

# OWLS2WSDL: Semi-automatische Translation von OWL-S Diensten in WSDL und experimentelle Evaluierung

Vergleichende praktische Untersuchung zur Relation zwischen semantischen und konventionellen Webdiensten in OWL-S und WSDL (Matchmaking)

Oliver Fourman

Fachbereich Grundlagen Informatik Sensortechnik  
Hochschule für Technik und Wirtschaft des Saarlandes

Deutsches Forschungszentrum für Künstliche Intelligenz Saarbrücken

Masterthesis KIM 2006.2007

Vortrag DFKI 29. Juni 2007



# Einführung: Projekt-Kontext

Hochschule für  
Technik und Wirtschaft  
des Saarlandes  
University of Applied Sciences



Deutsches  
Forschungszentrum  
für Künstliche  
Intelligenz GmbH

- Masterarbeit im Studiengang Kommunikationsinformatik
- Thema bearbeitet im Forschungsbereich DMAS am DFKI
- Brückenthema der Projekte ATHENA-IP und SCALLOPS



# Einführung: Projekt-Kontext



- Masterarbeit im Studiengang Kommunikationsinformatik
- Thema bearbeitet im Forschungsbereich DMAS am DFKI
- Brückenthema der Projekte ATHENA-IP und SCALLOPS



## SCALLOPS



# Inhalt

- Teil I: Grundlagen
- Teil II: Translation OWL-S nach WSDL



# Inhalte von Teil I

## 1 Semantic Web Services

- Das Web Services Framework
- Semantic Web
- Web Ontology Language for Services (OWL-S)

## 2 Interoperabilität innerhalb einer SOA

- Use Cases
- Technologien



# Inhalte von Teil II

- 3 Zielbeschreibung
  - Analyse
- 4 Translation OWL-S nach WSDL
  - Konzeptionierung und Design
  - Translation OWL nach XML Schema
  - Translation OWL-S nach WSDL
- 5 Implementierung
  - Architektur
  - Vorstellung des Tools
- 6 Experimentelle Evaluierung
  - Die Technologien WA und OWLS-MX
  - Übersetzung der OWLS-TC
  - Ergebnisse



# Teil I

## Grundlagen



# Das Web Services Framework

- Grundbausteine sind Web Services
- Austausch von Nachrichten (Messaging Framework SOAP) zwischen *Service Provider* und dem *Service Requester*
- Herstellerunabhängig
- Plattformneutral
- Gemeinsames Kommunikationsagreement (WSDL)
- Registrierung und erweiterte Beschreibung des Service in einem Verzeichnisdienst (UDDI)





# Das Semantic Web

Wissensrepräsentation im *Semantic Web* über

- **Taxonomien:** Monohierarchische Klassenstruktur
- **Ontologien:** Netz von Klassenhierarchien und Beziehungen
- **Logik:** Schlussfolgerungen (*engl. reasoning*)

*“An ontology is a specification of a conceptualization.”*  
*Tom Gruber (1994)*

**Definiton Semantic Web:**

*“Netz von Daten, die direkt und indirekt von Maschinen  
verarbeitet werden können.” (Tim Berners-Lee, W3C)*



# Das Semantic Web

Wissensrepräsentation im *Semantic Web* über

- **Taxonomien:** Monohierarchische Klassenstruktur
- **Ontologien:** Netz von Klassenhierarchien und Beziehungen
- **Logik:** Schlussfolgerungen (*engl. reasoning*)

*“An ontology is a specification of a conceptualization.”*  
*Tom Gruber (1994)*

## Definiton Semantic Web:

*“Netz von Daten, die direkt und indirekt von Maschinen verarbeitet werden können.” (Tim Berners-Lee, W3C)*



# Web Ontology Language (OWL)

## OWL Sprachumfang:

- Klassen (*engl. concepts*) und Individuen
- Eigenschaften: Beziehungen in Tripleform (RDF)
  - Datatype Properties
  - Object Properties (Beziehungen zwischen Klassen)
- Restriktionen von Eigenschaften

## OWL Untersprachen:

- OWL-Lite
- OWL-DL (*description logic*)
- OWL-Full



# Web Ontology Language (OWL)

## OWL Sprachumfang:

- Klassen (*engl. concepts*) und Individuen
- Eigenschaften: Beziehungen in Tripleform (RDF)
  - Datatype Properties
  - Object Properties (Beziehungen zwischen Klassen)
- Restriktionen von Eigenschaften

## OWL Untersprachen:

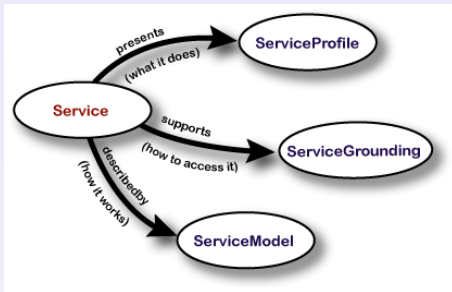
- OWL-Lite
- OWL-DL (*description logic*)
- OWL-Full



# Web Ontology Language for Services (OWL-S)

## OWL-S Ontologien (OWL):

- *ServiceProfile* enthält Service Informationen (vgl. UDDI)
- *ProcessModel* Ablaufsteuerung
- *Grounding* referenziert WSDL



## Abbildung: OWL-S: Upper ontology

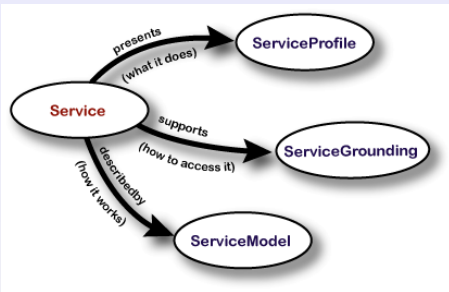
Nutzung weiterer Ontologien wie z.B. ActorDefault.owl: "Actor represents a Requester or Provider who might request or offer a service."



# Web Ontology Language for Services (OWL-S)

## OWL-S Ontologien (OWL):

- *ServiceProfile* enthält Service Informationen (vgl. UDDI)
- *ProcessModel* Ablaufsteuerung
- *Grounding* referenziert WSDL



## Abbildung: OWL-S: Upper ontology

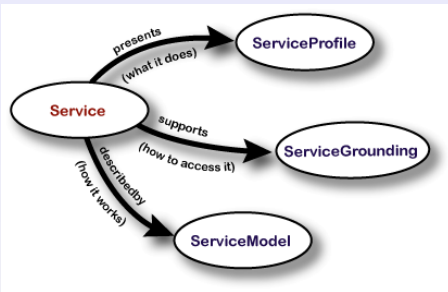
Nutzung weiterer Ontologien wie z.B. ActorDefault.owl: "Actor represents a Requester or Provider who might request or offer a service."



