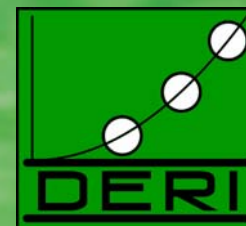


Entwicklung einer Ontologie für Semantische Web Services

Uwe Keller
uwe.keller@deri.org

ASG-Week: Industrie Workshop
Koblenz, 6. Juli 2006

- **Teil I: Einführung**
 - Semantik and Services
 - Ontologien
 - Web Services and Semantic Web Services
- **Teil II: Semantic Web Service Frameworks (aka. Ontologien)**
 - **WSMO**
 - *OWL-S*
 - *SWSF*
 - *WSDL-S*
- **Teil III: Beispiel in WSMO**
 - Domain-Ontologien
 - Funktionale Beschreibung von Semantic Web Services
 - Goal Beschreibung
 - Funktionales Matching zwischen Goal und Web Services



Teil I:

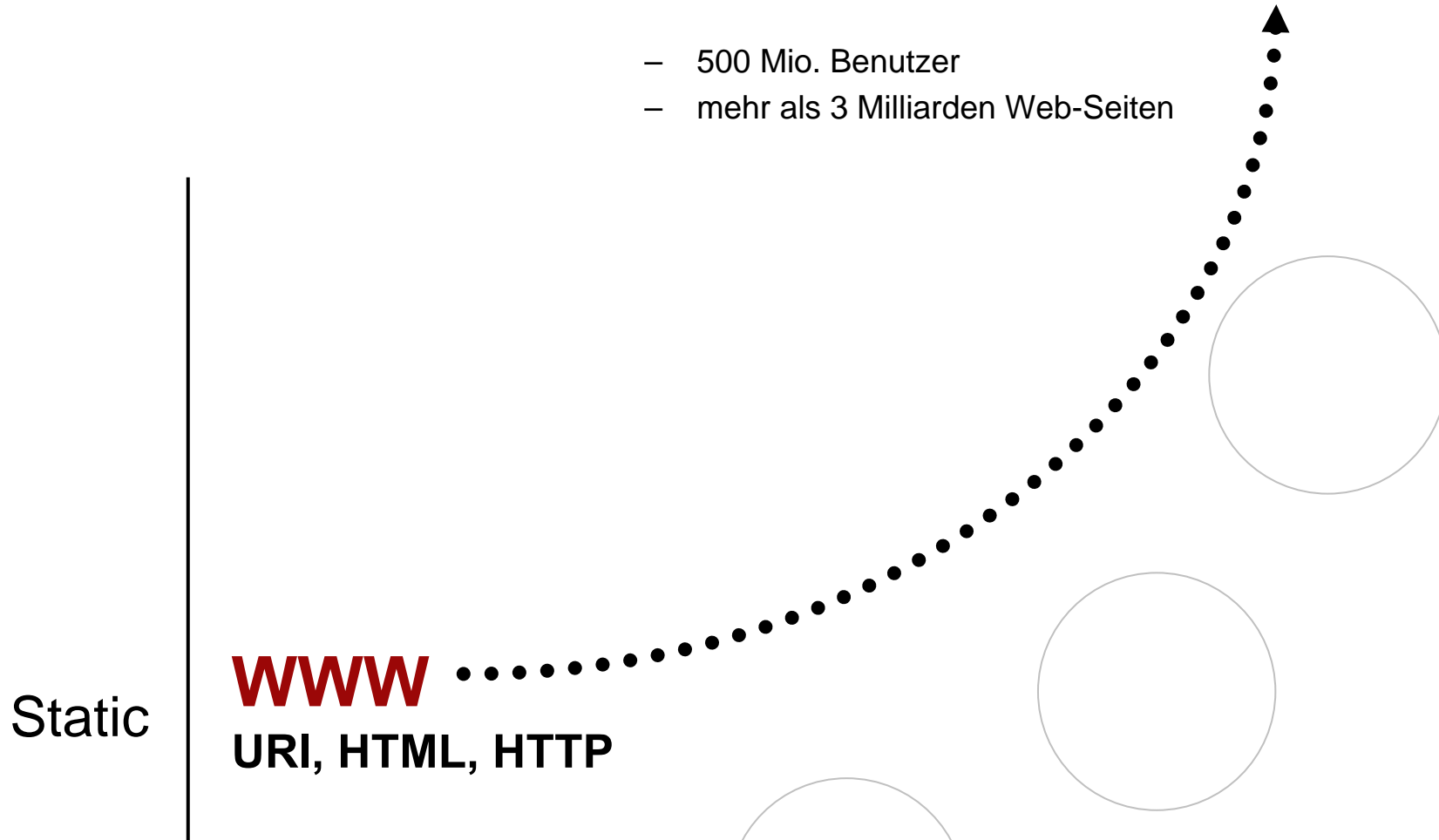
Einführung

*"Semantic differences remain the primary roadblock to smooth application integration, one which Web Services alone won't over-come. Until someone finds a way for applications to understand each other, the effect of Web services technology will be fairly limited. When I pass customer data across [the Web] in a certain format using a Web Services interface, the receiving program has to know what that format is. You have to agree on what the business objects look like. And no one has come up with a feasible way to work that out yet -- **not Oracle, and not its competitors...**"*

Oracle Chairman and CEO Larry Ellison

- Integration von Applikationen und Datenbeständen = **Kommunikationsakt**
- Erfolgreiche Kommunikation erfordert **gemeinsames Verständnis** von Sachverhalten
- „Geteiltest Verständnis“ findet auf der **Bedeutungsebene** statt (nicht auf Ebene der Datenrepräsentation)

- Akutes Beispiel: **Dieser Vortrag**, wie der Sprecher gestern morgen herausfand
 - Gemeinsam verwendetes Konzept: Titel
 - „Entwicklung einer Ontologie für Semantische Web Services“
 - Unterschiedliche Interpretation bei Beteiligten
 - Sprecher: „Entwicklung einer **Ontologie zur Beschreibung von Semantischen Web Services**“
 - Autor der Tutorial Beschreibung: „**Entwicklung einer Ontologie** im Kontext Semantischer Web Services“
 - **Effekt? Auweia! Das gibt hoffentlich keine Haue :)**



Probleme bei automatischer Informationsverarbeitung:
Das Web für Menschen gemacht, nicht für Maschinen

- Auffinden
- Extraktion
- Repräsentation
- Interpretation
- Zusammenfügen
- Wartung

⇒ Semantic Web als ein Web für Maschinen

Static

WWW
URI, HTML, HTTP

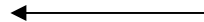


Semantic Web
RDF, RDF(S), OWL

Dynamic

Web Services

UDDI, WSDL, SOAP



Der nächste Schritt: **Web Services (WS)**

Ermöglichen verteilte und geteilte Berechnungen über das Internet

Aus einem Web von statischen Ressourcen (Datenelementen) wird ein Web das dynamische/aktive/interaktive Ressourcen (WS) zur Verfügung stellt.

