

# Intelligente Agenten für das Internet der Dienste

PD Dr. Matthias Klusch



## Themen

- Internet der Dienste
- Intelligente Agenten
- Semantische Webdienste

## Themen

- Internet der Dienste
- Intelligente Agenten
- Semantische Webdienste

## Vision

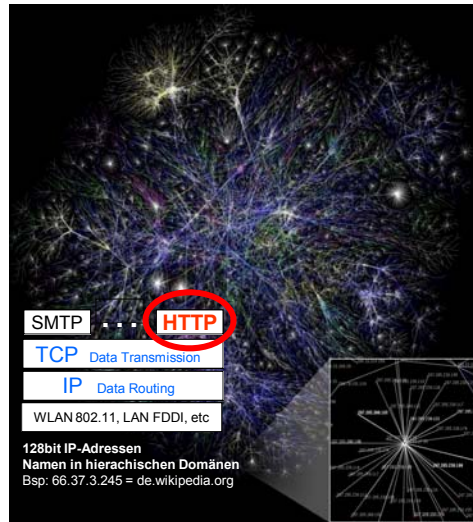
*“Das Internet der Dienste umfaßt*

*alle Dienstleistungen im Internet, die von jedem Nutzer  
als Anbieter, Vermittler oder Konsument **allgegenwärtig**  
verwendet, zusammengestellt und gehandelt werden können.”*

cf. Jorge Villasante, European Commission, 2008



## Internet



- **Das „Netz der Netze“**
  - ARPAnet 1969 (4), 1972 (23) USA
  - INTERNet 1982: Kommunikation **zwischen** Netzen (TCP/IP, 1974)
  - 2010: 1.9 Mrd Nutzer (WP 6.8 Mrd)
- **Dienste**
  - Rechnerfernzugriff (telnet, SSH)
  - Dateizugriff (FTP)
  - Netzadministration (SNMP, ICMP), Namensauflösung (DNS)
  - E-Mail (SMTP, IMAP, POP3)
  - VoIP Telefonie (SIP)
  - Chatdienste (IRC, Skype etc)
  - Usenet Diskussionsforen (NNTP)
  - P2P Tauschbörsen (Gnutella, etc.)
  - **World Wide Web (HTTP/S)**

13.12.2010, U Saarland

Klusch

5

## Web



„Network of sites in the Internet to search and retrieve documents with simplified addresses (URL) as embedded links using the hypertext transfer protocol (HTTP).“  
Sir Tim Berner-Lee

- 1989 @CERN: **HTTP, URL**
- 1992: **HTML**, 1993: MosaicX Browser
- 1994: W3C
- 2010: ca 30 Mrd Webseiten (w/o Deep Web)

- **Webadressen: URL - Universal Resource Locator**  
scheme://username:password@domain:port/path?query\_string#anchor
- **XHTML Webseiten; HTTP Protokolloperatoren:** GET, POST, PUT, DELETE



**Vielfalt von Anwendungen und Diensten im Web 1.0 und sozialen Web 2.0.**

13.12.2010, U Saarland

Klusch

6

## Webdienst



Ein **Webdienst** ist eine Softwarekomponente, die *per URL adressiert* ihre Funktionalität über eine *veröffentlichte XML Standard-basierte Schnittstelle* anbietet und *für beliebige Anwendungen* über ein im Internet verwendetes Protokoll *XML-Nachrichtenorientiert* und *programmatisch zugreifbar* ist. (Jeckle, 2004)

### ■ Dienstbasierte Webanwendung



Vorteile: Interoperabilität (XML), Wiederverwendbarkeit / „Lose Kopplung“

### ■ Architekturen von Webdiensten: REST & WSDL / SOAP

## Architektur I: REST-basierte Dienste

### REST – Representational State Transfer



### ■ HTTP Client-Server-basierte Dienstinteraktion

- **Dienst = Adressierbare Ressource im Web:** Dienst[.Operation#] URL
- **HTTP zustandslos:** Keine Veraltung von Dienstinteraktionszuständen
- **Repräsentation von Dienstinteraktionszustand:**  
Dienstaussagen (URLs von Ressourcen) und Infos  
in XML-Antwortnachricht an Client. (Cookies/Erweiterte Nachrichten)

## Beispiel: REST-Dienst „PartsDepot“

**Service:** PartsDepot

**Operation:** PartsDepot.Get\_Parts\_List

**URL:** [www.parts-depot.com/parts](http://www.parts-depot.com/parts)

**Method:** GET

**Parameters:** cid - Engine component ID

**Returns:** XML parts\_list

**Web API** des REST-Dienstes PartsDepot  
(Verwaltung von Maschinenteilen mit Teilleisten von Maschinenkomponenten)

**Verweis** auf REST-Dienst in XHTML Seite mit Mikroformat **hREST**:

```
<div class="service" id = "PartsDepot">
<div class="operation" id = "part"> operation
<code class="label"> Get_Parts_List </code> is invoked with
<span class="method"> GET </span> at
<code class="address"> http://www.parts-depot.com/{id}?cid=123
</code> <span class="input"> for component 123 </span>.
Returns <span class="output">parts list </span></div> </div>
```

**Antwortnachricht**  
in XML:

```
<?xml version="1.0"?>
<p:Parts xmlns:p="http://www.parts-depot.com" xmlns:xlink="http://www.w3.org/1999/xlink">
  <Part id="00345" xlink:href="http://www.parts-depot.com/parts/00345"/>
  <Part id="00346" xlink:href="http://www.parts-depot.com/parts/00346"/>
  <Part id="00347" xlink:href="http://www.parts-depot.com/parts/00347"/> </p:Parts>
```

## Beispiel: Heterogene REST-Dienstschnittstellen

**API Doc**

Overview

User Authentication

Submissions (Scrobbling)

Radio API

Playlists

Downloads

REST requests

XML-RPC requests

**API Methods**

**Album**

album.addTags

album.getInfo

album.get

album.rsm

album.sear

**Artist**

artist

artist

artist.getImages

artist.getInfo

artist.getPastEvents

artist.getPodcast

artist.getShouts

artist.getSimilar

artist.getTags

artist.getTopAlbums

artist.getTopFans

artist.getTopTags

artist.removeTag

artist.search

artist.share

artist.shout

**Last.fm Web Services**  
API | Feeds | Your API Account

**album.getInfo**

Get the metadata for an album on Last.fm using the album name or a musicbrainz id. See playlist:fech on how to get the album playlist.

e.g. [http://ws.audioscrobbler.com/2.0/?method=album.getInfo&api\\_key=b25b959554ed75058ac220b7b2e0a026...](http://ws.audioscrobbler.com/2.0/?method=album.getInfo&api_key=b25b959554ed75058ac220b7b2e0a026...)

**Params**

**artist** (Optional) : The artist name in question

**album** (Optional) : The album name in question

**mbid** (Optional) : The musicbrainz id for the album

**username** (Optional) : The username for the context of the request. If supplied, the user's playlist for this album is included in the response.

**lang** (Optional) : The language to return the biography in, expressed as an ISO 639 alpha-2 code.

**api\_key** (Required) : A Last.fm API key.

**Auth**

This service does not require authentication.