# Web Ontology Language for Services (OWL-S)

## OWL-S Ontologien (OWL):

- *ServiceProfile* enthält Service Informationen (vgl. UDDI)
- *ProcessModel* Ablaufsteuerung
- *Grounding* referenziert WSDL



## Abbildung: OWL-S: Upper ontology

Nutzung weiterer Ontologien wie z.B. ActorDefault.owl: "Actor represents a Requester or Provider who might request or offer a service."



# Service Beschreibung

## Prozesse (*ProcessModel*)

- AtomicProcess, SimpleProcess, CompositeProcess
- Parametertypen: Primitive Datentypen oder OWL-Klassen

## AtomicProcess (*engl.capabilities*)

- Eingaben (*inputs*)
- Ausgaben (*outputs*)
- Vorbedingungen (*preconditions*)
- Nachbedingungen (*results*)

Ein- und Ausgaben eines Prozesses bilden dessen Signatur.





# Service Beschreibung

## Prozesse (*ProcessModel*)

- AtomicProcess, SimpleProcess, CompositeProcess
- Parametertypen: Primitive Datentypen oder OWL-Klassen

## AtomicProcess (*engl.capabilities*)

- Eingaben (*inputs*)
- Ausgaben (*outputs*)
- Vorbedingungen (*preconditions*)
- Nachbedingungen (*results*)

Ein- und Ausgaben eines Prozesses bilden dessen Signatur.



# Service Beschreibung

## Prozesse (*ProcessModel*)

- AtomicProcess, SimpleProcess, CompositeProcess
- Parametertypen: Primitive Datentypen oder OWL-Klassen

## AtomicProcess (*engl.capabilities*)

- Eingaben (*inputs*)
- Ausgaben (*outputs*)
- Vorbedingungen (*preconditions*)
- Nachbedingungen (*results*)

Ein- und Ausgaben eines Prozesses bilden dessen Signatur.



# WSDL Grounding / Mapping der Service-Signatur

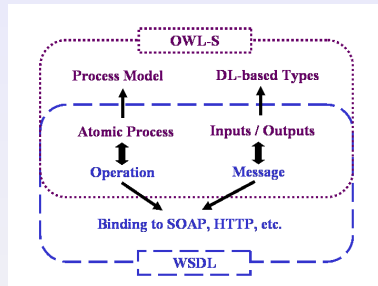


Abbildung: WSDL Grounding (W3C)



# WSDL Grounding / Mapping der Service-Signatur

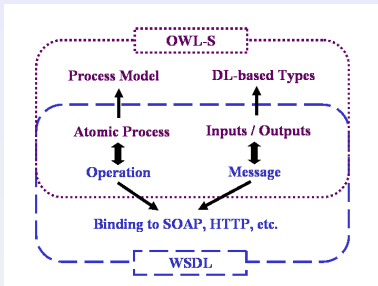


Abbildung: WSDL Grounding (W3C)

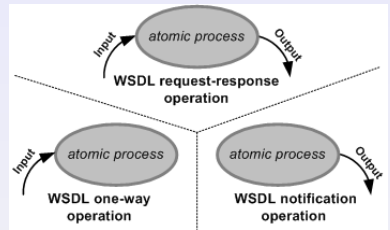


Abbildung: Mapping OWL-S, WSDL

# Use Cases (innerhalb einer SOA)

## Basic Use Cases

- Provide (Semantic) Web Service(s)
- Discover (Semantic) Web Service(s)
- Invoke (Semantic) Web Service(s)

## Operations (Translation, Conversion)

- WSDL Grounding of OWL-S Process
- OWL2XSD, XML2OWL
- **OWLS2WSDL**, WSDL2OWL-S
- OWLS2BPEL

## Rollen

- Ontology Engineer
- Web Services Engineer
- Application Developer
- SOA Expert
- Service Consumer



# Use Cases (innerhalb einer SOA)

## Basic Use Cases

- Provide (Semantic) Web Service(s)
- Discover (Semantic) Web Service(s)
- Invoke (Semantic) Web Service(s)

## Operations (Translation, Conversion)

- WSDL Grounding of OWL-S Process
- OWL2XSD, XML2OWL
- **OWLS2WSDL**, WSDL2OWL-S
- OWLS2BPEL

## Rollen

- Ontology Engineer
- Web Services Engineer
- Application Developer
- SOA Expert
- Service Consumer



# Use Cases (innerhalb einer SOA)

## Basic Use Cases

- Provide (Semantic) Web Service(s)
- Discover (Semantic) Web Service(s)
- Invoke (Semantic) Web Service(s)

## Operations (Translation, Conversion)

- WSDL Grounding of OWL-S Process
- OWL2XSD, XML2OWL
- **OWLS2WSDL**, WSDL2OWL-S
- OWLS2BPEL

## Rollen

- Ontology Engineer
- Web Services Engineer
- Application Developer
- SOA Expert
- Service Consumer



# Interoperabilität von Semantischen Web Services

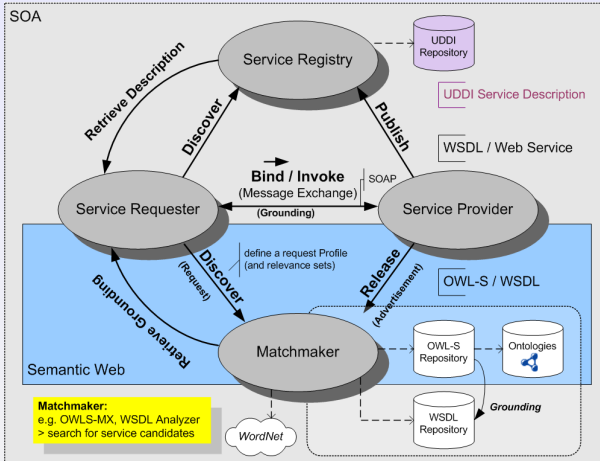
Task	OWL-S	WSDL
<b>Discovery</b>	Service Profile (set) → Matchmaking	UDDI → Using API
<b>Composition Orchestration</b>	Process Model	WS-BPEL aka BPEL4WS
<b>Invocation</b>	WSDL Grounding XSLT Transformation	WSDL/SOAP

On the idea of ISRI-Seminar-05 (Katia Sycara, CMU)





# Technologien zum Auffinden von Web Services



## Teil II

# Besprechung des Themas



# Inhalte von Teil II

- 3 Zielbeschreibung
  - Analyse
- 4 Translation OWL-S nach WSDL
  - Konzeptionierung und Design
  - Translation OWL nach XML Schema
  - Translation OWL-S nach WSDL
- 5 Implementierung
  - Architektur
  - Vorstellung des Tools
- 6 Experimentelle Evaluierung
  - Die Technologien WA und OWLS-MX
  - Übersetzung der OWLS-TC
  - Ergebnisse



# Ziele dieser Arbeit

- 1 Untersuchung der Möglichkeiten der Generierung von WSDL aus OWL-S und Konkretisierung von WSDL Groundings
- 2 Semi-automatische Translation von OWL-S nach WSDL (Toolentwicklung)
- 3 Vergleichende Analyse der Performanz einer ähnlichkeitsbasierten Dienstsuche mit *WSDL Analyzer* und *OWLS-MX* über eine entsprechend aufgebaute Testkollektion.