Static

WWW

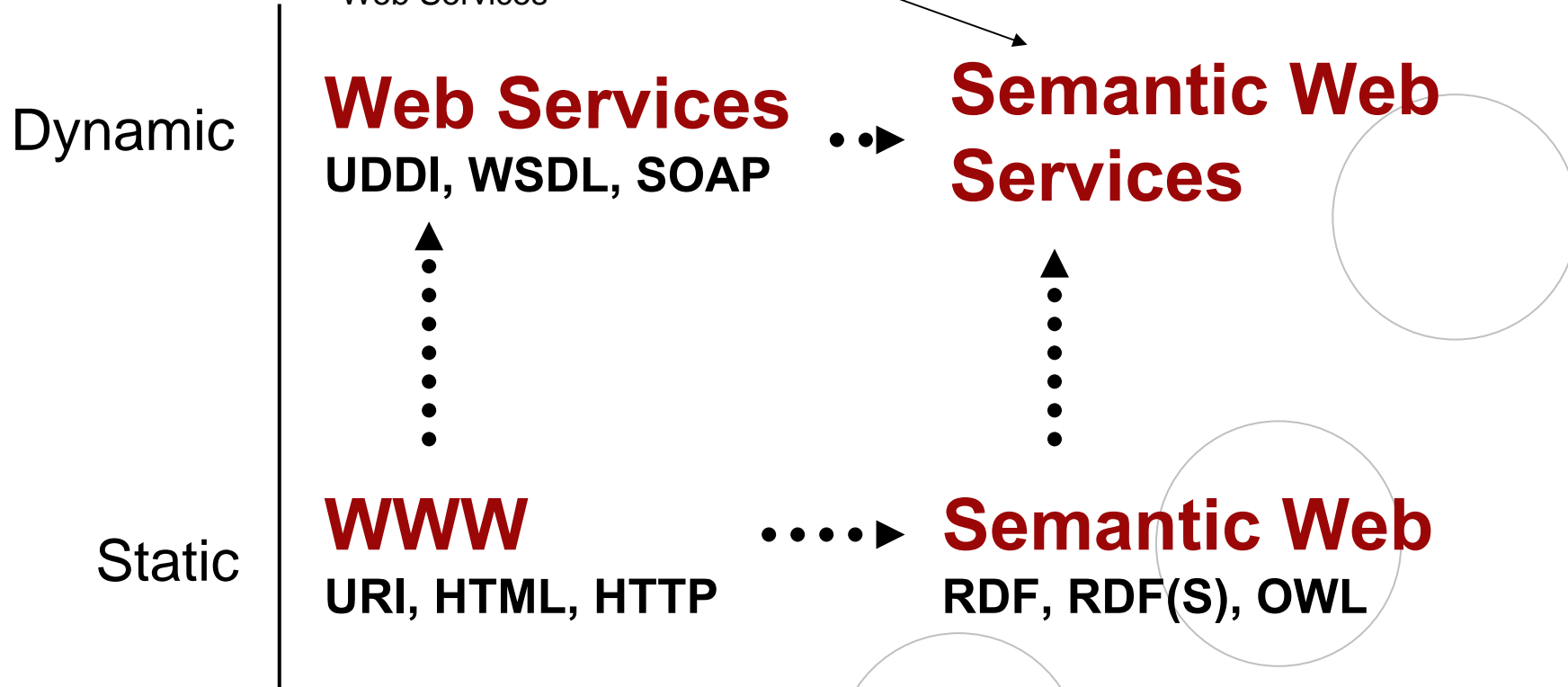
URI, HTML, HTTP



Semantic Web

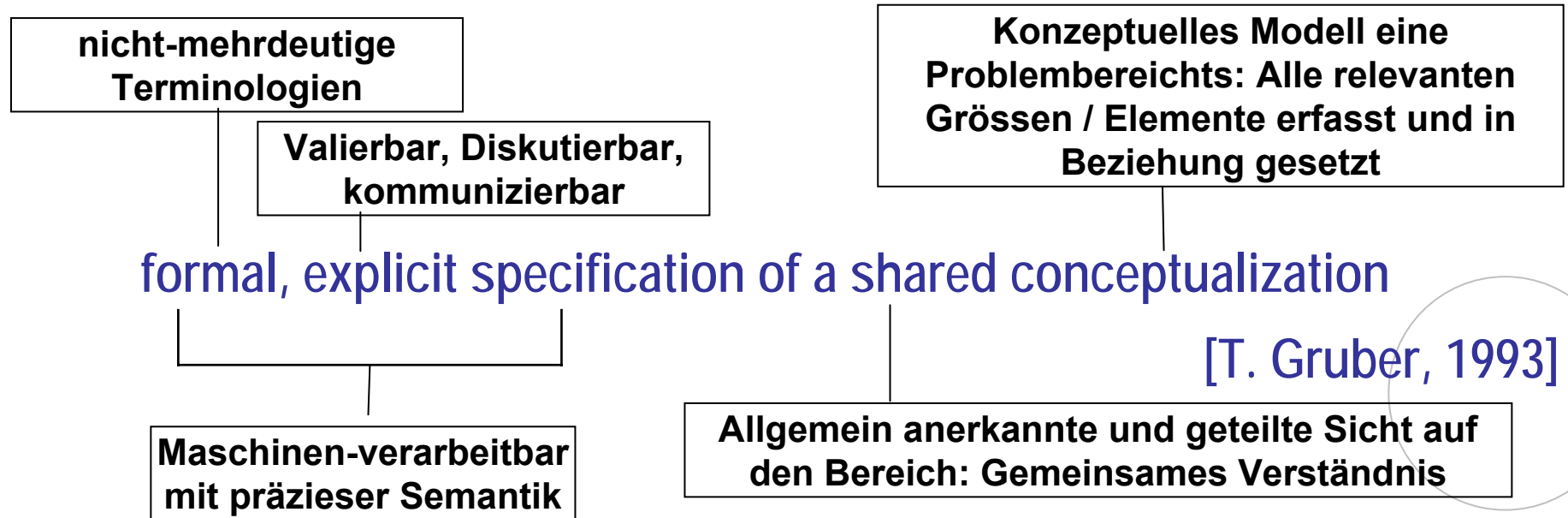
RDF, RDF(S), OWL

Web Service Technologien sind für Menschen gemacht
Kombination mit Semantik & Semantik Web Technologien:
Automatische Unterstützung bei der Verwendung / Arbeit mit verfügbaren
Web Services



- Nächste Generation des WWW
- Information hat maschinen-verarbeitbare und maschinen-verständliche Semantik (semantik-bezogene Metadaten)
- Kein separates Web, sondern Erweiterung des Bestehenden
- **Ontologien** als unterliegende Basis-Technologie, die gemeinsames Verständnis und Integration von Daten (Wissen) ermöglichen
- **Neuerdings auch: Das Web als Ort für dynamische Interaktion: Web Services und die Anwendung von Semantik auf WS**
 - Web Services = abstrakte (komplexe) Resource, die man semantisch beschreibt
 - WS Technologien sind dokument-orientiert (mit anwendungsspez. Nutzlast):
Nutze sem. Annotation solcher Informationselemente

- **Gemeinsames Verständnis eines Diskurs/Problembereichs**
 - Finanzen, Telekom, Tourismus, ...
 - Allgemeine (cross-domain) vs. Anwendungsspezifische Ontologien
- Eine Ontologie verkörpert **einen spezielle Sicht** auf den Problembereich **innerhalb einer Gruppe / Community** kooperierender Aktoren (Menschen oder Maschinen)
- Oft in Form von Konzepten, deren Definitionen und deren Beziehungen zueinander
- Dies wird auch „conceptualization“ genannt.



- **Ontologien stellen Werte an sich dar (*Model-getriebene SW-Entw.*)**
 - Müssen erstellt und gewartet werden (mögl. substantielle Kosten)
 - Sind dokumentiert und können kommuniziert werden
 - Können zu Ontologie-Netzwerken verflochten werden (innerhalb und zwischen Problembereichen)

Concept

conceptual entity of the domain

Property

attribute describing a concept

Relation

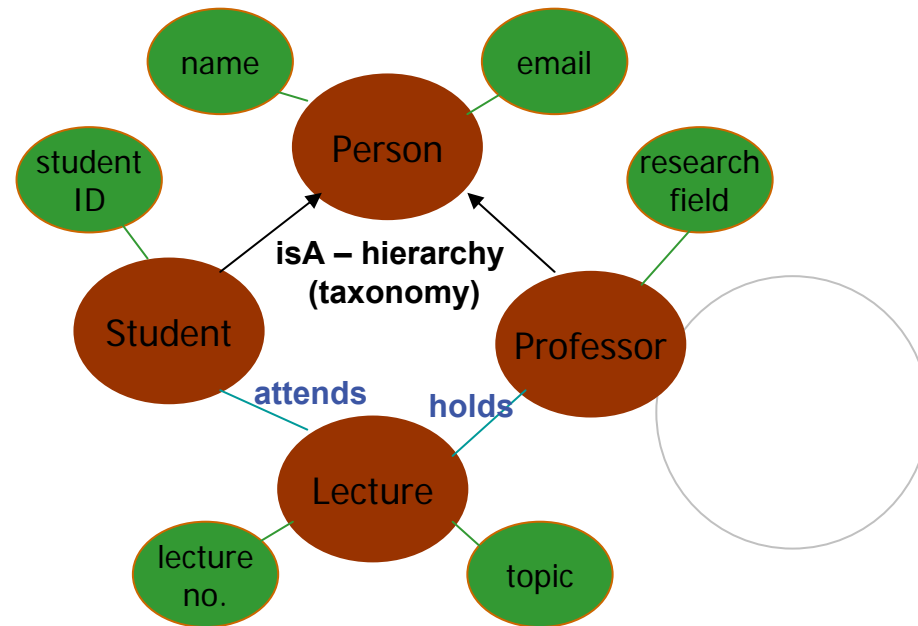
relationship between concepts or properties

Axiom

coherency description between Concepts / Properties / Relations via logical expressions

Instance

individual in the domain



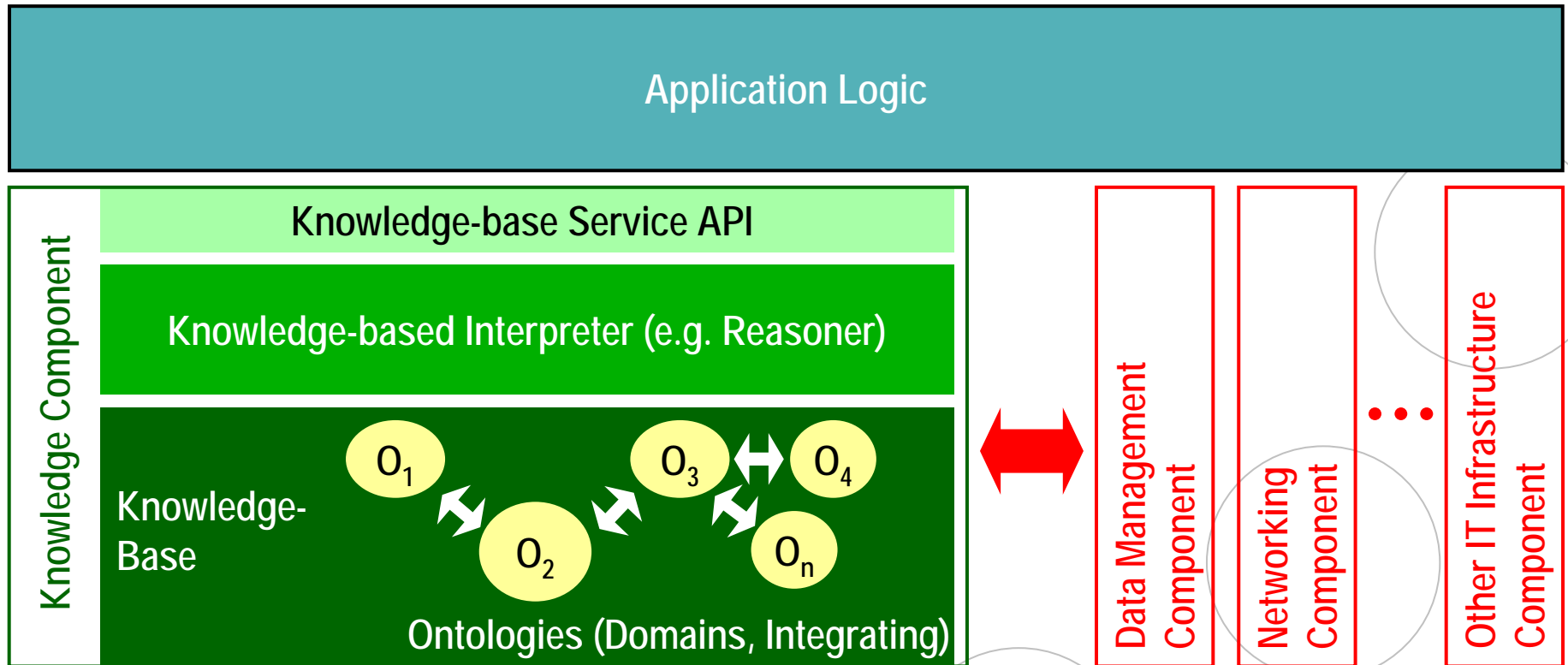
`holds(Professor, Lecture) =>`
`Lecture.topic = Professor.researchField`

