ?V_2)$$

  - Keep selection algorithms domain-independent
  - Attribute hierarchies reduce number of matching rules

Web Service Selection

- Multi-attribute Matching Problem (MMP)
  - Find optimal configuration for one request/offer-pair
  - Calculation based on policies: $\max_{c \in C} F_i(c,k) - P_j(c)$

- Solving MMP using Variant [V1]:

$$\textbf{mmp}(?R,?O,?K,?U) \leftarrow \mathit{provides}(?O,?S_1), \mathit{supports}(?S_1,?C_1),$$
$$\mathit{requests}(?R,?S_2), \mathit{provides}(?S_2,?C_2), \mathit{compare}(?C_1,?C_2),$$
$$\mathit{price}(?O,?C_1,?P), \mathit{score}(?R,?C_2,?K,?S), \mathit{sub}(?S,?P,?U)$$

evaluates pricing policy

evaluates scoring policy

Invokes matching rules for all attributes

Web Service Selection

- Solving MMP using Variant [V2]:
  - Assumption: additive utility function policies
  - Each attribute can be optimized separately
  - Implementation using DL-safe rules (including matching rules)

$$\underbrace{\max_{c \in C} F_i(c,k)}_{|C| = \prod_{l=1}^{n} |A_l| \text{ iterations}} - P_j(c) = \underbrace{\sum_{l=1}^{n} \max_{a_{le} \in A_l} \left(w_{il} f_{il}(a_{le}, k) - w_{jl} p_{jl}(a_{le}) \right)}_{\sum_{l=1}^{n} |A_l| \text{ iterations}} - p_j^{base}$$

Web Service Selection

- Solving MMP using Variant [V3]:
  - Formulation as mixed integer linear program
  - Invocation of standard LP solver via built-in predicate

$$\max \quad \sum_{l=1}^{n} \sum_{e=1}^{|A_l|} (w_{il} f_i(a_{le}, k) - w_{jl} p_j(a_{le})) x_{le} - p_j^{base}$$

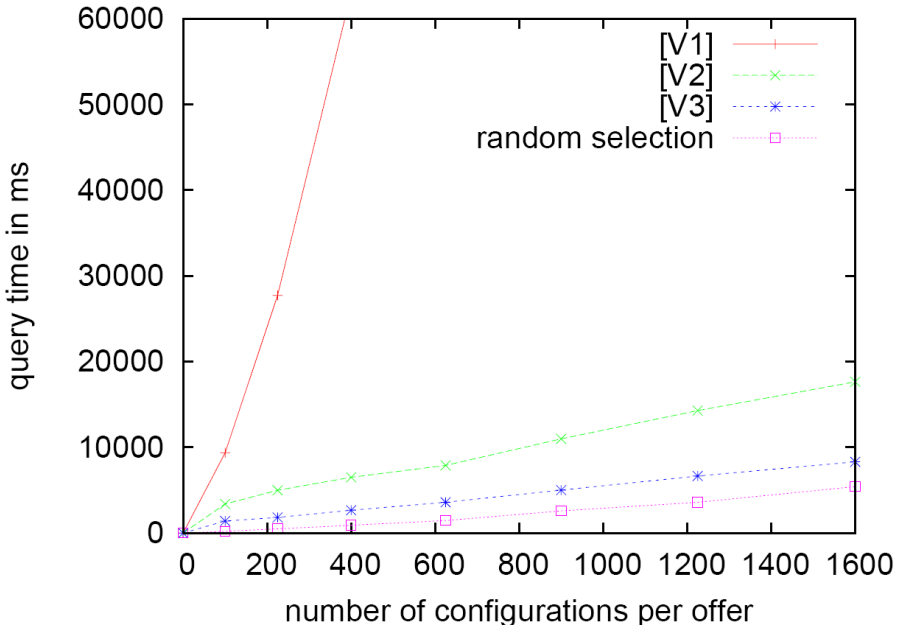
$$s.t. \quad \sum_{e=1}^{|A_l|} x_{le} = 1 \quad \forall\ 0 < l \leq n,$$

$$x_{le} \in \{0, 1\} \forall\ 0 < l \leq n, \forall\ 0 < e \leq |A_l|$$

- **Solving LAP:**
  - Execution of the MMP-rule for each request/offer-pair
  - Ranking according to utility