# Ziele dieser Arbeit

- 1 Untersuchung der Möglichkeiten der Generierung von WSDL aus OWL-S und Konkretisierung von WSDL Groundings
- 2 Semi-automatische Translation von OWL-S nach WSDL (Toolentwicklung)
- 3 Vergleichende Analyse der Performanz einer ähnlichkeitsbasierten Dienstsuche mit *WSDL Analyzer* und *OWLS-MX* über eine entsprechend aufgebaute Testkollektion.



# Ziele dieser Arbeit

- 1 Untersuchung der Möglichkeiten der Generierung von WSDL aus OWL-S und Konkretisierung von WSDL Groundings
- 2 Semi-automatische Translation von OWL-S nach WSDL (Toolentwicklung)
- 3 Vergleichende Analyse der Performanz einer ähnlichkeitsbasierten Dienstsuche mit *WSDL Analyzer* und *OWLS-MX* über eine entsprechend aufgebaute Testkollektion.



# Analyse

- Diskussion *top down* Ansatz
- Untersuchung des Mappings OWL-S auf WSDL
- Einordnung des Themas, Abgrenzung
- Bedeutung der Translation OWL nach XML Schema

## Vorgehensmodell

- Konzeptionierung, Diagnose und Korrekturen (iterativ)
  - Validierung der generierten Ergebnisse (*XMLSpy*, *CodeGen*)
  - Evaluierung (*WSDL Analyzer*, *OWLS-MX*)
- Anlehnung an das *Clean Room* Vorgehensmodell



# Analyse

- Diskussion *top down* Ansatz
- Untersuchung des Mappings OWL-S auf WSDL
- Einordnung des Themas, Abgrenzung
- Bedeutung der Translation OWL nach XML Schema

## Vorgehensmodell

- Konzeptionierung, Diagnose und Korrekturen (iterativ)
  - Validierung der generierten Ergebnisse (*XMLSpy*, *CodeGen*)
  - Evaluierung (*WSDL Analyzer*, *OWLS-MX*)
- Anlehnung an das *Clean Room* Vorgehensmodell





# Analyse

- Diskussion *top down* Ansatz
- Untersuchung des Mappings OWL-S auf WSDL
- Einordnung des Themas, Abgrenzung
- Bedeutung der Translation OWL nach XML Schema

## Vorgehensmodell

- Konzeptionierung, Diagnose und Korrekturen (iterativ)
  - Validierung der generierten Ergebnisse (*XMLSpy*, *CodeGen*)
  - Evaluierung (*WSDL Analyzer*, *OWLS-MX*)
- Anlehnung an das *Clean Room* Vorgehensmodell



# Analyse

- Diskussion *top down* Ansatz
- Untersuchung des Mappings OWL-S auf WSDL
- Einordnung des Themas, Abgrenzung
- Bedeutung der Translation OWL nach XML Schema

## Vorgehensmodell

- Konzeptionierung, Diagnose und Korrekturen (iterativ)
  - Validierung der generierten Ergebnisse (*XMLSpy*, *CodeGen*)
  - Evaluierung (*WSDL Analyzer*, *OWLS-MX*)
- Anlehnung an das *Clean Room* Vorgehensmodell



# Analyse

- Diskussion *top down* Ansatz
- Untersuchung des Mappings OWL-S auf WSDL
- Einordnung des Themas, Abgrenzung
- Bedeutung der Translation OWL nach XML Schema

## Vorgehensmodell

- Konzeptionierung, Diagnose und Korrekturen (iterativ)
  - Validierung der generierten Ergebnisse (*XMLSpy*, *CodeGen*)
  - Evaluierung (*WSDL Analyzer*, *OWLS-MX*)
- Anlehnung an das *Clean Room* Vorgehensmodell



# Konzeptionierung und Design

## Design Direktiven

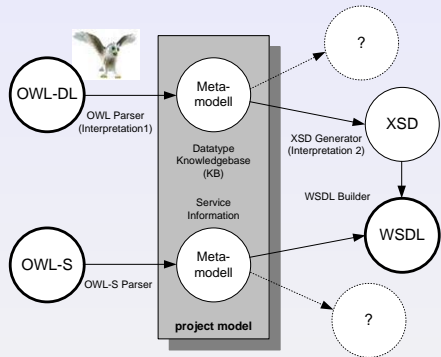
- Nutzung von Standards
- Trennung von Parsen und Code Generierung
- Wissensbasis für Datentypen
- Metamodell für Datentypen und Service-Beschreibungen



# Konzeptionierung und Design

## Design Direktiven

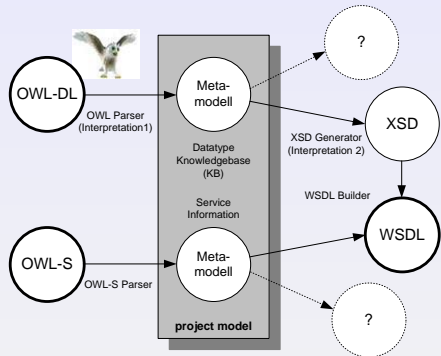
- Nutzung von Standards
- Trennung von Parsen und Code Generierung
- Wissensbasis für Datentypen
- Metamodell für Datentypen und Service-Beschreibungen



# Konzeptionierung und Design

## Design Direktiven

- Nutzung von Standards
- Trennung von Parsen und Code Generierung
- Wissensbasis für Datentypen
- Metamodell für Datentypen und Service-Beschreibungen



Herausforderung: Interpretation von OWL-DL



# OWL Parser

## Parsen von OWL (Definitionen)

- Reguläre Klassen (*concepts*) und anonyme Klassen (DL)
- Eigenschaften
- Kardinalitäten und Wertebereich (*Individuals*)
- Klassenhierarchien (*depth*)

## Interpretieren von OWL Konzepten (Meta-Ebene)

- Bildung von Datentypen aus OWL-Klassen
- Mapping auf passende XML Schema Elemente
- Abbildung der OWL Grammatik mit XML Schema  
(*experimental*)



# OWL Parser

## Parsen von OWL (Definitionen)

- Reguläre Klassen (*concepts*) und anonyme Klassen (DL)
- Eigenschaften
- Kardinalitäten und Wertebereich (*Individuals*)
- Klassenhierarchien (*depth*)

## Interpretieren von OWL Konzepten (Meta-Ebene)

- Bildung von Datentypen aus OWL-Klassen
- Mapping auf passende XML Schema Elemente
- Abbildung der OWL Grammatik mit XML Schema (*experimental*)