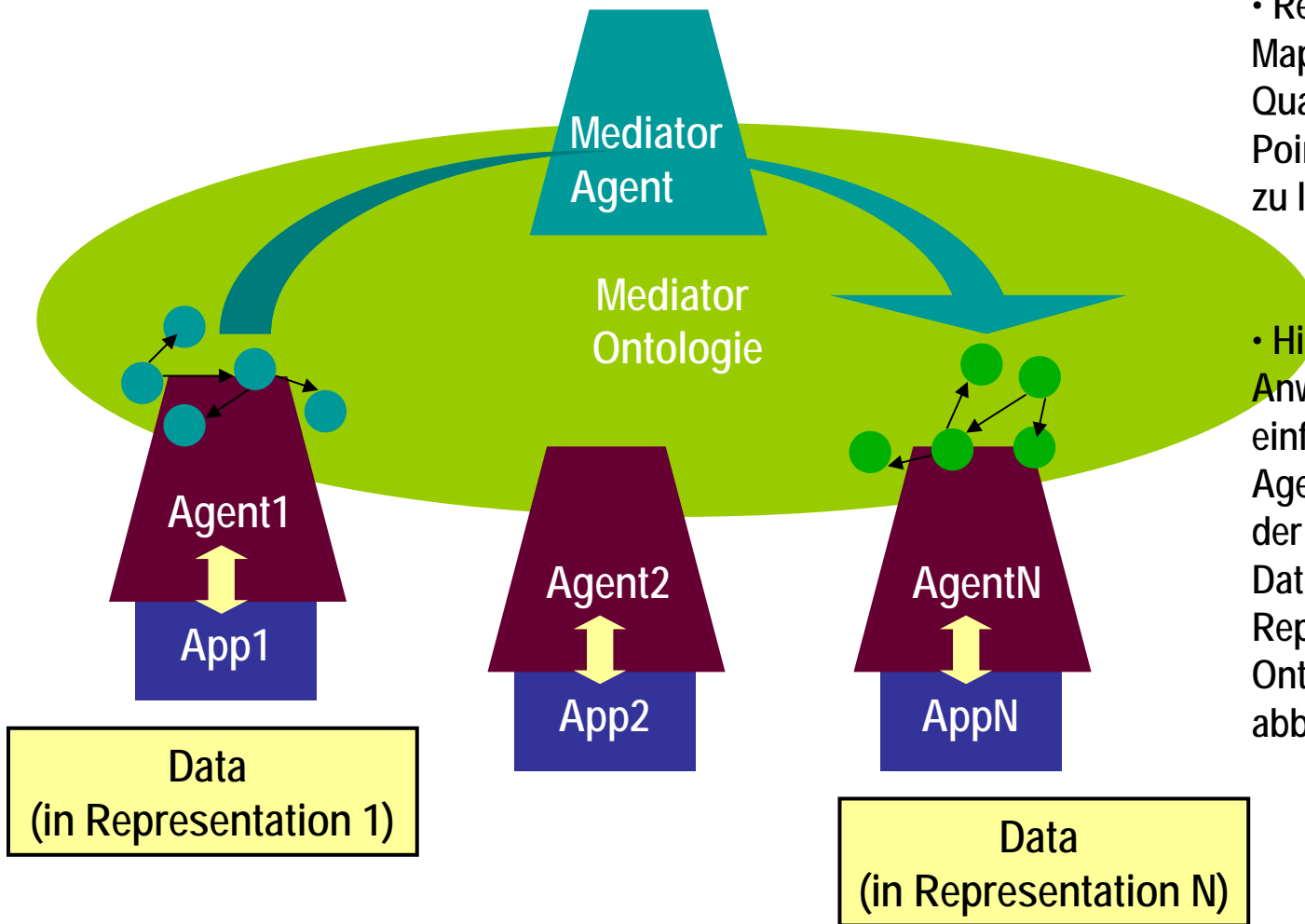
Ann memberOf student
name = Ann Lee
studentID = 12345

- Grad der Formalisierung
 - Informell, semi-informell, semi-formal, hochgradig formal
 - Beachte: Informelle Ontologien sind nicht für Maschinen verarbeitbar (aber dennoch nützlich: Interaktion zwischen Menschen)
- Grad der Detaillierung
 - Gemeinsam-verwendetes Vokabular (e.g. Dublin Core)
 - Einfache Taxonomien (e.g. eClassOWL)
 - Komplexe definitionen von Beziehungen, ausgiebige Verwendung von Axiomen (e.g CYC, PSL-Ontology, TOVE)

- **Ontologie (Beschreibungs-)Sprachen:**
 - Ausdruckstärke
 - Maschinen-Zugriff auf das dargestellte Wissen: Unterstützung von Reasoning
 - Web-compliance: Integration mit bestehenden Technologien im Internet
- **Ontologie Reasoning:**
 - Skalierbarer Umgang mit Wissen (bspw. ontologie-basierte Informations Integration)
 - Unterschiedliche Aspekte: Verteilung, Paraconsistent, Unsicheres Wissen, ...
- **Ontologie Management:**
 - (kollaboratives) Erstellen, Editieren und Untersuchen von Ontologien
 - Speicherung und Auffinden / Abruf von Ontologien
 - Versionierung und Evolutions-Unterstützung für Ontologien
 - Lernen von Ontologien
- **Ontologie Integration:**
 - Ontologie mapping, alignment, merging
 - Maschinen-unterstützung dieser Integrationsaufgaben

Ontologien als eine eigenständige Komponente eines Software-Systems



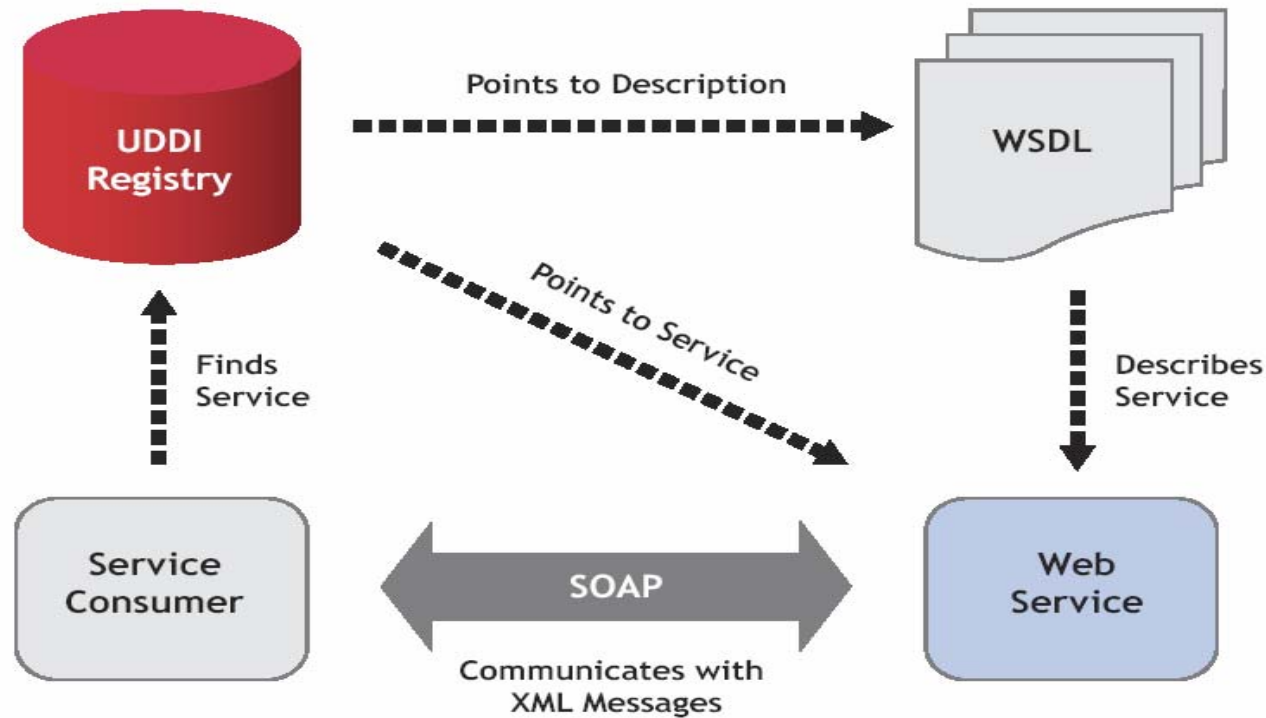


- Reduktion der Anzahl von Mappings von Quadratisch vielen (Point-to-Point-Lösung) zu linear vielen

- Hinzufügen neuer Quellen / Anwendungen einfach (Definition eines Agenten, der von der Anwendungsspezifischen Daten-Repräsentation auf die Ontologische Ebene abbildet und umgekehrt)

- Lose gekoppelte, wiederverwendbare Software-Komponenten
 - Kapseln diskrete Funktionalität
 - Verteilt
 - Über Standard Internet Protokolle ansprech-/zugreiff-/nutzbar
 - Erweitern das bestehende Internet um eine weitere Schicht: Von reinen Informationen zu Services
- => Bilden die **Basis-Technologie für Service-oriented Architectures (SOA)** über das Web

Web-basierte SOA als "neues" System Design Paradigma



Für Nutzung durch den Menschen entwickelt. Maschinelle Unterstützung schwer!
Durch Verbreitung von SOA (insbesondere über Grenzen organisatorischer Einheiten hinweg), wird maschinelle Unterstützung wünschenswert.

Semantic Web Technology

- ermöglicht maschinen-unterstützte Daten Interpretation
- Ontologien als konzeptuelles Datenmodell

+

Web Service Technology

Automatisierung beim Auffinden, Auswahl, Komposition, und web-basierter Ausführung von Web Services

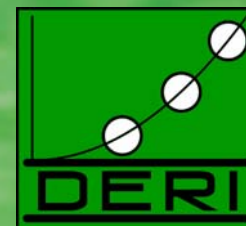
=> Semantic Web Services als integrierte Lösung zur Umsetzung der Vision der nächsten Generation des Web und von effektiven Service-oriented Architekturen

Grundlegende Idee:

- Definiere ein umfassendes Beschreibungs-Framework zur Spezifikation von Web Services und verwandter Aspekte (**Web Service Description Ontologien**)
- Verwende Ontologien als unterliegendes Datenmodell um maschinen-unterstützte Daten Interpretation zu ermöglichen (**Semantic Web Aspekt**)
- Definiere Semantik-basierte Technologien zur Automatisierung von Aspekten der Web Service Benutzung (**Web Service Aspekt**)

1. **Deployment** Erzeugung und Veröffentlichung von Web service Beschreibungen
2. **Discovery** Auffinden von nutzbaren Web Services für einen Benutzer (bzgl. einer Goal-Beschreibung)
3. **Composition** Kombination von WS zur Realisierung eines Benutzer-Ziels
4. **Selection** Auswahl des geeignetsten WS aus den anwendbaren WS für einen Benutzer
5. **Mediation** Beseitigung von Heterogenität (Daten, Protocol, Prozess) die Inter-operation verhindert
6. **Execution** Aufruf eines WS basierend auf technischen Beschreibungen

- **Monitoring** Kontrolle / Beobachtung der Ausführung
- **Compensation** Transaktions-Unterstützung und Vermeidung ungewünschter Effekte
- **Replacement** Ermöglichung des Austausch von WS durch entsprechende andere
- **Auditing** Überprüfung auf Korrektheit der Ausführung entsprechend der Erwartung (Spezifikation)



Teil II:

Semantic Web Service Frameworks

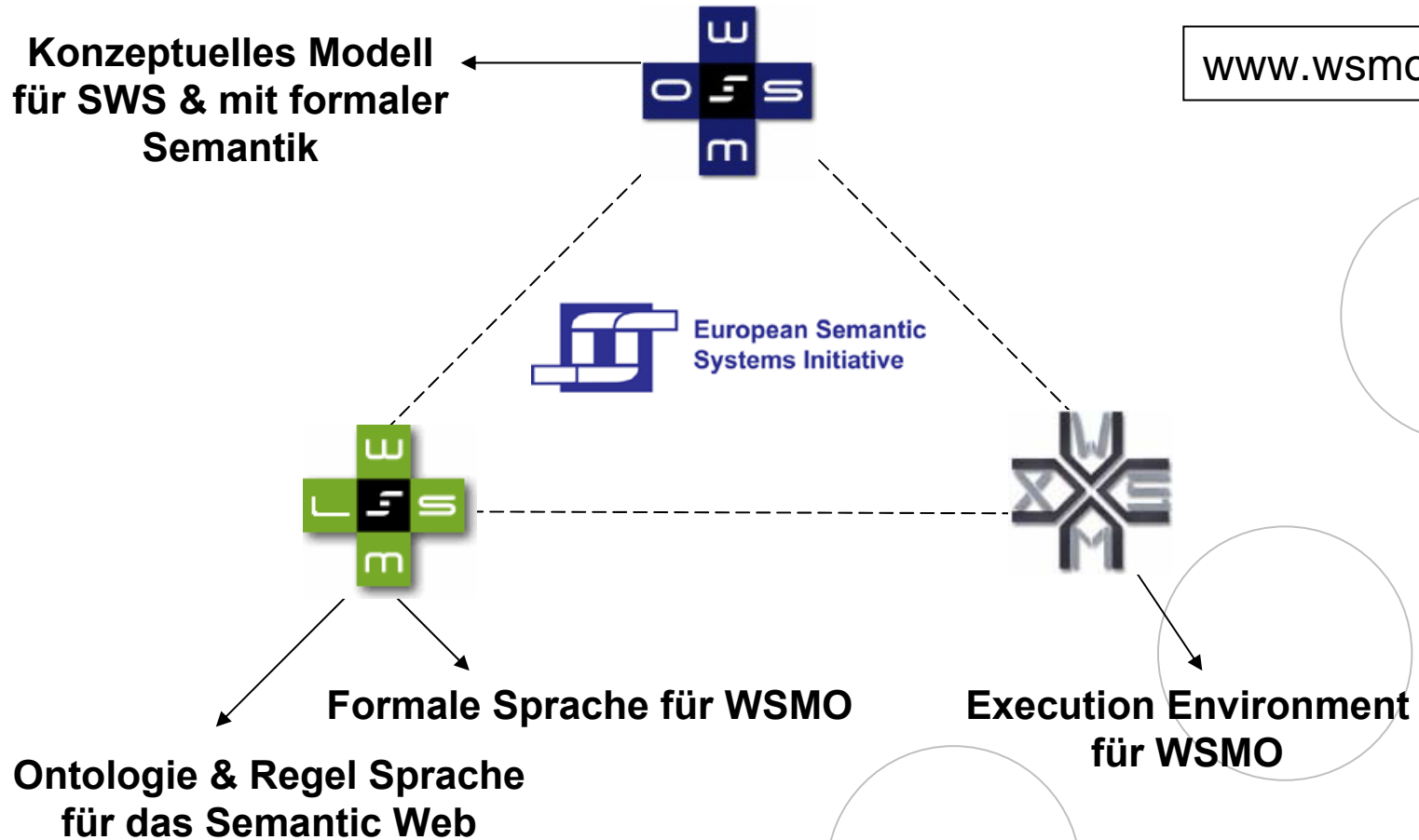
- **Ein Frameworks für Semantic Web Services muss**

- Alle relevanten Aspekte von WS darstellen, die ein Automatisierung in der Web Service Benutzung ermöglichen
- Ein konzeptuelles Modell dieser Aspekte und dessen Semantik festlegen
- Eine formale Sprache für Semantische Beschreibungen von SWS basierend auf dem konzeptuellen Modell bieten

- **Ansätze (W3C Member Submissions)**

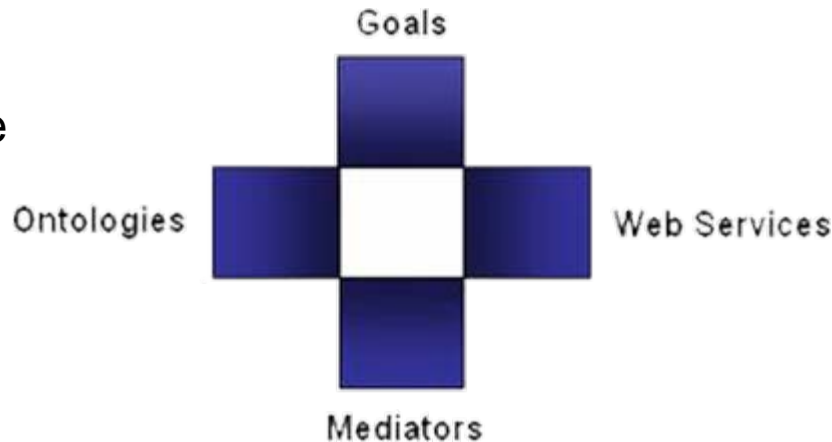
1. **WSMO:** Ontologies, Goals, Web Services, Mediators
2. **OWL-S** WS Description Ontology (Profile, Service Model, Grounding)
3. **SWSF** Process-based Description Model & Language for WS
4. **WSDL-S** semantic annotation of WSDL descriptions

- **Umfassendes Framework, langfristig auf **SESA** abzielend**
SESA = Semantically Empowered Service-Oriented Architecture
 - Top-level Elemente = SESA Kern Elemente
 - Konzeptionelles Modell (WSMO Ontologie)
 - Formale Sprache (WSML), insbesondere Ontologie & Regel Sprache
- **Internationales Konsortium (überwiegend Europäisch)**
 - Start in 2004
 - 78 Mitglieder aus 20 Organisationen
 - W3C member submission im April 2005



Ziele, die ein Client durch die Nutzung von Web Services erreichen möchte

Formal spezifizierte Terminologien zur Beschreibung aller anderen Komponenten in WSMO



Semantische Beschreibung von WS:

- **Capability** (*funktional*)
- **Interfaces** (*interaktion / Nutzung*)

Konnektoren zwischen Komponenten zur Handhabung von Heterogenitäten (verschiedenster Form)

W3C submission 13 April 2005

- **Ontologien als allgegenwärtiges konzeptuelles ‘Datenmodell’ in WSMO**
 - Alle WSMO Element Beschreibungen basieren auf Ontologien
 - Daten, die beim Web Service Verwendung ausgetauscht werden, sind idealerweise semantisch Annotiert (basierend auf Ontologien)
 - Ermöglichen semantische Verarbeitung von Informationen und Beschreibungen (durch Ontologie reasoning)
- **WSMO bietet eine eigene Ontologie-Sprache (ebenso WSML genannt)**
 - Basiert auf unterschiedliches Wissensrepräsentations Formalisms (Bspw. Regeln und DL-basierte Terminologische Beschreibungen, Deduktive Datenbanken)
- **WSMO Ontologie Entwurf unterstützt durch folgende Prinzipien**
 - Modularisierung: Import / Wiederverwendung von Ontologies, Modularer Ansatz zum Entwurf komplexer Ontologien
 - Entkopplung: Heterogenität durch sogenannte OO-Mediators

- **Non functional properties** Erweiterung der Dublin Core Elemente
- **Imported Ontologies** Importierung existierender Ontologien im Falle von Homogenität zwischen den Ontologien
- **Used mediators** OO Mediatoren (Ontologie Import mit Auflösung von Terminologischen Mismatches durch ausgezeichnete Komponente)

Ontology Elemente:

Concepts	Menge von Konzepten, die für die anwendungsspezifische Sicht auf den Problembereich relevant sind
Attributes	Menge von Attributen, die Konzepte reichhaltiger strukturieren
Relations	Beziehungen, die zwischen den identifizierten Konzepten vorherrschen
Functions	Spezielle Relationen zur Beschreibung von Berechnungen
Instances	Menge von Instanzen, die d. Problembereich zugeordnet werden können
Axioms	Feinkörnige (quantitative), axiomatische Beschreibung von weiteren Abhängigkeiten zwischen den Konzepten und entsprechenden Instanzen

- complete item description
- quality aspects
- Web Service Management

- Advertising of Web Service
- Support for WS Discovery

Non-functional Properties

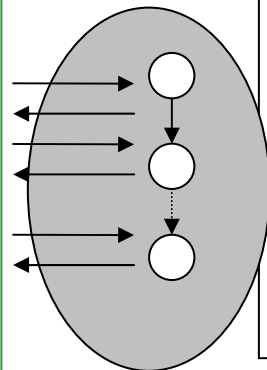
Capability

DC + QoS + Version + financial

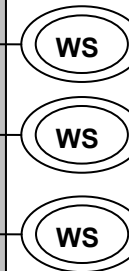
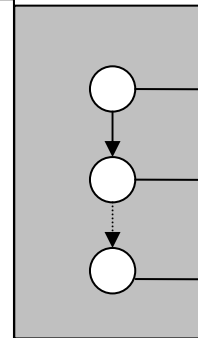
functional description

client-service interaction interface for consuming WS

- External Visible Behavior
- Communication Structure
- 'Grounding'