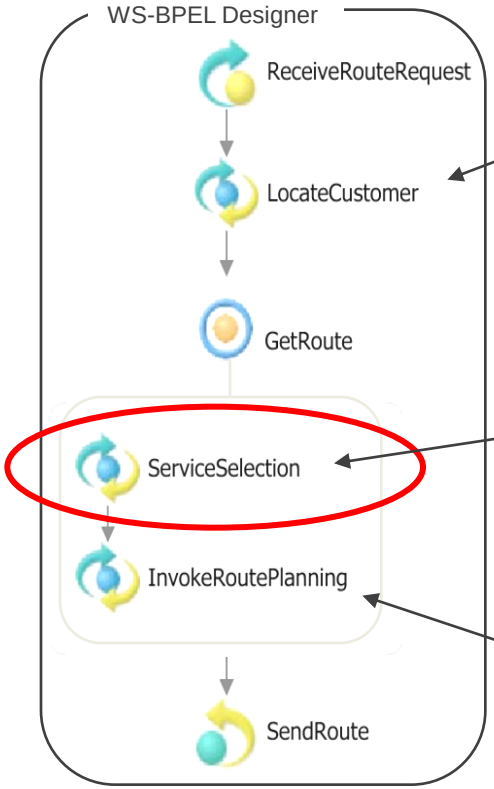
Evaluation

- Investigated algorithms:
  - [V1] No assumptions
  - [V2] Additive policies
  - [V3] Implementation with LP
  - Random selection
- Settings:
  - 1000 Offers in the repository
  - Varying number of configurations
  - Measure selection time
  - Average of 20 runs



- Results:
  - Non-additive policies major source of complexity (improvement [V1] → [V2])
  - Optimal strategy introduces an overhead below 2 seconds

Implementation




```
<invoke name="LocationCheck" partnerLink="LocationServicePLT"                    8
    portType="ns1:LocationService" operation="executeQuery"                      9
    inputVariable="clientID" outputVariable="location"/>                         10
<assign name="Assign_Query">                                                     11
    <copy> <from expression='"concat(string(\"SELECT ?O , ?U WHERE {             12
            EVALUATE mmp(ns1:RequestOperatorX, ?O,\" ),                          13
            bpws:getVariableData('location','Country'),string(\",?U ) . }\"))"'/>14
        <to variable="requestQuery" part="queryMessage"/>                        15
    </copy>                                                                      16
</assign>                                                                        17
<invoke name="ServiceInvoke" partnerLink="SelectionServicePLT"                   18
    portType="ns1:SelectionService" operation="executeQuery"                     19
    inputVariable="requestQuery" outputVariable="responseQuery"/>                20
<assign name="Assign_Port">                                                      21
    <copy> <from variable="responseQuery" part="queryResult"/>                   22
        <to partnerLink="RoutePlanner"/>                                         23
    </copy>                                                                      24
</assign>                                                                        25
<invoke name="RoutePlanningInvoke" partnerLink="RoutePlanner"                    26
    portType="ns1:RoutePlanningService" operation="requestRoute"                 27
    inputVariable="routeRequest" outputVariable="responseRoute"/>                28
                                                                                 29
```

Next lecture

- Invited talk: York Sure (SAP)
  - Topic: Internet of Services
  - Wednesday (June 18, 2008)

- Übung:
  - Montag, 23. Juni 2008
  - 11:30 – 13:00 Uhr

Acknowledgements

- ESSI WSMO working group
  - http://www.wsmo.org/
- OWL-S coalition
  - http://www.daml.org/services/owl-s/
- SAWSDL W3C working group
  - http://www.w3.org/2002/ws/sawsdl/

End