# XSD Generator

## Generierung von

- Schema Elementen
- Kardinalitäten (*particles*)
- SimpleType Typen
- ComplexType Typen
- Abgeleiteten Typen (*restriction, extension*)

### Entwurfsmuster

- Venetian Blind
- Hierarchy Pattern

### Konfigurationsmöglichkeiten

- Defaults (*base type*)
- Hierarchy Pattern
- Anonyme Typen



# XSD Generator

## Generierung von

- Schema Elementen
- Kardinalitäten (*particles*)
- SimpleType Typen
- ComplexType Typen
- Abgeleiteten Typen (*restriction, extension*)

## Entwurfsmuster

- Venetian Blind
- Hierarchy Pattern

## Konfigurationsmöglichkeiten

- Defaults (*base type*)
- Hierarchy Pattern
- Anonyme Typen



# XSD Generator

## Generierung von

- Schema Elementen
- Kardinalitäten (*particles*)
- SimpleType Typen
- ComplexType Typen
- Abgeleiteten Typen (*restriction, extension*)

### Entwurfsmuster

- Venetian Blind
- Hierarchy Pattern

### Konfigurationsmöglichkeiten

- Defaults (*base type*)
- Hierarchy Pattern
- Anonyme Typen



# OWL2XSD, Basis Translation

`owl:Class`  
`owl:DatatypeProperty`       $\longrightarrow$       `xsd:simpleType`



# OWL2XSD, Basis Translation

owl:Class  
owl:DatatypeProperty      →      xsd:simpleType

owl:Class  
owl:ObjectProperty      ↘      xsd:simpleType  
                                 ↙      xsd:complexType



# OWL2XSD, Basis Translation

owl:Class  
owl:DatatypeProperty      →      xsd:simpleType

owl:Class  
owl:ObjectProperty      ↘  
                                 ↗      xsd:simpleType  
                                 ↘      xsd:complexType

owl:Restriction      →      xsd:restriction  
                                 xsd:enumeration  
                                 min-/maxOccurs



# OWL2XSD, erweiterte Translation

## ■ Übersetzung der hierarchischen Klassenstruktur

`rdfs:subClassOf` → `xsd:complexContent`  
`xsd:restriction base`  
`substitutionGroup`  
`abstract`

- Übersetzung der Beschreibungs-Logik (DL: anonyme Klassen)
- Reasoning (durch Parser) erweitert die Wissensbasis

# OWL2XSD, erweiterte Translation

## ■ Übersetzung der hierarchischen Klassenstruktur

`rdfs:subClassOf` → `xsd:complexContent`  
`xsd:restriction base`  
`substitutionGroup`  
`abstract`

- Übersetzung der Beschreibungs-Logik (DL: anonyme Klassen)
- Reasoning (durch Parser) erweitert die Wissensbasis



# OWL2XSD, erweiterte Translation

- Übersetzung der hierarchischen Klassenstruktur

`rdfs:subClassOf` → `xsd:complexContent`  
`xsd:restriction base`  
`substitutionGroup`  
`abstract`

- Übersetzung der Beschreibungs-Logik (DL: anonyme Klassen)
- Reasoning (durch Parser) erweitert die Wissensbasis

# OWLS2WSDL, Translations-Schritte

## Parsen von OWL-S

- Signatur des atomaren Prozesses
- Festlegung der Parameter-Reihenfolge
- Speicherung des Mappings in Metamodell (Grounding)

## Generieren von WSDL

- *Straight Forward* anhand der abstrakten Service-Beschreibung
- Abhängigkeiten zu Parametertypen
- Generierung von XML Schema Typen (Parameter, OWL2XSD)



# OWLS2WSDL, Translations-Schritte

## Parsen von OWL-S

- Signatur des atomaren Prozesses
- Festlegung der Parameter-Reihenfolge
- Speicherung des Mappings in Metamodell (Grounding)

## Generieren von WSDL

- *Straight Forward* anhand der abstrakten Service-Beschreibung
- Abhängigkeiten zu Parametertypen
- Generierung von XML Schema Typen (Parameter, OWL2XSD)



# Abbildungsmöglichkeiten OWL-S nach WSDL

## Serialisierung von OWL-S

Die Abbildung von OWL-S nach WSDL ist nicht eindeutig.

Interpretation von OWL und OWL-S (Ontologien)

- abhängig von dem zu parsenden Sprachumfang (OWL-DL)
- je nach Vererbungstiefe
- durch Einbeziehung von Individuen (Wertebereich)
- durch Interpretation von *sufficient* und *necessary*

Service-Schnittstelle

- Parameterposition



# Abbildungsmöglichkeiten OWL-S nach WSDL

## Serialisierung von OWL-S

Die Abbildung von OWL-S nach WSDL ist nicht eindeutig.

### Interpretation von OWL und OWL-S (Ontologien)

- abhängig von dem zu parsenden Sprachumfang (OWL-DL)
- je nach Vererbungstiefe
- durch Einbeziehung von Individuen (Wertebereich)
- durch Interpretation von *sufficient* und *necessary*

### Service-Schnittstelle

- Parameterposition



# Abbildungsmöglichkeiten OWL-S nach WSDL

## Serialisierung von OWL-S

Die Abbildung von OWL-S nach WSDL ist nicht eindeutig.

### Interpretation von OWL und OWL-S (Ontologien)

- abhängig von dem zu parsenden Sprachumfang (OWL-DL)
- je nach Vererbungstiefe
- durch Einbeziehung von Individuen (Wertebereich)
- durch Interpretation von *sufficient* und *necessary*

### Service-Schnittstelle

- Parameterposition



# Interpretation

## Zwischenmodell

Zwischenmodell unterschiedlich gut (parserabhängig).

## Zielformat

Zielformat unterschiedlich gut (generatorabhängig).

- Welche XML Schema Sprachelemente werden genutzt?
- Wie werden anonyme OWL Konstrukte interpretiert?



# Interpretation

## Zwischenmodell

Zwischenmodell unterschiedlich gut (parserabhängig).

## Zielformat

Zielformat unterschiedlich gut (generatorabhängig).

- Welche XML Schema Sprachelemente werden genutzt?
- Wie werden anonyme OWL Konstrukte interpretiert?





# Interpretation

## Zwischenmodell

Zwischenmodell unterschiedlich gut (parserabhängig).

## Zielformat

Zielformat unterschiedlich gut (generatorabhängig).

- Welche XML Schema Sprachelemente werden genutzt?
- Wie werden anonyme OWL Konstrukte interpretiert?

“However, OWL is developed as a vocabulary extension of the Resource Description Format (RDF). RDF is a data model whose models can be serialized in XML syntax in many syntactically different but semantically equivalent ways. As a result, the encoding of an OWL model in XML syntax is a one-to-many mapping.”