Web Service Implementation
(not of interest in Web Service Description)



realization of functionality by aggregating other Web Services

- functional decomposition
- WS composition

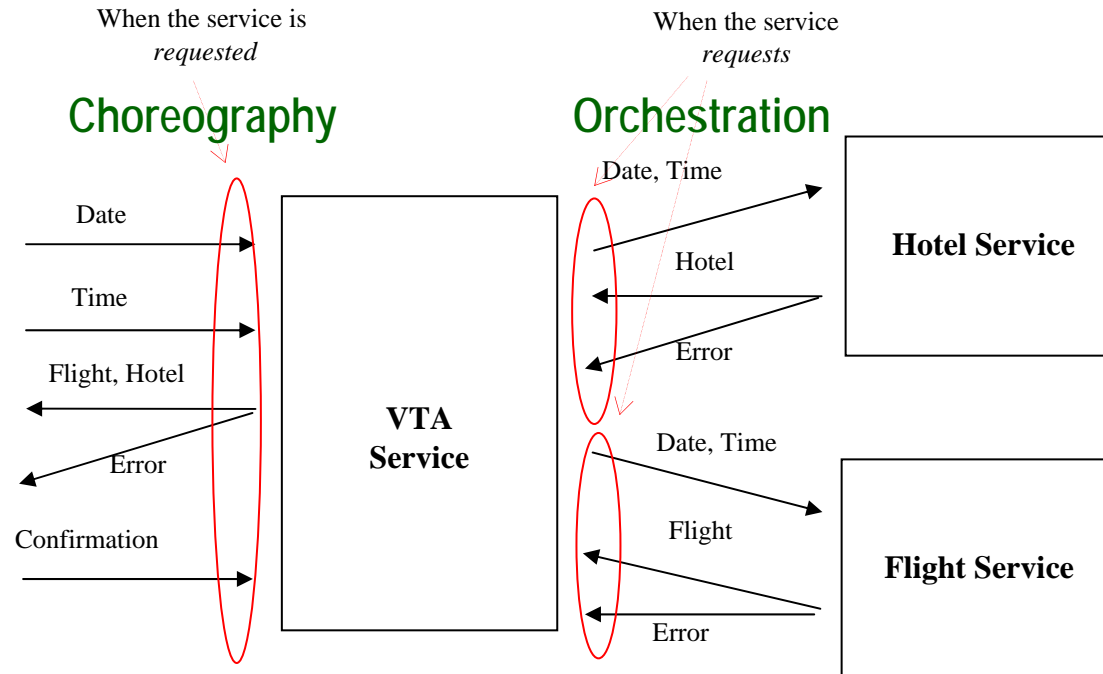
Choreography --- Service Interfaces --- **Orchestration**

- **Non functional properties, Imported Ontologies**
- **Used mediators:** *OO Mediator, WG Mediator*
- **Pre-conditions / Assumptions**

Beschreibt Anforderungen an Eingaben und Zustand der Welt, die zur (erfolgreichen) Ausführung des WS erfüllt sein müssen.
- **Post-conditions / Effects**

Beschreibt die resultierenden Ausgaben und den Zustand der Welt (bspw. Veränderungen), der durch die Ausführung des Web Service mit den gegebenen Eingaben erreicht wird.

- Virtuelles Reisebüro (VTA) Szenario:



- **Choreography** =

Wie mit dem WS interagiert werden muss, um die beschriebene Funktionalität zu nutzen/erhalten

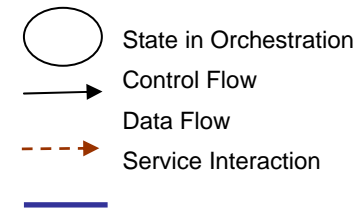
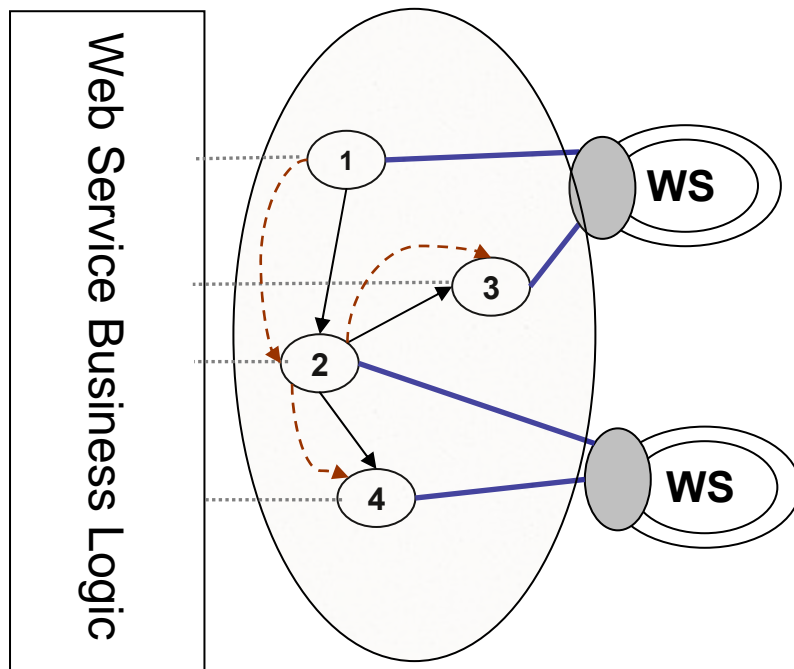
- **Orchestration** =

Wie die beschriebene Funktionalität aufgrund der Nutzung andere WS erreicht wird (öffentliche Elemente eines Workflows)

Beschreibung der Interaktion (Protokoll) zur Nutzung des Web Service

- **Nach aussen sichtbares Verhalten des WS**
 - Aspekte eines Workflows (WF) der innerhalb eines WS and denen Interaktion mit dem Aufrufer notwendig sind
 - Beschrieben durch WF Konstrukte wie: sequence, split, loop, parallel
- **Kommunikationsstruktur**
 - Erwartete und gesendete Nachrichten (Dokumente)
 - deren (zeitliche) Abfolge
- **Grounding**
 - Abbildung von konzeptioneller Ebene auf WS Technologien zur konkreten Ausführung
 - Choreography-bezogenen Fehler (bspw. invalide Eingaben, Message timeout, etc.)
- **Formal Model**
 - Abstract-state Machine basiertes Modell (ausdrucksstarke Automaten)
 - Ermöglicht Reasoning über Web Service Interfaces (bspw. Compatibility of Interact.)

Beschreibt Interface zur Interaktion mit aggregierten Web Services

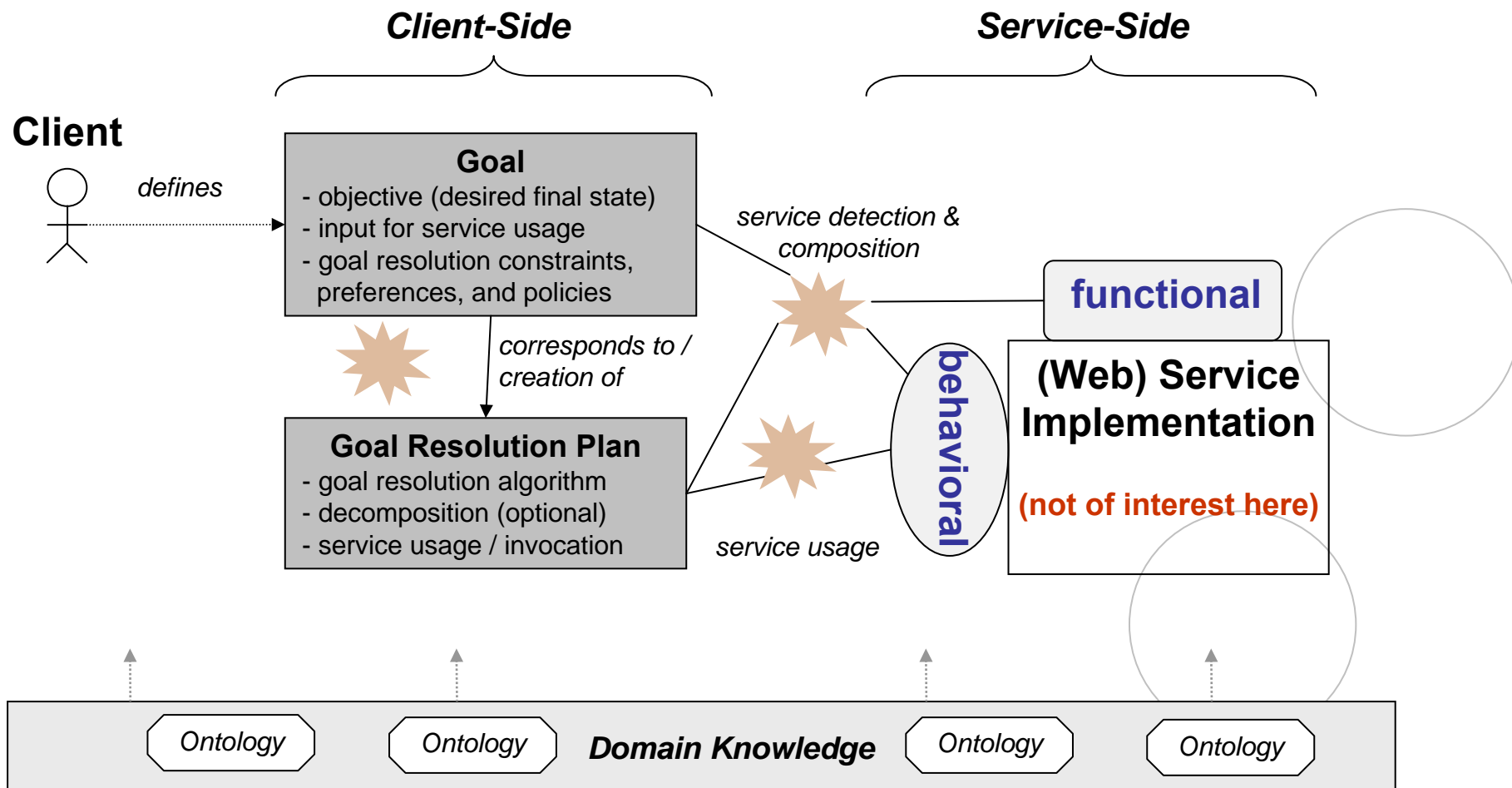


- Dekomposition der WS Funktionalität
- Aggregierte WS werden über deren Choreography Interface angesprochen

- Verhaltensorientierte Interface Beschreibungen für WS in WSMO als “peer-2-peer” Interaktion zu sehen: WS kann selbst Client der aggregierten WS sein.
- Choreography und Orchestration als Subkonzepte eines unifizierenden Konzepts Service Interface mit gemeinsamer Beschreibungssprache
- In WSMO darzustellende Aspekte des Web Service Interface
 - **Dynamik** des Informations Austauschs während des Web Service Ausführung
 - **Nutzung von Ontologien** als unterliegendes, integrierendes Datenmodell
 - Abbildung vom der konzeptuellen Ebene auf entsprechende konkrete Kommunikations Technologien (**Grounding**)
 - Präzises **formales Model (mit definierter Semantik)** für Service Interface Beschreibungen