**Sample Response**

```
<album>
  <name>Believe</name>
  <artist>Cher</artist>
  <id>2024125</id>
  <mbid>11d03f8b-b8a9-18f4-81d1-7eb027c6dfb</mbid>
  <url>http://www.last.fm/music/Cher/Believe</url>
  <releasedate>6 Apr 1999, 00:00</releasedate>
  <image size="small">...</image>
  <image size="medium">...</image>
  <image size="large">...</image>
  <lastmod>47602</lastmod>
  <playcount>212991</playcount>
```

Service

Operation2

Input

Output

## Suche und Komposition

### ■ Verzeichnisse

 : 2.323 APIs

### ■ Suchmaschinen

- Seekda.com

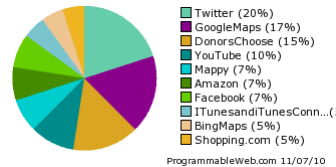
### ■ Dienstselektion

- **Manuelle** Suche und Relevanzbewertung „verständlicher“ REST-APIs

### ■ Komposition („Mashup“)

- **Manuelle** Kombination von REST-basierten Diensten
- Werkzeuge
  - XML Datentypabgleich (Parameters>Returns), -konversion

Google Maps  
Amazon AWS  
Ebay, Yahoo!  
Twitter, YouTube  
Facebook,  
NY Times, ...



ProgrammableWeb.com 11/07/10



## Beispiele von Mashups

### ■ Ortsanzeige von lokalen Ereignissen

- 15 Dienste:  
GoogleMaps, IP Location,  
Eventful, Facebook, flickr, last.fm,  
ThisNext, Twitter, etc.



### ■ Preisvergleich für Online-Shopping

- 3 Dienste:  
Amazon A9, Shopping.com  
Amazon Product Advertising



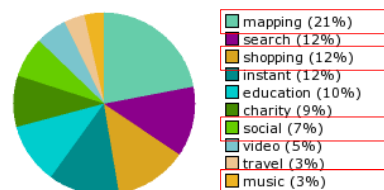
### ■ #LinS: GeoStatistik zu Kontakten in LinkedIn

- 2 Dienste: LinkedIn (socialgraph), GoogleCharts

### ■ ArtistCloud4iPhone&Android: Musiksuche für SmartPhones

- 5 Dienste: GoogleAppEngine, last.fm, MusicBrainz, YouTube, ChartLyrics Lyric

 : 5.351 Mashups



ProgrammableWeb.com 11/07/10

## Beispiele von Mashups

### Ortsanzeige von lokalen Ereignissen : 5.351 Mashups

#### 15 Dienste:

GoogleMaps, IP Location,  
Eventful, Facebook, flickr, last.fm,  
ThisNext, Twitter, etc.



### Preisvergleich für Online-Shopping

#### 3 Dienste:

A-

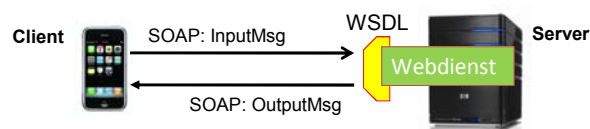
### Amazon.com

### AmazonCloud4iPhone&Android: Musiksuche für SmartPhones

#### 5 Dienste: GoogleAppEngine, last.fm, MusicBrainz, YouTube, ChartLyrics Lyric

**Keine automatische Suche + Komposition.  
Kein Standard.**

## Architektur II: SOAP-basierte Dienste in WSDL



### WSDL - Web Service Description Language

- Standardisierte XML-basierte Schnittstellenbeschreibung (2.0: W3C 2004)
- Funktional: Nur Signatur (E/A), Nicht-funktional: Name, Qualität, Preis, etc

### SOAP – Simple Object Access Protocol

- Standardisierte Nachrichtenorientierte Interaktion mit Diensten
- SOAP-XML-Nachrichtenformat (SOAP-Header, Body: E/A Nachrichten)

## Beispiel: WSDL-Dienst „PartsDepot“


```
<?xml version="1.0" encoding="utf-8" ?>
<description xmlns="http://www.w3.org/ns/wSDL" ... >
<types>
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:pd="http://www.parts-depot.com/schemas/pdc" ...>
    <xs:element name="cid" type="xs:string"/> ... </types>
</interface name="PartsListInterface">
  <operation name="GetPartsList"
    pattern="http://www.w3.org/ns/wSDL/in-out"...>
    <input messageLabel="In" element="xs:cid" />
    <output messageLabel="Out" element="pd:parts_list" />
  </operation> </interface>
<binding name="PListHTTPBinding"
  type="http://www.w3.org/ns/wSDL/http" interface="tns:PartsListInterface">
  <operation ref="tns:GetPartsList" http:method="GET"/> </binding>
<service name="PartsDepot" interface="tns:PartsListInterface">
  <endpoint name="PListHTTPEndpoint binding="tns:PListHTTPBinding"
    address="http://www.parts-depot.com/parts/"> </endpoint>
</description>
```

XMLS Datentypen der E/A Nachrichten des Dienstes

Operation/en (Signatur) des Dienstes

Bindung der Operation/en an Transfer-Protokoll

Ausführbare Operation (Dienstprogramm)

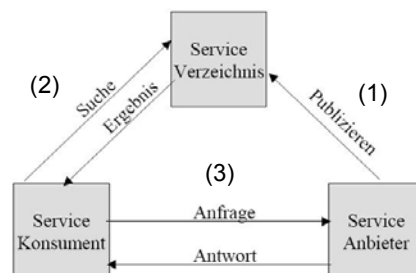
Andere WSDL-Dienste:  (bis 2009)   

13.12.2010, U Saarland

Klusch

15

## Suche



### Verzeichnisse

- UDDI – Universal Description, Discovery & Integration Standard
- [Xmethods.net](http://Xmethods.net)
- [webservicelist.com](http://webservicelist.com)
- [wsindex.org](http://wsindex.org), etc.

### Suchmaschinen

- [Seekda.com](http://Seekda.com) (WSDL, REST)

### Peer-To-Peer Dienstnetze

- B2B Plattformen (Avionik, Chemie, etc)

### Dienstselektion

- Relevanz von Diensten basiert auf **Textähnlichkeit** (strukturierte Schlagwortsuche über APIs)
- **Manuelle Wahl** aus Rangliste von Diensteanbietern



13.12.2010, U Saarland

Klusch

16

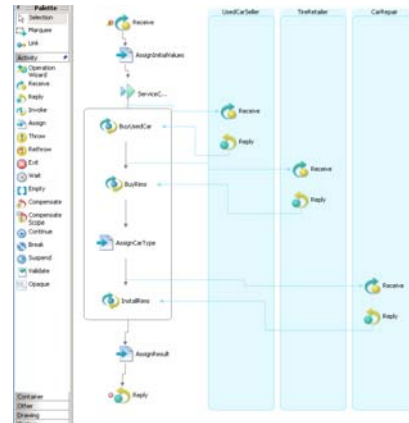


## Komposition

### Beschreibung

- Im OMG Standard **BPEL**  
(Business Process Execution Language)
- **Manuelle, graphische Komposition**  
in einem *BPEL-Diagramm*  
(Business Process Model Notation BPMN)
- Automatische Kompilierung in  
**ausführbares BPEL-Skript**
- Werkzeuge:
  - ActiveBPEL Designer
  - IBM WebSphere
  - ARIS Business Analyst, etc.