(Balzer, Liebig; Univ. Ulm)



# Semi-automatische Translation

## Konfigurationsmöglichkeiten

- Vererbungslevel (*depth*)
- Benutztes Entwurfsmuster
- Defaultwerte für Basistypen

## Manuelle Veränderung der Wissensbasis

- Setzen des Basistyps (*base type*) eines SimpleType
- Verwendung von primitiven Datentypen (XML Schema)

## Unterstützung durch das Tool

- Vererbung von Settings innerhalb der Hierarchie
- Auswertung von `rdf:type` `rdf:resource=`



# Semi-automatische Translation

## Konfigurationsmöglichkeiten

- Vererbungslevel (*depth*)
- Benutztes Entwurfsmuster
- Defaultwerte für Basistypen

## Manuelle Veränderung der Wissensbasis

- Setzen des Basistyps (*base type*) eines SimpleType
- Verwendung von primitiven Datentypen (XML Schema)

## Unterstützung durch das Tool

- Vererbung von Settings innerhalb der Hierarchie
- Auswertung von `rdf:type` `rdf:resource=`



# Semi-automatische Translation

## Konfigurationsmöglichkeiten

- Vererbungslevel (*depth*)
- Benutztes Entwurfsmuster
- Defaultwerte für Basistypen

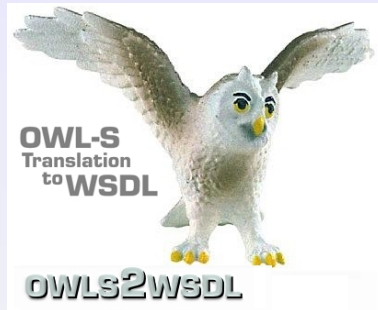
## Manuelle Veränderung der Wissensbasis

- Setzen des Basistyps (*base type*) eines SimpleType
- Verwendung von primitiven Datentypen (XML Schema)

## Unterstützung durch das Tool

- Vererbung von Settings innerhalb der Hierarchie
- Auswertung von `rdf:type` `rdf:resource=`