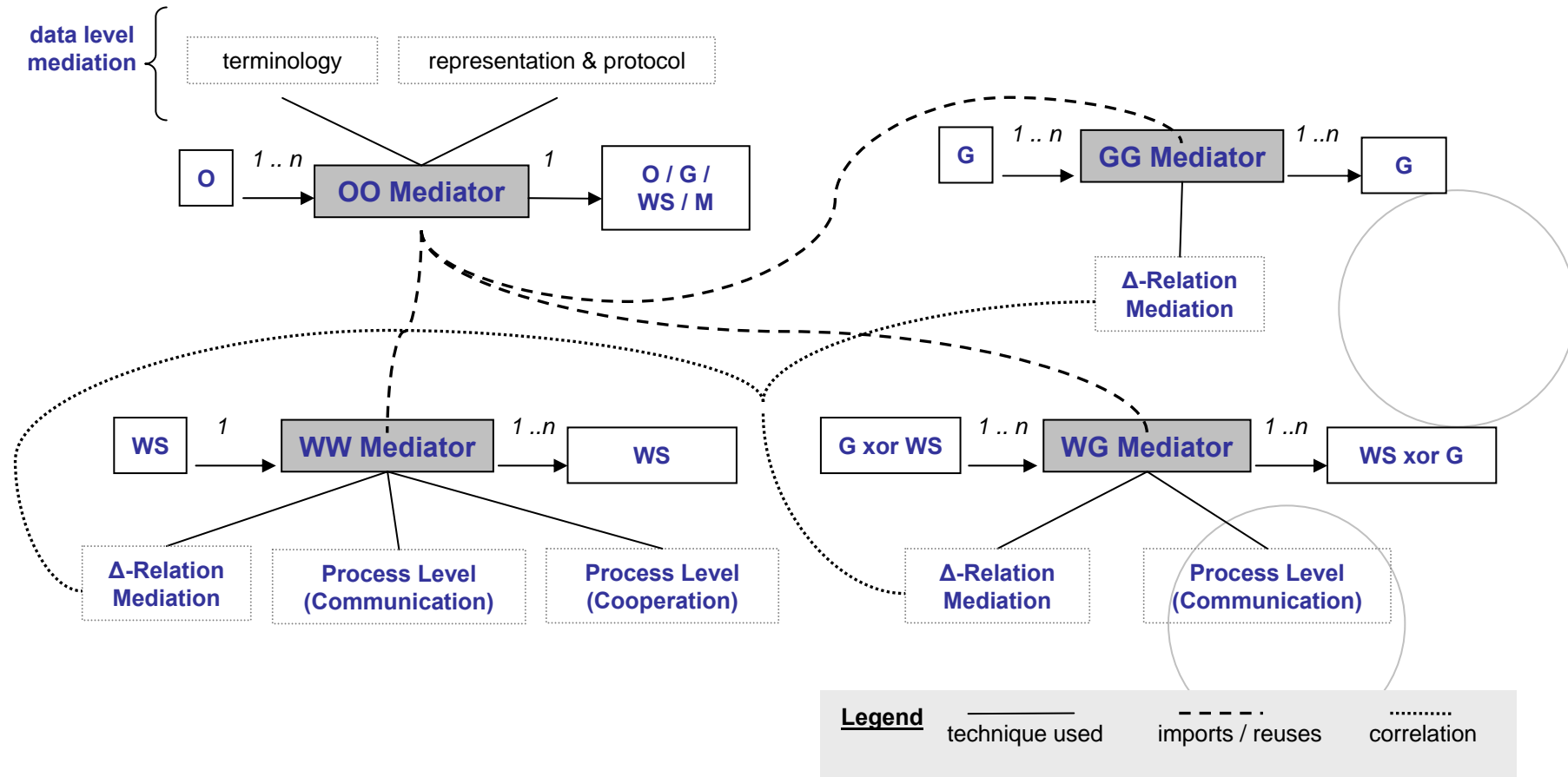
Zielbeschreibung eines Aufrufers: Was soll durch die Ausführung eines WS erreicht werden ?

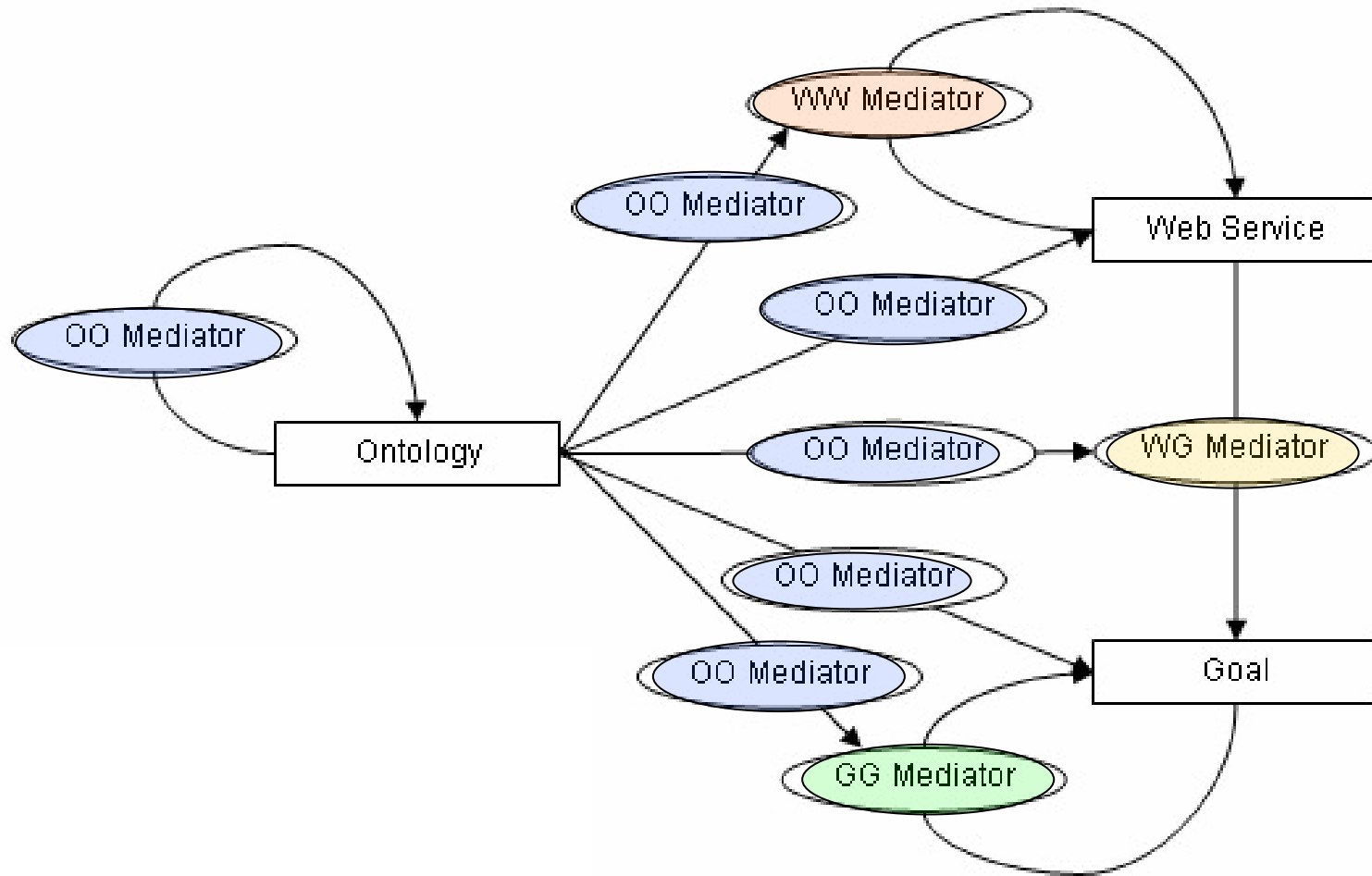
- **Goal-basierter Ansatz**, ähnlich rationalen Agenten in KI
 - Entkopplung von Requester und Provider, ggfs auch in der Art der Modellierung
 - 'intelligente' Mechanismen zur Konstruktion eines Plans zur Goal Erfüllung
 - Goals können wiederverwendet, ausgewählt und verfeinert werden
- **Verwendung von Goals**
 - Ein Requester (Mensch oder Maschine) definiert ein Goal auf der konzeptuellen Ebene in subjektiver Weise. Kann auf Goal Repositories und Goal Verfeinerung zurückgreifen.
 - SWS Umgebungen finden automatisch Web Services, die zur Erfüllung des Goals führen (discovery, composition, execution, etc.)



- **Heterogeneity ...**
 - mismatches on structural / semantic / conceptual / level
 - occur between different components that shall interoperate
 - especially in distributed & open environments like the Internet
- **Concept of Mediation (Wiederhold, 94):**
 - **Mediators** as components that resolve mismatches
 - declarative approach:
 - semantic description of resources
 - 'intelligent' mechanisms that resolve mismatches independent of content
 - mediation cannot be fully automated (integration decision)
- **Levels of Mediation within Semantic Web Services:**
 - (1) **Data Level:** heterogeneous Data Sources
 - (2) **Functional Level:** heterogeneous Functionalities
 - (3) **Protocol & Process Level:** heterogeneous Communication Processes

WSMO Mediators Overview





- WSMO nicht der einzige Vorschlag eines SWS Framework

OWL-S:

- Upper ontology für die semantische Beschreibung von WS
- Basiert auf der Ontologie-Sprache OWL
- Chronologisch erster Ansatz, hauptsächlich in USA entwickelt

SWSF:

- Prozessmodell für Web Services
- Ergebnis der SWSI (internationale Arbeitsgruppe)

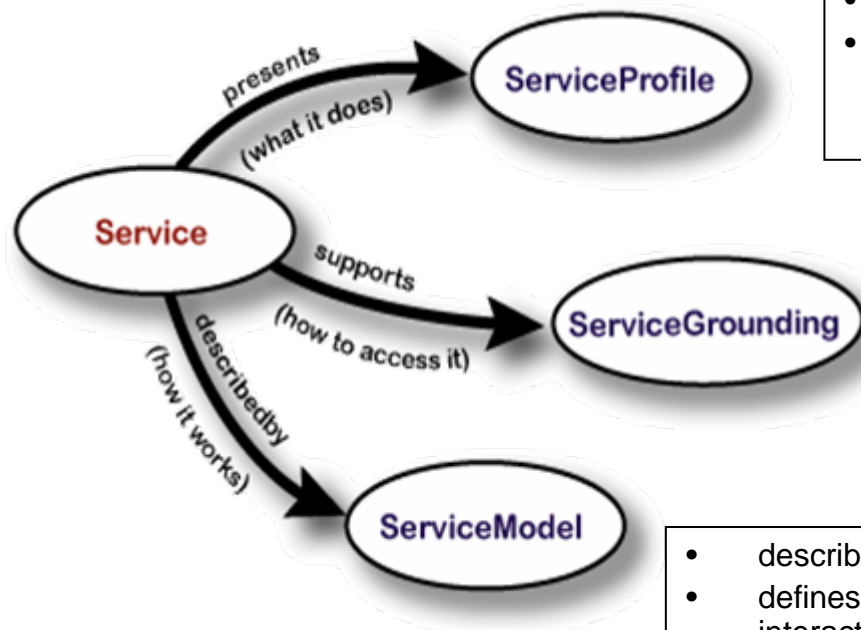
WSDL-S:

- Semantische Annotation von WSDL Descriptions (bottom-up)
- LSDIS Lap (Amit Seth Group) and IBM

- Wir zeigen im folgenden:

- Zentrale Features
- Gemeinsamkeiten und Unterschiede

Upper Ontology for Web Service Descriptions



- capability description (IOPE)
- non-functional properties
- usage: (1) WS advertisement, (2) WS request formulation

- specification of service access information
- builds upon WSDL to define message structure and physical binding layer
- specifies communication protocols & language, transport mechanisms, etc.