BPEL Diagramm in BPMN  
mit ActiveBPEL Designer 3 erstellt



## Beispiel: BPEL-Skript „InstallRimsOnUsedCar“

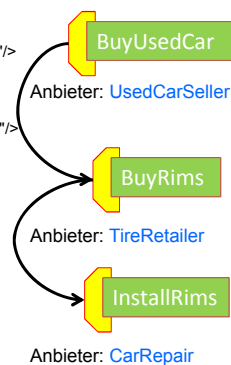
```

... <bpel:flow> <bpel:sequence name="ServiceComposition"> ...
<bpel:invoke
  inputVariable="orderCarMessage" name="BuyUsedCar" operation="OrderCar"
  outputVariable="orderResponse" partnerLink="UsedCarSeller" portType="ns4:CarProvider"/>

<bpel:invoke
  inputVariable="orderRimsRequest" name="BuyRims" operation="orderRims"
  outputVariable="orderRimsResponse" partnerLink="TireRetailer" portType="ns2:TireRetailer"/>
<bpel:assign name="AssignCarType">
  <bpel:copy> <bpel:from part="orderedCar" variable="orderResponse"/>
  <bpel:to part="car" variable="installMessage"/></bpel:copy>
  <bpel:copy> <bpel:from part="orderResponseDetails" variable="orderRimsResponse"/>
  <bpel:to part="rims" variable="installMessage"/></bpel:copy> ...

<bpel:invoke
  inputVariable="installMessage" name="InstallRims" operation="install"
  outputVariable="installResponse" partnerLink="CarRepair" portType="ns3:installRims"/>
</bpel:sequence>
... <bpel:copy> <
  <bpel:copy> <
  ... <bpel:copy> <
  <bpel:copy> <
  </bpel:flow>
</bpel:process>

```



**Nachrichtenorientierte Interaktion**  
zwischen **bekannten** Diensten von  
**kontraktierten** Diensteanbietern  
für gegebenen Geschäftsprozess.

## Beispiel: Verteilte Dienstorientierte Geschäftsprozesse

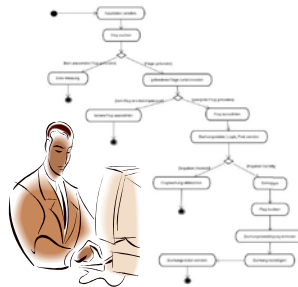
Service-Oriented Business Process Implementation (cf. SOA)



Production



Retail



Logistics



Finances



Subcontractors

13.12.2010, U Saarland

Klusch

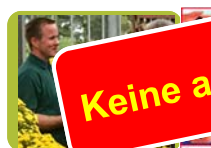
19

## Beispiel: Verteilte Dienstorientierte Geschäftsprozesse

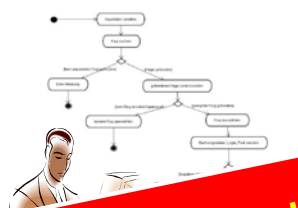
Service-Oriented Business Process Implementation (cf. SOA)



Production



Retail



Logistics



Finances



Subcontractors

**Keine automatische Suche + Komposition.**

13.12.2010, U Saarland

Klusch

20

## Herausforderung

**Automatische** und **intelligente** Koordination von Diensten.



*Persönliche digitale Assistenten*  
für Suche, Komposition, Verhandlung

Effiziente Suche und Komposition  
durch *besser maschinen-verständliche*  
Dienstbeschreibungen

## Herausforderung

**Automatische** und **intelligente** Koordination von Diensten.



*Persönliche digitale Assistenten*  
für Suche, Komposition, Verhandlung

Effiziente Suche und Komposition  
durch *besser maschinen-verständliche*  
Dienstbeschreibungen

## Themen

- Internet der Dienste
- **Intelligente Agenten**
- Semantische Webdienste

## Intelligenter Agent



George Luger

*“**Intelligente Agenten** sind zu einer aufgabenorientierten Problemlösung durch **autonome, reaktive und zielgerichtete** Anwendung von geeigneten **Methoden der Künstlichen Intelligenz (KI)** fähig.”*

### Relevante KI-Technologien für Agentenentwicklung:

- Wissensrepräsentation (Umgebung, Aktionen, Ziele)
- Logische Schlussfolgerung, Heuristische Lösungssuche (Planen)
- Maschinelles Lernen, Behandlung von unsicherem Wissen (Lernen)
- Intelligente Interaktion (Visualisierung, Natürlichsprachlicher Dialog)

***“Je mehr Verwendung von KI desto intelligenter der Agent”***

## Eigenschaften von Agenten

- **Autonom:** **Eigenverantwortliches** Handeln

- **Reaktiv**

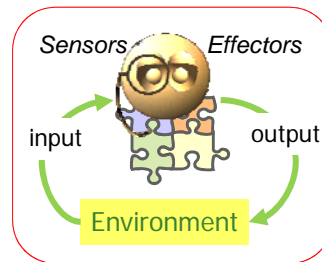
- kontinuierlich angepaßt (**lernend**) an Benutzer und Umgebung

- **Proaktiv**

- Eigeninitiativ und zielgerichtet **planend und entscheidend**

- **Kooperativ**

- Interaktion und Zusammenarbeit **mit anderen Agenten**



- **Softwareagenten**
- **Physische** Ausprägung: Roboter

## Beispiel: Automatisches Planen

- **Reaktives Planen von Aktionen**

*Unmittelbare oder sehr zeitnahe Reaktion auf unvorhergesehene Ereignisse in der Umgebung.*



- **Reiner Reflex:** Ereignisorientierte „Reiz-Reaktion“ Regeln mit „Vergessen“ (**ohne internen Zustand**)
- **Reflex:** Reiz-Reaktion-Regeln mit kontinuierlich gespeichertem Wissen über Umgebung (**mit internem Zustand & Weltmodell; „Erinnerung“**)
- **Planmusterinstanziierung** über Zustand („**Planning from 2nd Principles**“)

*Lernend:* Lernen der Nützlichkeit von Aktionen für individuelle Lernziele über internem Zustand und (partiell) beobachteter Umgebung.



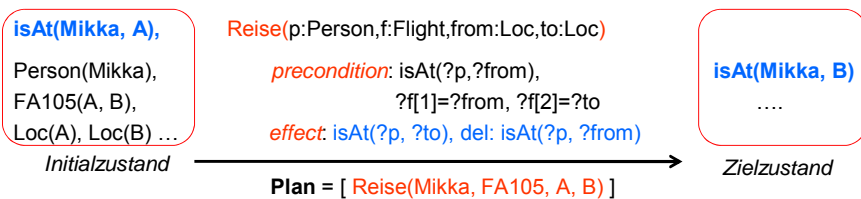
Schwarm mit reinen Reflexen ohne Kommunikation (Animationsfilmvorstudie von BioGraphics)

## Beispiel: Automatisches Planen

- **Deliberative Planerzeugung** („Planning from 1st Principles“)

- Beispiel: Zustandsbasiertes Vorwärtsplanen

*Finde aus gegebener Menge von Operatoren eine Aktionssequenz (Plan), dessen simulierte Ausführung in einem initialen Zustand (Fakten) einen gegebenen Zielzustand (Fakten) erreicht.*



- **Benutzerinteraktives Planen, Hybrides Planen, Verteiltes Planen ...**

13.12.2010, U Saarland

Klusch

27

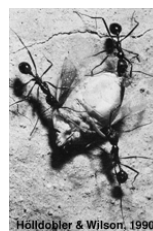
## Multiagentensysteme

- **Systeme interagierender Agenten für verteilte Problemlösung**

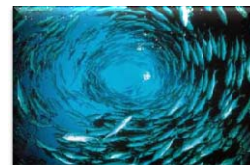
- **Hierarchische** Aufgabenverteilung (zB. Kontraktnetz)
- **Emergentes** Lösungsverhalten (Schwarm)

- **Koordination im System**

- **Kommunikation** zwischen Agenten (**mit/ohne**)
- **Kooperationsmodelle**
  - Gemeinsames Planen (Joint Planning)
  - Nutzenorientierte Verhandlungen, Auktionen, Koalitionen, Marktplatz, etc.