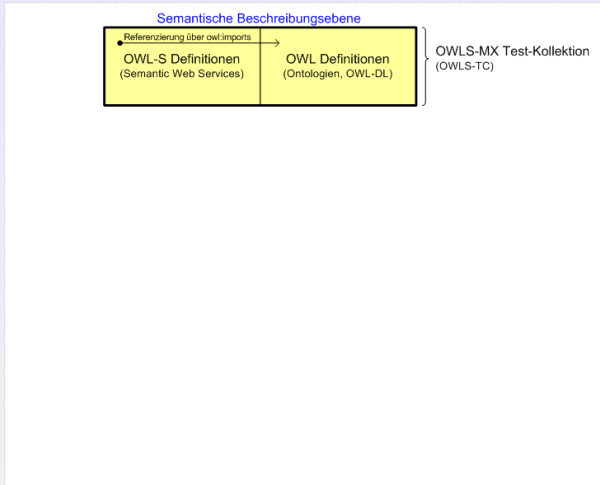


Abbildung: Translation-Stack



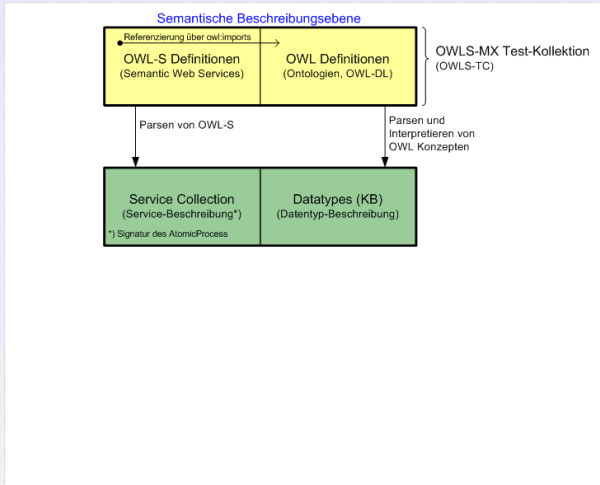


Abbildung: Translation-Stack



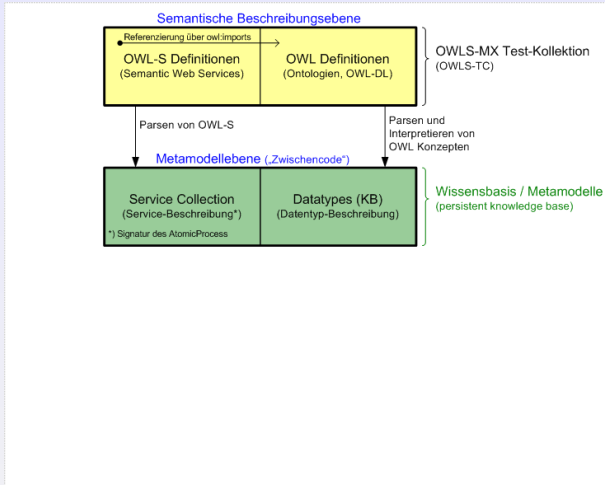


Abbildung: Translation-Stack





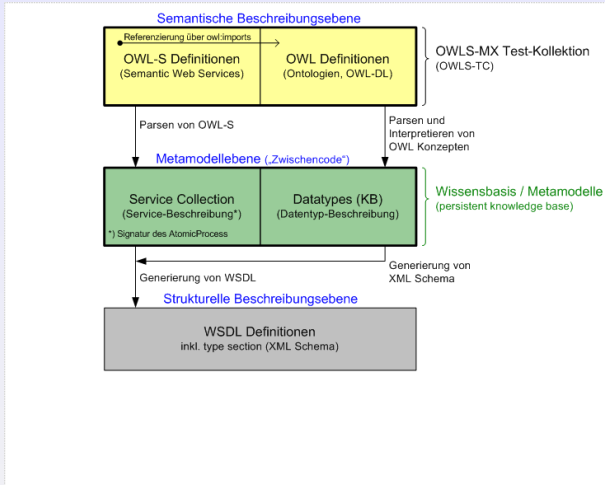


Abbildung: Translation-Stack



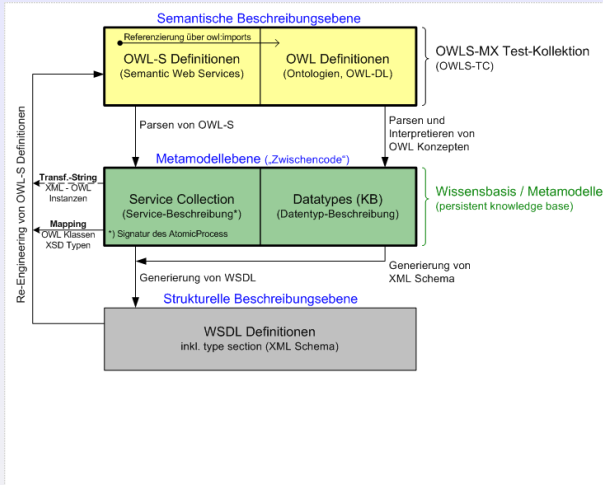


Abbildung: Translation-Stack



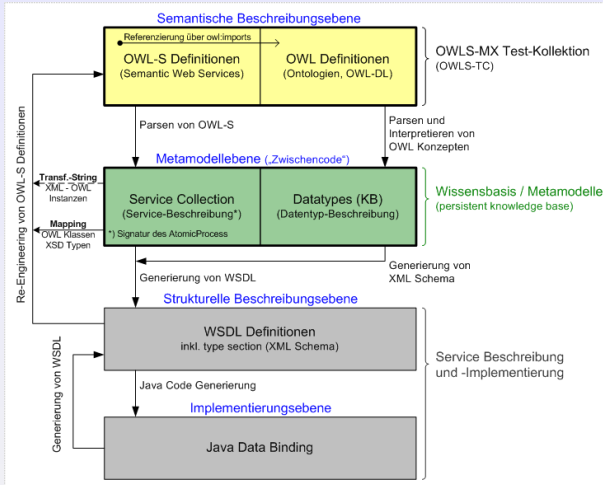


Abbildung: Translation-Stack



# Technologien (APIs)

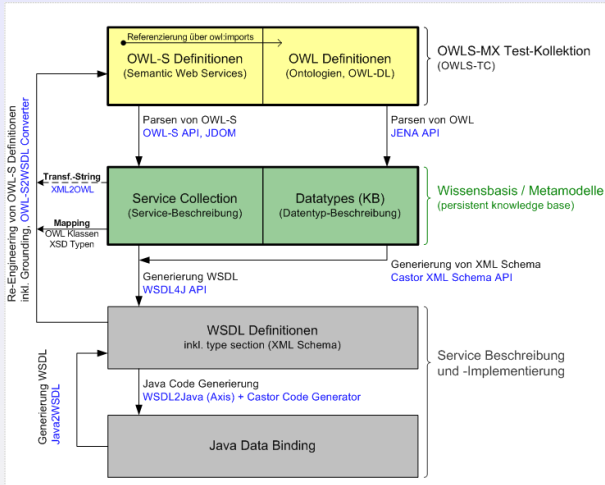
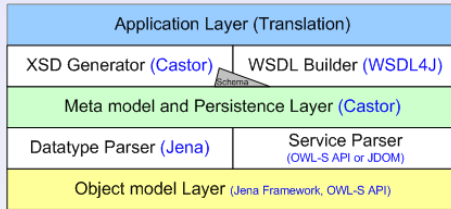


Abbildung: Translation-Stack

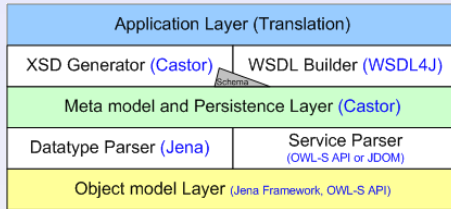
# Programmteile



## Weitere Funktionen

- Projekt (umfasst Service Collection und Knowledgebase)
- Command Line Interface
- Benutzeroberfläche (Konfiguration)

# Programmteile



## Weitere Funktionen

- Projekt (umfasst Service Collection und Knowledgebase)
- Command Line Interface
- Benutzeroberfläche (Konfiguration)

# OWLS2WSDL Tool

## Programmfunktionen

- Generierung von XML Schema aus OWL (OWL2XSD)
- Translation OWL-S nach WSDL (OWLS2WSDL)
- Re-Engineering: Konkretisierung eines WSDL Groundings

## Arbeitsweise

- 1 Anlegen eines Projektes
- 2 Laden der OWL-S Definitionen (Parser)
- 3 Auflösen von Abhängigkeiten der Schnittstelle zu Datentypen
- 4 Konfiguration, optionale Bearbeitung der Typen
- 5 Generierung von WSDL-Beschreibungen (inkl. XML Schema)



# OWLS2WSDL Tool

## Programmfunktionen

- Generierung von XML Schema aus OWL (OWL2XSD)
- Translation OWL-S nach WSDL (OWLS2WSDL)
- Re-Engineering: Konkretisierung eines WSDL Groundings

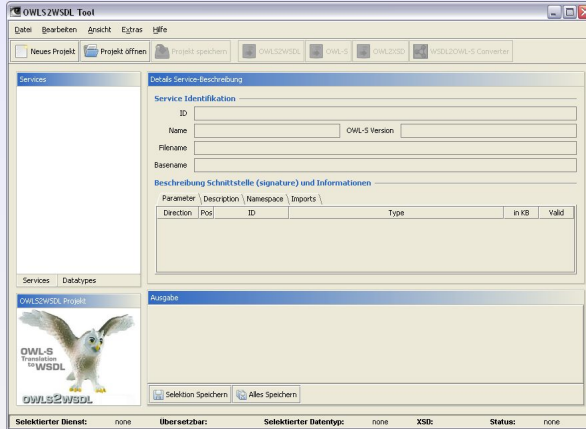
## Arbeitsweise

- 1 Anlegen eines Projektes
- 2 Laden der OWL-S Definitionen (Parser)
- 3 Auflösen von Abhängigkeiten der Schnittstelle zu Datentypen
- 4 Konfiguration, optionale Bearbeitung der Typen
- 5 Generierung von WSDL-Beschreibungen (inkl. XML Schema)





# Vorführung des OWLS2WSDL Tools



Beispiele: ZipCode, Student (OWL2XSD); CarPriceService



# Future Work

- OWL-S Objektmodell (OWL-S Parser, OWL-S API) \*
- Erweiterte Fehlerbehandlung (OWL Parser, Jena Eyeball)
- WSDL Builder
  - Variation des Zielformats (WSDL-S)
  - Automatische Konfiguration (WA)
- Re-Engineering
  - xsitTransformationString (Re-Engineering, JXML2OWL)
  - Generierung von OWL-S 1.0 (translator 1.1 nach 1.0) \*
  - Validierung von OWL-S Definitionen mit API (broken) \*
- Integration von WSDL2Java (*top down*)

\*) *voluntarily contributions (OpenSource)*



# Future Work

- OWL-S Objektmodell (OWL-S Parser, OWL-S API) \*
- Erweiterte Fehlerbehandlung (OWL Parser, Jena Eyeball)
- WSDL Builder
  - Variation des Zielformats (WSDL-S)
  - Automatische Konfiguration (WA)
- Re-Engineering
  - xsltTransformationString (Re-Engineering, JXML2OWL)
  - Generierung von OWL-S 1.0 (translator 1.1 nach 1.0) \*
  - Validierung von OWL-S Definitionen mit API (broken) \*
- Integration von WSDL2Java (*top down*)