- describes internal processes of the service
- defines service interaction protocol for (a) consumption and (b) WS interaction
- process types: simple, atomic, composite
- specifies: (1) abstract messages (ontological content), (2) control flow constructs, (3) *perform* construct

- **OWL-S** = Ontologie & Sprache zur Beschreibung von WS
- **WSMO** = Ontology & Sprache zur Beschreibung der Kern-Elemente einer SWS Umgebung (Systems), langfristig SESA

Main Description Elements Correlation:

OWL-S Profile \approx **WSMO capability + non-functional properties**

OWL-S Process Model \approx **WSMO Service Interfaces**

OWL-S Grounding \approx **derzeitigem WSMO Grounding**

- Goals und Mediators nicht in OWL-S betrachtet
- Unzulänglichkeiten im Service Model (Prozess Beschreibung / Sprache nicht adequat), führte zur Entwicklung von SWSF

- Prozess-Modell für Web Services (FLOWS)
- obwohl eigenständig, kann SWSF als eine Erweiterung von OWL-S angesehen werden (Verfeinerung des Service Models)

Module	Explanation	Major Concepts
FLAWS-Core	basic notions of services as activities composed of atomic activities	Service AtomicProcess composedOf message channel
Control Constraints	common workflow-style process constructs, including OWL-S process model concepts.	Split Sequence Unordered Choice Iterate IfThenElse RepeatUntil
Ordering Constraints	allow specification of activities defined by sequencing properties of atomic processes	OrderedActivity
Occurrence Constraints	support for nondeterministic activities within services	OccActivity
State Constraints	specify activities that are triggered by states (of an overall system)	TriggeredActivity
Exception Constraints	basic infrastructure for modeling exceptions	Exception

Semantische Annotation von WSDL Beschreibungen

1. Annotiere XML Schema in Bezug auf eine Domain-Ontologie

```
<xs:element name="processPOResponse" type="xs:string  
  wssem:modelReference="POOntology#OrderConfirmation" />
```

2. Pre-conditions & Effects für Operations

```
<interface name="PurchaseOrder">  
  <operation name="processPurchaseOrder" pattern=wsdl:in-out>  
    <input messageLabel="processPORequest" element="tns:processPORequest" />  
    <output messageLabel="processPOResponse" element="processPOResponse" />  
    <wssem:precondition name="AccExistsPrecond" wssem:modelReference="onto#AccountExists">  
      <wssem:effect name="ItemReservedEffect" wssem:modelReference="onto#ItemReserved" />  
    </operation>  
</interface>
```

3. WS Kategorisierung über ontology-basierte Begriffe

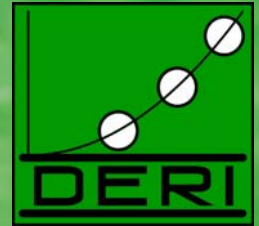
```
<wssem:category name="Electronics"  
  taxonomyURI="http://www.naics.com/" taxonomyCode="443112" />
```

- Ähnliche Ontologische Struktur von WS Beschreibungen
 - Funktionale Descriptions (preconditions & effects)
 - Verhaltens-orientierte Beschreibungen (Interaction)
 - Grounding (verbindung der konzeptuellen Ebene mit der technologischen Ebene; relevant bspw. im Rahmen der automatische Ausführung)
- Zentrale konzeptuelle Unterschiede
 - Formales models für Capabilities
 - Interfaces vs. business process
 - behavioral aspects:
state-based \Leftrightarrow process models \Leftrightarrow operation-level capabilities
- WSMO all-umfassender: definiert elementare “Kern-Elemente für SESA”, während die anderen Frameworks sich auf die Beschreibung von Web Services an sich beschränken

Überblick über die wichtigsten SWS Frameworks



	OWL-S	WSMO	SWSF	WSDL-S
Scope	description model for semantically describing Web services	description model & language for core elements of Semantic Web service technologies	extension of the OWL-S Process Model	semantic annotation of WSDL descriptions
Top Level Elements	Service Profile, Process Model, Grounding	Ontologies, Goals, Web Services, Mediators	Processes	Operations / WSDL descriptions
Service Level Description	non-functional aspects IOPE for service-level functional description	capability (PAPE) for provided and requested functionality	not in the scope	keyword classification (ontology-based)
Operation Level Description	IOPE for processes	interfaces for consumption (choreography) and interaction (orchestration)	internal behavior (atomic and composite processes)	preconditions & effects for WSDL operations
Language (static)	OWL	WSML	SWSLFOL & SWSLRules	not specified
Language (dynamic)	Process Model and OWL	Abstract State Machines	FLAWS	not specified



Teil III:

Beispiel in WSMO

Beispiel: Domain Ontologien (Produkt)



concept Product

provider ofType LegalEntity

description ofType _string

productClass ofType eClassProductIdentifier

concept LegalEntity

address ofType Location

description ofType _string

concept ComplexProduct subConcept of Product

part ofType (1 *) Product

concept Purchase

buyer ofType LegalEntity

seller ofType LegalEntity

product ofType Product

date ofType Date

Product

Reise

Geospatial

concept Reise subConceptOf Product
von ofType Ort
nach ofType Ort
zeitpunkt ofType Datum

concept ZugReise subConcept of Reise
klasse ofType ZugKlasse

concept FlugReise subConcept of Reise
klasse hasValue FlugKlasse

concept EinfachesReise subConcept of Reise

concept KombiReise subConcept of ComplexProduct
part ofType (1 *) EinfacheReise
von ofType Ort
nach ofType Ort

!- ?x memberOf ZugReise
and ?x memberOf FlugReise

instance ersteKlasse memberOf ZugKlasse
instance zweiteKlasse memberOf ZugKlasse

instance businessClass memberOf FlugKlasse
instance economyClass memberOf FlugKlasse
instance firstClass memberOf FlugKlasse

concept Ticket subConcept of Purchase
product ofType Reise
concept ZugTicket subConcept of Ticket
product ofType ZugReise

concept FlugTicket subConcept of Ticket
product hasValue FlugReise

concept KombiTicket subConcept of Ticket
product ofType KombiReise

Beispiel: Domain Ontologien (Geospatial)



concept Location
name ofType _string
place ofType Coordinates
inhabitants ofType _int

concept City subConceptOf Location

concept Region subConceptOf Location
diameter ofType _int

concept Country subConcept of Region

concept Continent subConcept of Region

locatedIn(?l,?r) impliedBy
locatedIn(?l, ?r0) and locatedIn(?r0, ?r)

!- ?x memberOf City and ?r1 memberOf Region
and ?r2 memberOf Region and locatedIn(?x, ?r1)
and locatedIn(?x, ?r2)
naf locatedIn(r1, r2) and naf locatedIn(r2, r1)

instance Köln memberOf City
instance Innsbruck memberOf City
instance Koblenz memberOf City
instance Bukarest memberOf City

instance Austria memberOf Country
instance Germany memberOf Country
instance Romania memberOf Country

instance Europe memberOf Region
instance Australia memberOf Continent
instance EastEurope memberOf Region

locatedIn(Innsbruck, Austria) hasAirport(Köln)
locatedIn(Koblenz, Germany) hasAirport(Innsbruck)
locatedIn(Köln, Germany) hasAirport(Bukarest)
locatedIn(Bukarest, Romania)
locatedIn(Austria, Europe)
locatedIn(Germany, Europe)
locatedIn(Romania, EastEurope)
locatedIn(EastEurope, Europe)

- Webservice-1:
 - Zugreisen innerhalb Deutschlands bis ein Jahr im Voraus

WS1(?start, ?ziel, ?wann)

post: out(?Ticket) equivalent

?start memberOf City and ?ziel memberOf City and ?wann memberOf Datum
and locatedIn(?start, Germany) and locatedIn(?ziel, Germany)
and in-periode(?wann, heute, addiere-periode(heute, 1, „Jahr“))
?Ticket memberOf ZugTicket
and ?Ticket.product[von hasValue ?start,
nach hasValue ?ziel,
zeitpunkt hasValue ?wann]

- Webservice-2:

- Flugreisen zwischen allen Städten mit Flughafen innerhalb Europas bis sechs Monate im Voraus

WS2(?start, ?ziel, ?wann)

post: out(?Ticket) equivalent

?start memberOf City and ?ziel memberOf City and ?wann memberOf Datum
and locatedIn(?start, Europe) and locatedIn(?ziel, Europe)
and hasAirport(?start) and hasAirport(?ziel)
and in-periode(?wann, heute, addiere-periode(heute, 6, „Monat“))
?Ticket memberOf ZugTicket
and ?Ticket.product[von hasValue ?start,
nach hasValue ?ziel,
zeitpunkt hasValue ?wann]

- Goal-1

- Benutzer möchte am 20.Juli 2006 von Innsbruck nach Köln reisen

Goal1

post: out(?Ticket) equivalent

?Ticket memberOf Ticket

and ?wann = _date(2006-07-20)

and ?Ticket.product[von hasValue Innsbruck,
nach hasValue Köln,
zeitpunkt hasValue ?wann]

- Goal-2
 - Benutzer möchte in zwei Wochen von Innsbruck nach Koblenz reisen

Goal2

post: out(?Ticket) equivalent

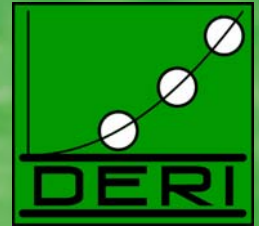
?Ticket memberOf Ticket

and ?wann = addiere-periode(heute, 2, „Woche“)

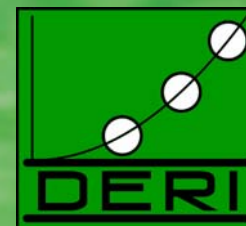
and ?Ticket.product[von hasValue Innsbruck,
nach hasValue Koblenz,
zeitpunkt hasValue ?wann]

- Welche Beschreibungen von WS erfüllen nun welche Goals?
- **Goal-1**
 - Webservice-1 matcht nicht, da nur Reisen innerhalb Deutschlands angeboten werden, aber Innsbruck in Österreich (somit nicht in Deutschland) liegt.
 - Webservice-2 matcht falls die Anfrage bis zum 20. Januar 2006 durchgeführt wird, da sowohl Innsbruck als auch Köln europäische Städte mit Flughafen sind
- **Goal-2**
 - Webservice-1 matcht nicht, aus dem selben Grund wie oben
 - Webservice-2 matcht nicht, da Koblenz keinen Flughafen besitzt
 - Aber Komposition beider Webservices zu einem Web Service, der Kombiticket bzgl WS1 und WS2 liefert (Innsbruck -> Köln per Flug, Köln -> Koblenz per Zug), würde matchen

- SOA wird der Verbreitung existierender WS Technologien einen wahren Boost geben. Umgang mit grosser Zahl von WS erforderlich
- Automatisierung von Aspekten der Web Service Nutzung ist daher wünschenswert, um kosten-effektiver und agiler zu handeln
- Semantik für WS ermöglicht automatisierte Unterstützung zahlreicher solcher Aspekte bei der Nutzung von WS
- Verschiedene Frameworks zur Semantischen Beschreibung von WS sind in Entwicklung (und unter Standardisierung in näherer Zukunft)
- Ontologien sind die wesentliche Basis-Technologie die Integration und Interoperation von System auf der Bedeutungsebene (statt der rein technischen Ebene der Datenrepräsentation) ermöglicht



Danke für Ihre Aufmerksamkeit!