Hölldobler &amp; Wilson, 1990



Schwarm *mit* Kommunikation („Mundpropaganda-Effekt“)

13.12.2010, U Saarland

Klusch

28

## Prominente Anwendungsbereiche



### Transport & Logistik

*Dezentrale, situative Optimierung von Transportrouten, Warenlagerverwaltung, DFKI: Flottenmanagement für Speditionen Konz, Fixemer (Saarland)*



### Produktionssteuerung

*Dezentrale, situative Optimierung von Produktionsabläufen DFKI: für Saarstahl AG*

### Elektronischer Handel

*Dezentrale, rationale Verhandlungen, Online-Auktionsteilnahmen DFKI: Holzauktion – CASA Saarland; Intelligente Online-Einkaufsassistenten*



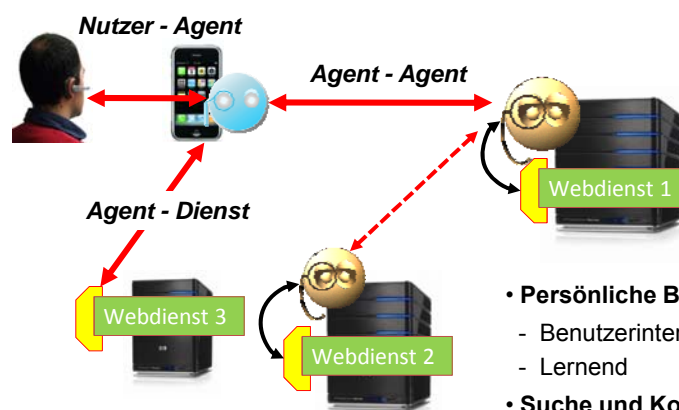
### Edutainment

*Intelligente Avatare in 3D Welten DFKI: ISReal Projekt*



## Agenten und Webdienste

### • Arten der agentenbasierten Dienstbereitstellung:

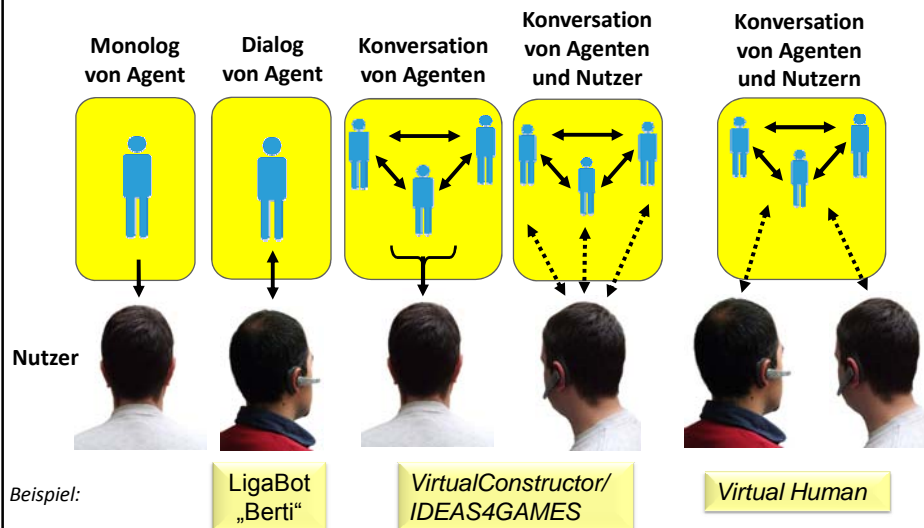


### • Persönliche Bereitstellung

- Benutzerinteraktiv
- Lernend

### • Suche und Komposition

## Persönliche Dienstbereitstellung: Benutzerinteraktiv



13.12.2010, U Saarland

Klusch

31

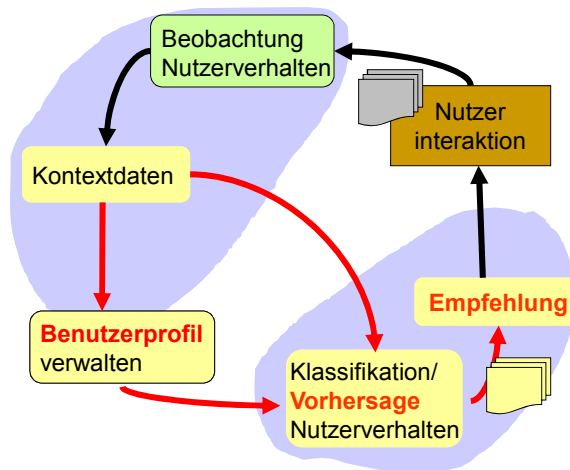
## Persönliche Dienstbereitstellung: Lernend

### Kontextdaten (Nutzer, Ressource)

- Ressourcenbewertung
  - explizit
  - implizit (Aktionen)
- Anfragenhistorie
- Ressourceneigenschaften
  - Themen, Ortsgebunden, etc
  - Relation zu anderen Ressourcen

### Präferenzen über Ressourceneigenschaften

- Langzeitpräferenzen
- Vs. Kurzzeitpräferenzen



Präferenzbasierte Empfehlung von (neuen) Ressourcen für Nutzer

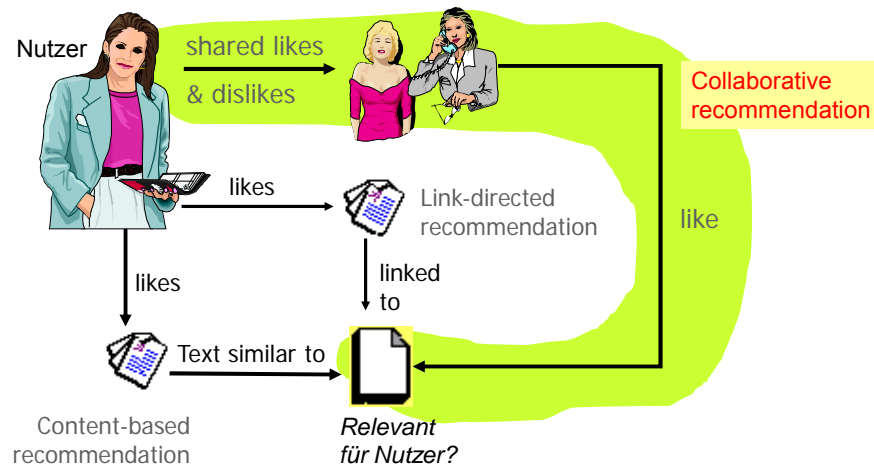
13.12.2010, U Saarland

Klusch

32



## Beispiel: Kollaborative Empfehlung (Social Filtering)



13.12.2010, U Saarland

Klusch

33

## Beispiel: Kollaborative Empfehlung (2)

Voraussage der

- **Bewertung**  $r_{xj}$  eines Artikels (**item**)  $j$  aus einer Menge  $J$  durch **Benutzer**  $x$
- basierend auf Bewertungen  $r_{ij}$  **anderer Benutzer**  $i$  von Artikeln **aus**  $J$

$$r_{xj} = \sum_{i \neq x} w(x, i) \cdot (r_{ij} - \hat{r}_{ij}), \quad \hat{r}_{ij} = \frac{1}{|J|} \sum_J r_{ij}$$

Korrelation der Präferenzen/  
Bewertungsvektoren  $r_x, r_i$  mit Kosinusmaß:

$$w(x, i) = \cos(\vec{r}_x, \vec{r}_i) = \frac{\sum_{j \in J} r_{x,j} \cdot r_{i,j}}{\sqrt{\sum_{j \in J} (r_{x,j})^2 \cdot \sum_{j \in J} (r_{i,j})^2}}$$

Pearson-Korrelation (GroupLens, 1994):

$$w(x, i) = \frac{\sum_{j \in J} [r_{xj}^+ - r_{xj}^-] \cdot [r_{ij}^+ - r_{ij}^-]}{\sqrt{\sum_{j \in J} [r_{xj}^+ - r_{xj}^-]^2 \cdot \sum_{j \in J} [r_{ij}^+ - r_{ij}^-]^2}}$$

$r_{xj}^{+/-}$  positive/negative ratings of user  $x$

13.12.2010, U Saarland

Klusch

34

### Beispiel: Kollaborative Empfehlung (3)

Durch (explizite/implizite) Nutzerbewertung erlernte

**Nutzer-Artikel-Präferenzmatrix:**

User/Item	Star Wars	Jurassic Park 2	Terminator	Godzilla	Alien I
User <sub>1</sub>	8	6	8	1	9
User <sub>2</sub>	3	1	8	0	-
User <sub>3</sub>	1	9	2	8	
User <sub>4</sub>	2	0	10	1	7
...					