\*) *voluntarily contributions (OpenSource)*



# Future Work

- OWL-S Objektmodell (OWL-S Parser, OWL-S API) \*
- Erweiterte Fehlerbehandlung (OWL Parser, Jena Eyeball)
- WSDL Builder
  - Variation des Zielformats (WSDL-S)
  - Automatische Konfiguration (WA)
- Re-Engineering
  - xsltTransformationString (Re-Engineering, JXML2OWL)
  - Generierung von OWL-S 1.0 (translator 1.1 nach 1.0) \*
  - Validierung von OWL-S Definitionen mit API (broken) \*
- Integration von WSDL2Java (*top down*)

\*) *voluntarily contributions (OpenSource)*



# Future Work

- OWL-S Objektmodell (OWL-S Parser, OWL-S API) \*
- Erweiterte Fehlerbehandlung (OWL Parser, Jena Eyeball)
- WSDL Builder
  - Variation des Zielformats (WSDL-S)
  - Automatische Konfiguration (WA)
- Re-Engineering
  - xsltTransformationString (Re-Engineering, JXML2OWL)
  - Generierung von OWL-S 1.0 (translator 1.1 nach 1.0) \*
  - Validierung von OWL-S Definitionen mit API (broken) \*
- Integration von WSDL2Java (*top down*)

*\*) voluntarily contributions (OpenSource)*



# Future Work

- OWL-S Objektmodell (OWL-S Parser, OWL-S API) \*
- Erweiterte Fehlerbehandlung (OWL Parser, Jena Eyeball)
- WSDL Builder
  - Variation des Zielformats (WSDL-S)
  - Automatische Konfiguration (WA)
- Re-Engineering
  - xsltTransformationString (Re-Engineering, JXML2OWL)
  - Generierung von OWL-S 1.0 (translator 1.1 nach 1.0) \*
  - Validierung von OWL-S Definitionen mit API (broken) \*
- Integration von WSDL2Java (*top down*)

\*) *voluntarily contributions (OpenSource)*



# Future Work

- OWL-S Objektmodell (OWL-S Parser, OWL-S API) \*
- Erweiterte Fehlerbehandlung (OWL Parser, Jena Eyeball)
- WSDL Builder
  - Variation des Zielformats (WSDL-S)
  - Automatische Konfiguration (WA)
- Re-Engineering
  - xsltTransformationString (Re-Engineering, JXML2OWL)
  - Generierung von OWL-S 1.0 (translator 1.1 nach 1.0) \*
  - Validierung von OWL-S Definitionen mit API (broken) \*
- Integration von WSDL2Java (*top down*)

\*) *voluntarily contributions (OpenSource)*



# Durchführung der Evaluierung

- 1 Gegeben: OWLS-TC mit Matchmakingwerten des OWLS-MX
  - 29 Queries (Referenzdienste)
  - *Relevance Sets*
- 2 Translation OWL-S nach WSDL
  - OWL-S *Query* wird zu WSDL *Requirement*
  - Dienstbeschreibungen des *Relevance Set* werden *Candidates*
- 3 Ermittlung von Ähnlichkeitswerten mit dem WA (Ranking)
- 4 Vergleich der Matchmakingwerte (WA und OWLS-MX)





# Durchführung der Evaluierung

- 1 Gegeben: OWLS-TC mit Matchmakingwerten des OWLS-MX
  - 29 Queries (Referenzdienste)
  - *Relevance Sets*
- 2 Translation OWL-S nach WSDL
  - OWL-S *Query* wird zu WSDL *Requirement*
  - Dienstbeschreibungen des *Relevance Set* werden *Candidates*
- 3 Ermittlung von Ähnlichkeitswerten mit dem WA (Ranking)
- 4 Vergleich der Matchmakingwerte (WA und OWLS-MX)



# Durchführung der Evaluierung

- 1 Gegeben: OWLS-TC mit Matchmakingwerten des OWLS-MX
  - 29 Queries (Referenzdienste)
  - *Relevance Sets*
- 2 Translation OWL-S nach WSDL
  - OWL-S *Query* wird zu WSDL *Requirement*
  - Dienstbeschreibungen des *Relevance Set* werden *Candidates*
- 3 Ermittlung von Ähnlichkeitswerten mit dem WA (Ranking)
- 4 Vergleich der Matchmakingwerte (WA und OWLS-MX)



# Durchführung der Evaluierung

- 1 Gegeben: OWLS-TC mit Matchmakingwerten des OWLS-MX
  - 29 Queries (Referenzdienste)
  - *Relevance Sets*
- 2 Translation OWL-S nach WSDL
  - OWL-S *Query* wird zu WSDL *Requirement*
  - Dienstbeschreibungen des *Relevance Set* werden *Candidates*
- 3 Ermittlung von Ähnlichkeitswerten mit dem WA (Ranking)
- 4 Vergleich der Matchmakingwerte (WA und OWLS-MX)





## WSDL Analyzer (WA), Ranking

**Mapping results for car\_price\_service.wsdl**

Requirements: [Message: get\_PRICERequest, Parameter: \_CAR, SimpleType: CarType]  
Candidate: [Message: get\_PRICERequest, Parameter: \_AUTO, ComplexType: AutoType]

Requirements	Candidate	Ranking
Document:car_price_service	Document:auto_price_service	100.0 % car_price_service.wsdl
Service:CarPriceService	Service:AutoPriceService	85.07 % amount_of_moneycar_price_service.wsdl
Port:CarPriceSoap	Port:AutoPriceSoap	85.07 % car_priceauto_service.wsdl
Binding:SoapBinding	Binding:SoapBinding	85.07 % car_pricecolor_service.wsdl
PortType:CarPriceSoap	PortType:AutoPriceSoap	85.07 % car_pricequality_service.wsdl
Operation:get_PRICE	Operation:get_PRICE	85.07 % car_pricereport_service.wsdl
Message:get_PRICERequest	Message:get_PRICERequest	85.07 % car_taxedpriceprice_service.wsdl
Parameter:_CAR	Parameter:_AUTO	85.07 % cheapcar_price_service.wsdl
Message:get_PRICEResponse	Message:get_PRICEResponse	85.07 % expensivecar_price_service.wsdl
Parameter:_PRICE	Parameter:_PRICE	85.07 % vehicle_price_service.wsdl
ComplexType:PriceType	ComplexType:Price	77.61 % auto_price_service.wsdl
Element:currency	Element:current	70.14 % amount_of_moneycar_pricecompany_service.wsdl
Element:amount	Element:amount	70.14 % _RedFerrariprice_service.wsdl
		70.14 % _Toyotaprice_service.wsdl
		62.68 % auto_yearprice_service.wsdl
		52.23 % car_recommendedpriceindollar_service.wsdl
		52.23 % car_recommendedpriceineuro_service.wsdl
		52.23 % car_recommendedprice_service.wsdl
		52.23 % car_report_service.wsdl
		37.31 % fastcar_recommendedprice_service.wsdl