Literatur

- [Alonso et al., 2004] Alonso, G., Casati, F., Kuno, H., and Machiraju, V. (2004). Web Services: Concepts, Architectures and Applications. Data-Centric Systems and Applications. Springer, Berlin, Heidelberg.
- [Berners-Lee, 1999] Berners-Lee, T. (1999). Weaving the Web. Harper, San Francisco, USA.
- [Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. Scientific American, 284(5):34-43.
- [Bussler, 2003] Bussler, C. (2003). B2B Integration: Concepts and Architecture. Springer, Berlin, Heidelberg.
- [Fensel, 2003] Fensel, D. (2003). Ontologies: A Silver Bullet for Knowledge Management and E-Commerce. Springer, Berlin, Heidelberg, 2 edition.
- [Gómez-Pérez et al., 2003] Gómez-Pérez, A., Corcho, O., and Fernandez-Lopez, M. (2003). Ontological Engineering. With Examples from the Areas of Knowledge Management, E-Commerce and Semantic Web. Series of Advanced Information and Knowledge Processing. Springer, Berlin, Heidelberg.
- [Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition, 5:199-220.

- [de Bruijn et al., 2006] de Bruijn, J., Fensel, D., Lausen, H., Polleres, A., Roman, D., and Stollberg, M. (2006). Enabling Semantic Web Services. The Web Service Modeling Ontology. Springer.
- [Fensel and Bussler, 2002] Fensel, D. and Bussler, C. (2002). The Web Service Modeling Framework WSMF. Electronic Commerce Research and Applications, 1(2).
- [McIlraith et al., 2001] McIlraith, S., Cao Son, T., and Zeng, H. (2001). Semantic Web Services. IEEE Intelligent Systems, Special Issue on the Semantic Web, 16(2):46-53.
- [Preist, 2004] Preist, C. (2004). A Conceptual Architecture for Semantic Web Services. In Proc. of the Int. Semantic Web Conf. (ISWC 2004).
- [Roman et al., 2005] Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., and Fensel, D. (2005). Web Service Modeling Ontology. Applied Ontology, 1(1):77-106.
- [Stollberg et al., 2006] Stollberg, M., Feier, C., Roman, D., and Fensel, D. (2006). Semantic Web Services - Concepts and Technology. In Ide, N., Cristea, D., and Tufis, D. (editors), Language Technology, Ontologies, and the Semantic Web. Kluwer Publishers.
- [Sycara et al. 2003] Katia Sycara, Massimo Paolucci, Anupriya Ankolekar and Naveen Srinivasan, "Automated Discovery, Interaction and Composition of Semantic Web services," Journal of Web Semantics, Volume 1, Issue 1, September 2003, pp. 27-46

OWL-S

[Martin, 2004] Martin, D. (2004). OWL-S: Semantic Markup for Web Services. W3C Member Submission 22 November 2004. online: <http://www.w3.org/Submission/OWL-S/>.

WSMO [see also www.wsmo.org]

[Lausen et al., 2005] Lausen, H., Polleres, A., and Roman (eds.), D. (2005). Web Service Modeling Ontology (WSMO). W3C Member Submission 3 June 2005. online: <http://www.w3.org/Submission/WSMO/>.

SWSF

[Battle et al., 2005] Battle, S., Bernstein, A., Boley, H., Grosz, B., Gruninger, M., Hull, R., Kifer, M., D., M., S., M., McGuinness, D., Su, J., and Tabet, S. (2005). Semantic Web Services Framework (SWSF). W3C Member Submission 9 September 2005. online: <http://www.w3.org/Submission/SWSF/>.

WSDL-S

[Akkiraju et al., 2005] Akkiraju, R., Farrell, J., Miller, J., Nagarajan, M., Schmidt, M.- T., Sheth, A., and Verma, K. (2005). Web Service Semantics - WSDL-S. W3C Member Submission 7 November 2005. online: <http://www.w3.org/Submission/WSDL-S/>.

- B. Benatallah, M. Hacid, C. Rey, F. Toumani **Towards Semantic Reasoning for Web Services Discovery**,. In *Proc. of the International Semantic Web Conference (ISWC 2003)*, 2003
- Herzog, R.; Lausen, H.; Roman, D.; Zugmann, P.: **WSMO Registry**. WSMO Working Draft D10 v0.1, 26 April 2004.
- Keller, U.; Lara, R.; Polleres, A. (Eds): **WSMO Web Service Discovery**. WSML Working Draft D5.1, 12 Nov 2004.
- Keller, U.; Lara, R.; Lausen, H.; Polleres, A.; Fensel, D.: **Automatic Location of Services**. In *Proc. of the 2nd European Semantic Web Symposium (ESWS2005)*, Heraklion, Crete, 2005.
- M. Kifer, R. Lara, A. Polleres, C. Zhao, U. Keller, H. Lausen and D. Fensel: **A Logical Framework for Web Service Discovery**. Proc. 1st. Intl. Workshop SWS'2004 at ISWC 2004, Hiroshima, Japan, November 8, 2004, CEUR Workshop Proceedings, ISSN 1613-0073
- Lara, R., Lausen, H.; Toma, I.: (Eds): **WSMX Discovery**. WSMX Working Draft D10 v0.2, 07 March 2005.
- Lei Li and Ian Horrocks. **A software framework for matchmaking based on semantic web technology**. In *Proc. of the Twelfth International World Wide Web Conference (WWW 2003)*, 2003.

Lei Li and Ian Horrocks. **A software framework for matchmaking based on semantic web technology.** In *Proc. of the Twelfth International World Wide Web Conference (WWW 2003), 2003*

Daniel J. Mandell and Sheila A. McIlraith. **Adapting BPEL4WS for the Semantic Web: The Bottom-Up Approach to Web Service Interoperation.** In *Proceedings of the Second International Semantic Web Conference (ISWC2003),*

Massimo Paolucci, Takahiro Kawamura, Terry R. Payne, Katia Sycara; **Importing the Semantic Web in UDDI.** In *Proceedings of Web Services, E-business and Semantic Web Workshop, 2002*

Massimo Paolucci, Takahiro Kawamura, Terry R. Payne, Katia Sycara; **"Semantic Matching of Web Services Capabilities."** In *Proceedings of the 1st International Semantic Web Conference (ISWC2002), 2002*

Preist, C.: **A Conceptual Architecture for Semantic Web Services.** In *Proceedings of the 3rd International Semantic Web Conference (ISWC 2004), 2004, pp. 395 - 409.*

Stollberg, M.; Keller, U.; Fensel, D.: **Partner and Service Discovery for Collaboration on the Semantic Web.** *Proc. 3rd Intl. Conference on Web Services (ICWS 2005), Orlando, Florida, July 2005.*

- [Berardi et al., 2003] Berardi, D., Calvanese, D., Giacomo, G. D., Lenzerini, M., and Mecella, M. (2003). Automatic Composition of e-Services that Export their Behavior. In Proc. of First Int. Conference on Service Oriented Computing (ICSOC).
- [Martens, 2003] Martens, A. (2003). On Compatibility of Web Services. Petri Net Newsletter, 65:12-20.
- [Sirin et al., 2004] Sirin, E., Parsia, B., Wu, D., Hendler, J., and Nau, D. (2004). HTN Planning for Web Service Composition Using SHOP2. Journal of Web Semantics, 1(4):377-396.
- [Pistore and Traverso, 2006] Pistore, M. and Traverso, P. (2006). Theoretical Integration of Discovery and Composition. Deliverable D2.4.6, Knowledge Web.
- [Stollberg, 2005] Stollberg, M. (2005). Reasoning Tasks and Mediation on Choreography and Orchestration in WSMO. In Proceedings of the 2nd International WSMO Implementation Workshop (WIW 2005), Innsbruck, Austria.
- [Traverso and Pistore, 2004] Traverso, P. and Pistore, M. (2004). Automatic Composition of Semantic Web Services into Executable Processes. In Proc. 3rd International Semantic Web Conference (ISWC 2004), Hiroshima, Japan.

- [Cimpian and Mocan, 2005] Cimpian, E. and Mocan, A. (2005). WSMX Process Mediation Based on Choreographies. In Proceedings of the 1st International Workshop on Web Service Choreography and Orchestration for Business Process Management at the BPM 2005, Nancy, France.
- [Mocan (ed.), 2005] Mocan (ed.), A. (2005). WSMX Data Mediation. WSMX Working Draft D13.3. available at: <http://www.wsmo.org/TR/d13/d13.3/v0.2/>.
- [Mocan et al., 2005] Mocan, A., Cimpian, E., Stollberg, M., Scharffe, F., and Scicluna, J. (2005). WSMO Mediators. WSMO deliverable D29 - nal draft 21 Dec 2005. available at: <http://www.wsmo.org/TR/d29/>.
- [Scharffe and de Bruijn, 2005] Scharffe, F. and de Bruijn, J. (2005). A language to specify mappings between ontologies. In Proc. of the Internet Based Systems IEEE Conference (SITIS05).
- [Stollberg et al., 2006] Stollberg, M., Cimpian, E., Mocan, A., and Fensel, D. (2006). A Semantic Web Mediation Architecture. In Proceedings of the 1st Canadian Semantic Web Working Symposium (CSWWS 2006), Quebec, Canada.
- [Wiederhold, 1994] Wiederhold, G. (1994). Mediators in the Architecture of the Future Information Systems. Computer, 25(3):38-49.