.....

Nutzer 4 mit höchster Präferenzkorrelation ( $w_{24}$ )

Kollaborative Empfehlung: "Alien I":  $r_{25} = 8.4$

Einfache Empfehlung: "Alien I":  $r_{25} = r_{45} = 7$

Aufsteigende Bewertungsskala 1 – 10; Nicht bewertet: -

### Beispiel: Kollaborative Empfehlung (4)

Empfehlungen für Sie > Ihre persönliche Seite

**FÜR SIE DOKUMENTIERT**  
Dr. Matthias Klusch, hier sind Ihre kürzlich angesehenen Artikel. (Wenn Sie nicht Dr. Matthias Klusch sind, [klicken Sie bitte hier.](#))

**Kürzlich angesehen Produkte**

☒ [Intelligent Agents for Data Mining and Information Retrieval](#) von Masoud Mohammadian (Herausgeber)

☒ [Die Zeitmaschine](#) von Herbert G. Wells

☒ [Quantum Computing verstehen](#) von Matthias Homeister, und andere

**Ihre letzten Suchanfragen**

☒ [intelligent information agents](#)

☒ [quantum computing](#)

[Aktualisieren](#) [Alle löschen](#)

Möchten Sie Ihre persönliche Seite und die Liste kürzlich angesehener Artikel ändern, [klicken Sie hier.](#)

**Brachen Sie Hilfe?**

Für weitere Informationen t  
Sie unsere [Hilfeseiten.](#)

Kunden, die [Die Zeitmaschine](#) gekauft haben, bestellten auch:



[Der Krieg der Welten](#)  
von Herbert G. Wells, und andere  
Erscheinungsdatum: 1. Mai 2005

Preis: ~~EUR 9,90~~ **Gebraucht & neu** ab EUR 5,50

☐ Gehört mir ☐ Kein Interesse ☒ [Bewertung](#)

**Kollaborative Kaufempfehlung**



[1984](#)  
von George Orwell, Michael Wal  
Erscheinungsdatum: 1. Januar 2004

Preis: **E** Kunden, die Artikel gekauft haben, welche Sie sich kürzlich angesehen haben, kauften auch:

☐ Gehört mir ☐ Kai



[Fahrenheit 451](#)  
von Ray  
Erscheinungsdatum: 1. Januar 2004

Preis: **E**



[Fischer Taschenbücher, Bd. 26, Schöne neue Welt](#)  
von Aldous Huxley



[1984](#)  
von George Orwell

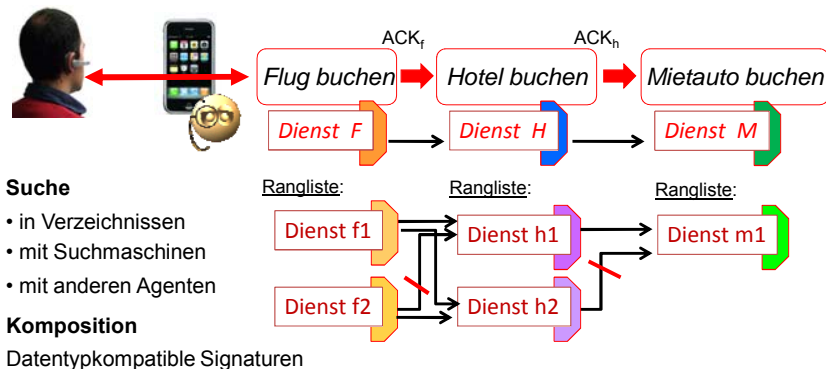
**Individuelle Bewertung (Präferenzupdate)**

**Generelles Problem:**  
„Cold Start“ (ohne Predictor) und „Popularity Bias“.

[weitere Artikel dieser Art](#)

## Suche und Komposition von Webdiensten

Nutzeranfrage: „Flug nach Berlin mit Hotel und Mietwagen.“



Präzision der Suche bestimmt durch Selektion:  
**Dienstabgleich + Relevanzbewertung**

## Beispiel: Struktureller WSDL-Dienstabgleich

**Input:** WSDL Dienste S, Q

**Output:** Struktureller Ähnlichkeitswert  $\text{sim}(S, Q) \in [0, 1]$

- **Rekursive Berechnung struktureller Ähnlichkeit von XML-Graphen**

$T_S, T_Q$  von S, Q:  $\text{sim}(T_S, T_Q) = \text{sim}(\text{root}(T_S), \text{root}(T_Q))_{\text{norm}}$

$$\text{sim}(a, b) = \begin{cases} \omega_1 * \text{sim}_{\text{name}}(l(a), l(b)) + \\ \omega_2 * \max\{\sum_k \text{sim}(n_i, m_j) \mid (a, n_i), (b, m_j) \in E, m_j = \varphi_k(n_i), \varphi_k \text{ inj.}\}, & \text{if } l(a), l(b) \notin D \\ \text{sim}_{\text{type}}(l(a), l(b)), & \text{if } l(a), l(b) \in D \end{cases}$$

$\text{sim}_{\text{name}}$  : **Textähnlichkeit der Bezeichner von Graphknoten**  
(WordNet-distance + keyword string matching)

$\text{sim}_{\text{type}}$  : **Datentypkompatibilität**  $\in [0, 1]$  (Look-up table); Gewichte  $w_1 + w_2 = 1$

[Zinnikus, Rupp & Fischer, Proceedings 2nd WS Web Services and Interoperability, 06]



## Herausforderung

**Automatische** und **intelligente** Koordination von Diensten.



*Persönliche digitale Assistenten*  
für Suche, Komposition, Verhandlung

Effiziente Suche und Komposition  
durch *besser maschinen-verständliche*  
Dienstbeschreibungen

## Themen

- Internet der Dienste
- Intelligente Agenten
- **Semantische Webdienste**
- Ausblick

## Semantisches Web

Idea: „To make the semantics of Web resources *machine-understandable*.“



2001

Tim Berners-Lee,  
Ora Lassila, Jim Hendler



.. but how ?

## Semantisches Web

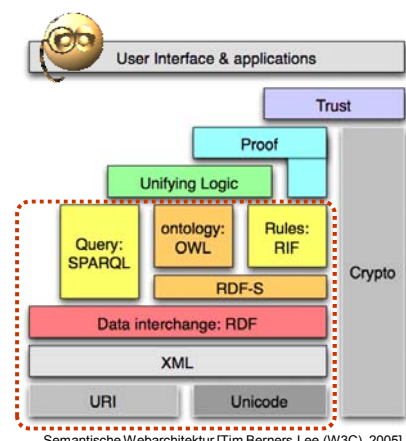
### Logikbasiert semantische Annotation:

- Referenzen auf Konzepte und Relationen in **formal-logischen Ontologien**
- **Standard** Logikbasierte Ontologiesprachen: RDFS, OWL2

### Hinreichend maschinen-verständlich:

- **Standard logische Inferenzverfahren** über Ressourcen und deren Relationen (insbes. Erwerb **impliziten Wissens**).
- Automatische **Planung** über Ressourcen

→ Intelligente Anwendungen im Sinne der KI durch **Softwareagenten**



Semantische Webarchitektur [Tim Berners-Lee (W3C), 2005]

## Beispiel: Semantische Annotation und Suche

Verweise auf RDFS/OWL2 Ontologien in HTML-Seite mit Mikroformat RDFa.