Next Mapping      Prev Mapping      Mapping      Best Mapping

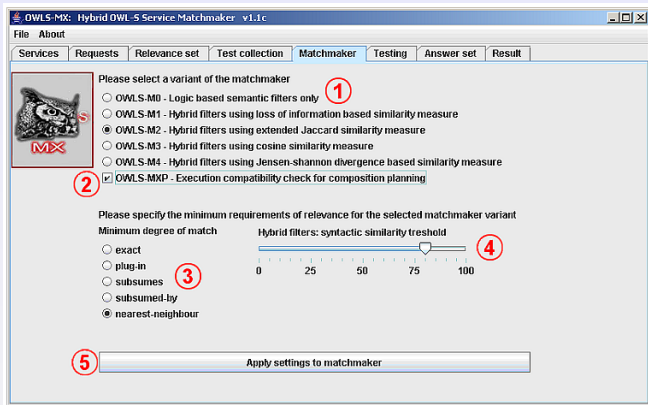
Current      Max

1      1

0      5200

Close

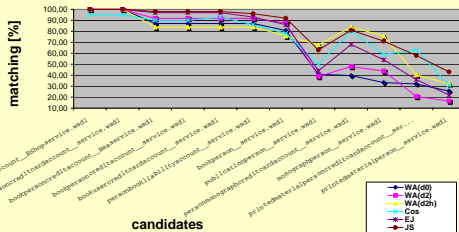
# Der OWLS-MX, Konfiguration



# Vergleich WA und OWLS-MX (Query 7)

Candidate service	WA(d0)	WA(d2)	WA(d2h)	Cos	EJ	JS	Semantic degree
bookpersoncreditcardaccount__BShopService.wsdl	100,00	100,00	100,00	95,00	100,00	100,00	Exact
bookpersoncreditcardaccount__Service.wsdl	100,00	100,00	100,00	95,00	100,00	100,00	Exact
bookpersoncreditcardaccount__Beaservice.wsdl	87,18	91,73	83,87	90,00	97,00	98,00	Plugin
bookpersoncreditcardaccount__Service.wsdl	87,18	91,73	83,87	90,00	97,00	98,00	Plugin
bookusercreditcardaccount__Service.wsdl	87,18	91,73	83,87	93,00	97,00	98,00	Failed
personbookliabilityaccount__Service.wsdl	87,18	91,73	83,87	86,00	93,00	96,00	Plugin
bookperson__Service.wsdl	80,78	87,60	75,80	78,00	86,00	92,00	Exact
publicationperson__Service.wsdl	40,99	38,82	67,74	50,00	44,00	63,00	Plugin
personmonographcreditcardaccount__Service.wsdl	39,71	47,91	83,87	80,00	68,00	81,00	Plugin
monographperson__Service.wsdl	33,31	43,78	75,80	59,00	54,00	71,00	Plugin
printedmaterialpersoncreditcardaccount__Service.wsdl	32,03	20,65	40,32	63,00	36,00	58,00	Plugin
printedmaterialperson__Service.wsdl	25,62	16,52	32,25	31,00	22,00	43,00	Plugin

bookpersoncreditcardaccount\_\_Service.wsdl



## Diskussion der Ergebnisse

- Matchmaking-Ergebnisse durchaus vergleichbar (tendenziell)
- Unterschiedlich gute Ergebnisse je nach Konfiguration
- Teils sehr große Schnittstellen (abhängig von Hierarchiegröße)
- Zyklen in XML Schema Definition

## Arbeit mit dem WSDL Analyzer

- Nicht alle validierten WSDL Beschreibungen können verarbeitet werden. Zyklen machen Probleme.
- Verbesserung des WA aufgrund des Vorgehensmodells.





## Diskussion der Ergebnisse

- Matchmaking-Ergebnisse durchaus vergleichbar (tendenziell)
- Unterschiedlich gute Ergebnisse je nach Konfiguration
- Teils sehr große Schnittstellen (abhängig von Hierarchiegröße)
- Zyklen in XML Schema Definition

## Arbeit mit dem WSDL Analyzer

- Nicht alle validierten WSDL Beschreibungen können verarbeitet werden. Zyklen machen Probleme.
- Verbesserung des WA aufgrund des Vorgehensmodells.



## Diskussion der Ergebnisse

- Matchmaking-Ergebnisse durchaus vergleichbar (tendenziell)
- Unterschiedlich gute Ergebnisse je nach Konfiguration
- Teils sehr große Schnittstellen (abhängig von Hierarchiegröße)
- Zyklen in XML Schema Definition

## Arbeit mit dem WSDL Analyzer

- Nicht alle validierten WSDL Beschreibungen können verarbeitet werden. Zyklen machen Probleme.
- Verbesserung des WA aufgrund des Vorgehensmodells.



## Diskussion der Ergebnisse

- Matchmaking-Ergebnisse durchaus vergleichbar (tendenziell)
- Unterschiedlich gute Ergebnisse je nach Konfiguration
- Teils sehr große Schnittstellen (abhängig von Hierarchiegröße)
- Zyklen in XML Schema Definition

## Arbeit mit dem WSDL Analyzer

- Nicht alle validierten WSDL Beschreibungen können verarbeitet werden. Zyklen machen Probleme.
- Verbesserung des WA aufgrund des Vorgehensmodells.



## Diskussion der Ergebnisse

- Matchmaking-Ergebnisse durchaus vergleichbar (tendenziell)
- Unterschiedlich gute Ergebnisse je nach Konfiguration
- Teils sehr große Schnittstellen (abhängig von Hierarchiegröße)
- Zyklen in XML Schema Definition

## Arbeit mit dem WSDL Analyzer

- Nicht alle validierten WSDL Beschreibungen können verarbeitet werden. Zyklen machen Probleme.
- Verbesserung des WA aufgrund des Vorgehensmodells.



## Diskussion der Ergebnisse

- Matchmaking-Ergebnisse durchaus vergleichbar (tendenziell)
- Unterschiedlich gute Ergebnisse je nach Konfiguration
- Teils sehr große Schnittstellen (abhängig von Hierarchiegröße)
- Zyklen in XML Schema Definition

## Arbeit mit dem WSDL Analyzer

- Nicht alle validierten WSDL Beschreibungen können verarbeitet werden. **Zyklen machen Probleme.**
- Verbesserung des WA aufgrund des Vorgehensmodells.



## Diskussion der Ergebnisse

- Matchmaking-Ergebnisse durchaus vergleichbar (tendenziell)
- Unterschiedlich gute Ergebnisse je nach Konfiguration
- Teils sehr große Schnittstellen (abhängig von Hierarchiegröße)
- Zyklen in XML Schema Definition

## Arbeit mit dem WSDL Analyzer

- Nicht alle validierten WSDL Beschreibungen können verarbeitet werden. **Zyklen machen Probleme.**
- **Verbesserung des WA aufgrund des Vorgehensmodells.**



# ENDE

*Vielen Dank für Ihre Aufmerksamkeit.*