www.mercedes-retailer.com

```
<html xmlns="http://www.mo.com/O2">
<div about="#M37" typeOf="#Coupe">
  [M37] Rotes Mercedes Coupé:
  BJ 2009, Guter Zustand, 2.Hd, Preis: ...
</div> ...
</html>
```

**Coupe**  $\sqsubseteq$  (and **O1:Car** (=3 hDoors))  
(M37 typeOf Coupe)(M37 hColor red)  
(M37 hBrand Mercedes)

Ontologie O2

Anfrage: „Roter Mercedes 3-Türer?“

```
SELECT ?m
WHERE ?m isA Car.
```

M37 (Coupe)  
www.mercedes-retailer.com

Ontologie O1

**Car**  $\sqsubseteq$  (and Vehicle (<=5 hDoors)  
(=1 hColor.Colors) hBrand.Manufact ..),...

**Coupe**  $\sqsubseteq$  (and **Car** (=3 hDoors))

(M37 hColor red) (M37 typeOf Coupe)  
(M37 hBrand 'Mercedes')  
(M37 hDoors 3), (M37 isA Car)

**Suche und Speicherung**  
mit **logischer Inferenz** (Materialisierung).

s.a. RVL „Wissensbasierte Suche“ von PD Dr Ralf Schenkel.

13.12.2010, U Saarland

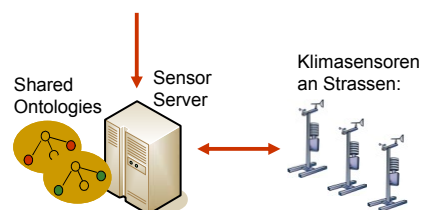
Klusch

45

## Beispiel: Semantisches Sensorweb Mashup [Sheth et al. 2009]



„Roads in Ohio with **potentially\_icy\_conditions**  
within the period of May 1 – 2, 2009.“



Logische Inferenz: **Potentially\_Icy\_Condition**

$\leftarrow$  measured(?obs, ?temp), Temperature(?temp),  
temp\_value(?temp, ?tval), lessThanOrEqual(?tval, 0),  
unit\_of\_measurement(?temp, "celsius").

Temporal-logische Inferenz:

„within the period of“

?time: 2009-05-01T05:00:00  $\in$  [01-05-09, 02-05-09]

Aktuelle semantisch annotierte Sensordaten

```
<swe:Time definition="urn:ogc:def:phenomenon:time"
uom="urn:ogc:def:unit:date-time">
  <sa:swe rdfs:about="?time" rdfs:instanceof="time:Instant">
    <sa:sml rdfs:property="xs:date-time"/> </sa:swe>
  </swe:Time>
  <swe:value name="weather-data-time"> 2009-05-01T05:00:00
  </swe:value>
```

Plus annotierte Temperaturdatenwerte für ?temp

13.12.2010, U Saarland

Klusch

46

# Semantischer Webservice

## Formale Ontologien

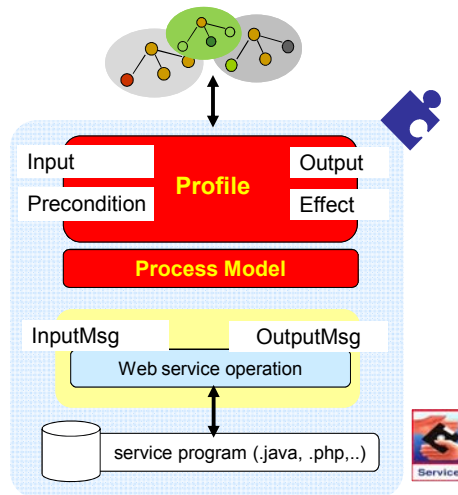
(in RDFS, OWL2, WSMO, SWRL etc)

## Semantischer Dienst

(in OWL-S, WSMO, SAWSDL, SAREST)

## Webdienst

(in WSDL, REST)



13.12.2010, U Saarland

Klusch

47

## Beispiel: Semantischer Dienst „BookFlight“ in OWL-S



Ontology O1

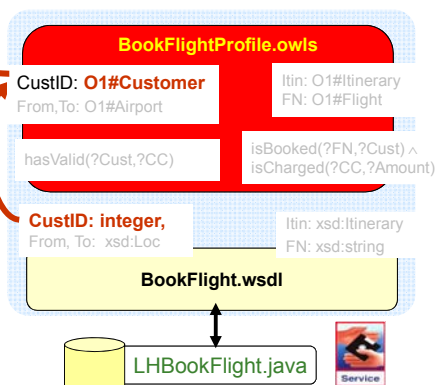
Concepts and Relations:

Person  $\sqsubseteq$  (and Human), ...

**Customer  $\sqsubseteq$  (and Person ( $\geq 1$  hasValid.CreditCard) ...)**

Facts: Customer(**Hans**), Flight(LH951), CreditCard(VISA), hasValid(Hans,VISA\*\*7)..

In OWL2-DL



- **Signatur** (I/O)
- **Spezifikation** (P/E)
- **Beschränkung** auf formale Ontologien in OWL
- **Kein Standard**

13.12.2010, U Saarland

Klusch

48



## Beispiel: Semantischer Dienst in SAWSDL

SAWSDL: Semantic Annotations for WSDL and XML Schema.

```
<wsdl xmlns:sawSDL="http://www.w3.org/2002/ws/sawSDL">
  <interface name="BookFlightInterface">
    <operation name="BookFlightOp" ...>
      <input messageLabel="CustID" element="xsd:integer"
        sawSDL:modelReference="O1#Customer"/>
      <output messageLabel="FN" element="xsd:string" /> ...
    </operation> </interface> <binding ... />
  </wsdl>
```

**sawSDL:modelReference = "O1#Customer"/>**

```
<output messageLabel="FN" element="xsd:string" /> ...
</operation> </interface> <binding ... />
<service name="BookFlight" interface="xs:BookFlightInterface">
  <endpoint ... address="http://www.airlineX.com/booking/" /> ...
</service>
```

BookFlight.sawSDL

**BookFlight Profile**

CustID: **O1#Customer**      Itin: O1#Itinerary  
From, To: O1#Airport      FN: O1#Flight

CustID: integer,      Itin: xsd:Itinerary  
From, To: xsd:Loc      FN: xsd:string

BookFlight.wSDL

- Nur **Signatur**annotation
- **Keine Beschränkung** auf formale Ontologien wie z.B. in OWL-S
- **W3C Standard 2007**

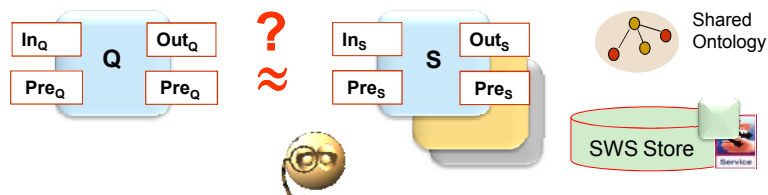
## Semantische Suche von Diensten

### ■ Spezielle Suchmaschinen

- Sousuo, Semantic Service Search Engine S3E
- Meta-Suche nach semantischen Dienstdateien im Web

### ■ Semantische Dienstvermittler („Service Matchmaker Agents“)

- Semantischer Abgleich von Dienstanfrage und registrierten Diensten
- Relevanzbewertung; Ausgabe relevanter, zugeordneter Webdienste



## Beispiel: Semantischer Dienstvermittler OWLS-MX2

**Input:** Set  $S$  of services, query  $Q$  in OWL-S, text similarity threshold  $\alpha \in [0, 1]$

Ontology KB in OWL-DL, Text index  $\mathcal{T}$

**Output:** Rank list of relevant services  $S$

For all services  $S$  in  $S$  and query  $Q$  do:

**Logical and text similarity matching of semantic signatures of  $S$ ,  $Q$ :**

$\text{Match}(S, Q, KB, \mathcal{T})_{\text{Hybrid}}$  if  $\text{Match}(S, Q, KB)_{\text{Logic}}$  and  $(\text{Match}(S, Q, \mathcal{T})_{\text{Syn}} > \alpha)$

Return: List  $[S, \text{Match}(S, Q, KB)_{\text{Hybrid}}]$  of relevant services  $S$

ranked by degree of (1)  $\text{Match}(S, Q, KB)_{\text{Logic}}$  (2)  $\text{Match}(S, Q, \mathcal{T})_{\text{Syn}} \in [0, 1]$

[Klusch, Fries, Sycara: Journal of Web Semantics, 7(3), 2009]



**Most popular semantic web service matchmaker world wide!**

More than 41.500 downloads @ semwebcentral.org (as of Dec 12, 2010)

## Logikbasiert Semantischer Dienstsingaturabgleich

$\text{Match}(S, Q, KB)_{\text{Logic}} \in \{\text{Exact}, \text{PlugIn}, \text{Subsumes}, \text{Subsumed-By}, \text{LogicalFail}\}$

1. **Exact ( $S, Q$ ):**

$(\forall C \in \text{In}_S \exists C' \in \text{In}_Q: C' \sqsubseteq C) \wedge (\forall D' \in \text{Out}_Q \exists D \in \text{Out}_S: D' = D)$

2. **Plug-In ( $S, Q$ ):**

$(\forall C \in \text{In}_S \exists C' \in \text{In}_Q: C' \sqsubseteq C) \wedge (\forall D' \in \text{Out}_Q \exists D \in \text{Out}_S: D' \sqsubseteq_1 D)$

3. **Subsumes ( $Q, S$ ):**

$(\forall C \in \text{In}_S \exists C' \in \text{In}_Q: C' \sqsubseteq C) \wedge (\forall D' \in \text{Out}_Q \exists D \in \text{Out}_S: D' \sqsupseteq D)$

4. **Subsumed-By ( $Q, S$ ):**

$(\forall C \in \text{In}_S \exists C' \in \text{In}_Q: C' \sqsubseteq C) \wedge (\forall D' \in \text{Out}_Q \exists D \in \text{Out}_S: D' \sqsubseteq D)$

5. **LogicalFail**



$C, C, D, D'$  in shared  
OWL2-DL ontology KB



Logical subsumption:  $KB \models (C \sqsubseteq D)$  iff  $KB \cup \{(C \sqcap \neg D)(a)\}$  unsatisfiable; Complexity: NEXPTIME

## Beispiel: Logisch Positiver Dienstabgleich

From standard test collection OWLS-TC4

### Dienstanfrage

Input

x: Patient

Output

y: Physician

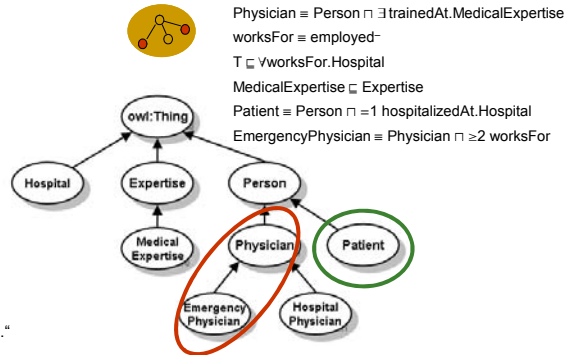
„Returns physician for given patient.“

### Relevanter Dienst

a: Patient

b: Emergency Physician

„Returns emergency physician for given patient.“



$\text{Match}(S, Q, KB)_{\text{Logic}} = \text{Plug-In}(S, Q)$  „relevant“

13.12.2010, U Saarland

Klusch

53

## Beispiel: Falscher Logisch Positiver Dienstabgleich

### Dienstanfrage

x: Train

z: CreditCard

Q

y: TrainTicket

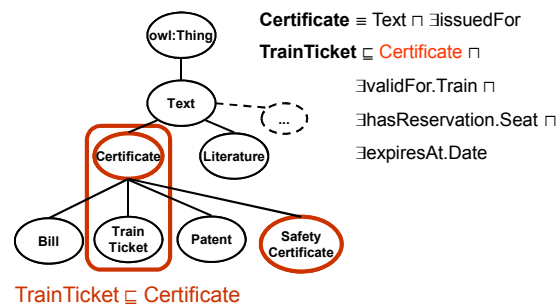
„Online purchase of train ticket with credit card.“

### Nicht relevanter Dienst

a: Train

b: Certificate

„Issues safety certificate for given train.“



$\text{Match}(S, Q, KB)_{\text{Logic}} = \text{Subsumed-By}(Q, S)$  „relevant“

**Geringe Textähnlichkeit** der Ausgabekonzepte in der Ontologie:

$\tau(\text{Certificate}) = (\text{and } \text{Certificate } \text{Text})$

$\tau(\text{TrainTicket}) = (\text{and } \text{TrainTicket } \text{Certificate } \text{Text} \text{ ex issuedFor ex validFor.Train ex hasReservation.Seat ex expiresAt.Date})$

13.12.2010, U Saarland

Klusch

54

## Beispiel: Falscher Logisch Negativer Dienstabgleich

### Dienstanfrage



„Returns hospital physician for given patient.“

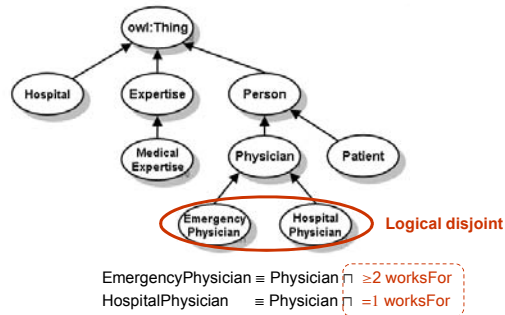
### Relevanter Dienst



„Returns emergency physician for given patient.“



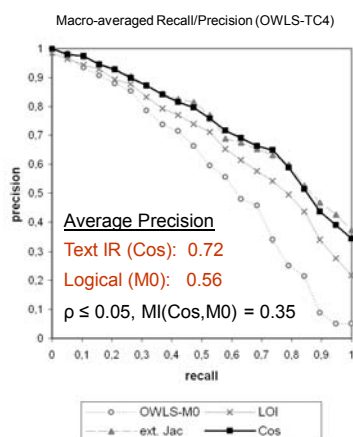
Match(S,Q,KB)<sub>Logic</sub> = Fail „irrelevant“



**Hohe Textähnlichkeit** der Ausgabekonzepte in der Ontologie:

$\tau(\text{EmergencyPhysician}) = (\text{and EmergencyPhysician Physician Person trainedAt MedicalExpertise worksFor atleast 2 Hospital})$   
 $\tau(\text{HospitalPhysician}) = (\text{and HospitalPhysician Physician Person trainedAt MedicalExpertise worksFor exactly 1 Hospital})$

## Nicht-Logikbasiert Semantischer Dienstabgleich Mit Textähnlichkeiten



### Semantische Dienstannotationen als Texte:

Repräsentation logischer Konzeptdefinitionen von semantischen Dienstprofilen als Häufigkeitsgewichtete (TF-IDF) Schlagwortvektoren  $S_{in}$ ,  $S_{out}$  (geg. Index I)

### Textähnlichkeit von Eingabe-/Ausgabekonzepten:

$$\text{Sim}(S, Q)_j = \frac{Q_j \cdot S_j}{\|Q_j\|_2 + \|S_j\|_2} \in [0,1], j \in \{in, out\}$$

### Textähnlichkeitsbasierter Dienstabgleich:



$$\text{Match}(S, Q, I)_{\text{syn}} = \frac{\text{Sim}(Q, S)_{in} + \text{Sim}(Q, S)_{out}}{2} \in [0,1]$$

## Hybrid Semantischer Dienstabgleich: Logik + Text

1. Exact (S,Q):  $(\forall C \in \text{In}_S \exists C' \in \text{In}_Q: C' \sqsubseteq C) \wedge (\forall D' \in \text{Out}_Q \exists D \in \text{Out}_S: D' = D)$
2. Plug-In (S,Q):  $(\forall C \in \text{In}_S \exists C' \in \text{In}_Q: C' \sqsubseteq C) \wedge (\forall D' \in \text{Out}_Q \exists D \in \text{Out}_S: D' \sqsupseteq_1 D)$   
 $\wedge \text{Match}(S,Q,I)_{\text{Syn}} > \alpha$
3. Subsumes (Q,S):  $(\forall C \in \text{In}_S \exists C' \in \text{In}_Q: C' \sqsubseteq C) \wedge (\forall D' \in \text{Out}_Q \exists D \in \text{Out}_S: D' \sqsupseteq D)$   
 $\wedge \text{Match}(S,Q,I)_{\text{Syn}} > \alpha$
4. Subsumed-By (Q,S):  $(\forall C \in \text{In}_S \exists C' \in \text{In}_Q: C' \sqsubseteq C) \wedge (\forall D' \in \text{Out}_Q \exists D \in \text{Out}_S: D' \sqsubseteq D)$   
 $\wedge \text{Match}(S,Q,I)_{\text{Syn}} > \alpha$
5. Syn(S,Q):  $\text{Match}(S,Q,I)_{\text{Syn}} > \alpha$  else Fail

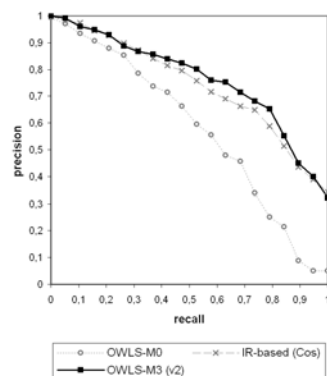


$\text{Match}(S,Q,KB,I)_{\text{Hybrid}} = (h, s)$

**Logikbasiert**  $h \in \{\text{Exact, Plug-In, Subsumes, Subsumed-By, Syn, Fail}\}$ ,

**Syntaktisch**  $s = \text{Match}(S,Q)_{\text{Syn}} \in [0,1]$

## Experimentelle Evaluierung von OWLS-MX2



### Präzision der Filtervarianten

- Logikbasiert: 0.41
- Textähnlichkeit: 0.74
- Hybrid (Logik  $\wedge$  Text): **0.9**

**Antwortzeit:** 5.4 sec

**International Semantic Service Selection Contest:**  
[www.dfki.de/~klusch/s3](http://www.dfki.de/~klusch/s3)

**Download @:**  
[projects.semwebcentral.org/projects/owlsmx](http://projects.semwebcentral.org/projects/owlsmx)

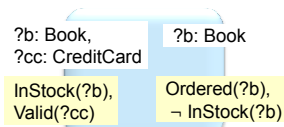


**Best Service  
Matchmaker**

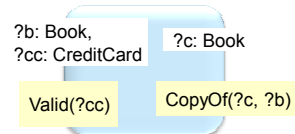
**2008**

## Logikbasiert Semantischer Dienstprofilabgleich (IOPE)

Dienstanfrage: *Book\_Purchasing*

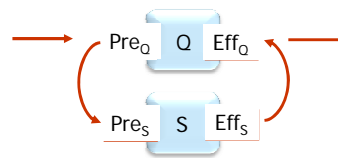


Nicht relevanter Dienst: *Book\_Copying*



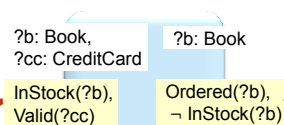
- IO: Logischer Signaturabgleich  $\text{Exact}(S, Q)$
- PE: Logischer Spezifikationsabgleich (Plug-In)

$$KB \models (\text{Pre}_Q \Rightarrow \text{Pre}_S) \wedge (\text{Eff}_S \Rightarrow \text{Eff}_Q)$$

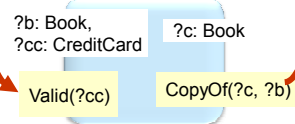


## Logikbasiert Semantischer Dienstprofilabgleich (IOPE)

Dienstanfrage: *Book\_Purchasing*



*Book\_Copying*



- IO: Logischer Signaturabgleich  $\text{Exact}(S, Q)$
- PE: Logischer Spezifikationsabgleich (Plug-In)

$$KB \models (\text{Pre}_Q \Rightarrow \text{Pre}_S) \wedge (\text{Eff}_S \times \text{Eff}_Q)$$



$\text{Match}(S, Q, KB)_{\text{Logic-PE}}$ : IO *and* PE  
= Fail „irrelevant“

Viele andere Arten von semantischem Dienstabgleich:

- Strukturelle Ähnlichkeit zugeordneter WSDL Dienste
- Strukturelle Konzeptähnlichkeit über Pfaddistanzen in Ontologie, usw.

## Herausforderung für semantische Suche



Wie kann ein Vermittleragent  
*beliebige* Arten von semantischem Dienstprofilabgleich  
 für eine *optimale Präzision* der semantischen Suche  
*automatisch kombinieren* ?

**Let the matchmaker agent just LEARN how to do it!**

## iSeM: Lernender Hybrid Semantischer Dienstvermittler

**Training:** Menge *positiver/negativer Beispiele* für semantische Relevanz

- Pro Beispiel (Q,S,1/-1) ein N-dimensionaler *Dienstabgleichsvektor* (q,s)  
 mit *Resultaten von allen geg. semantischen (IOPE) Abgleichsfiltern*
- *Pruning*: Regelbasierte Reduktion logischer Abgleichsfehler
- Erlerne binären Relevanzklassifizier  $d(q,s)$  mit Ähnlichkeitswert  $dr$   
 (Maschinelles Lernverfahren: SVM – Unterstützungsvektormethode)

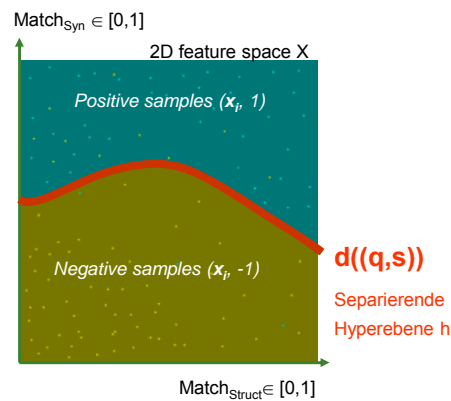


**Selektion:** Gegebene (evtl. unbekannte) Dienstanfrage Q

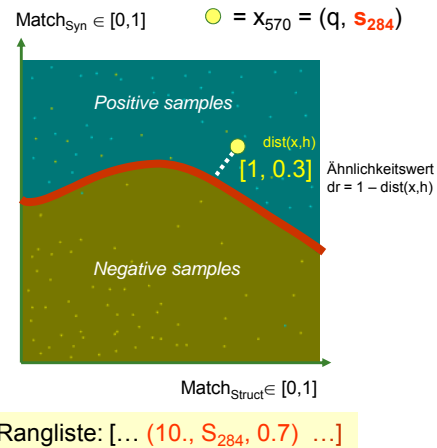
- Pro registrierten Dienst S: *Bestimme semantische Relevanz*  
 $(d(q,s), dr)$  für Abgleichsvektor (q, s)
- Ausgabe *relevanter* Dienste S für Q in *Rangfolge* r:  $[(r, S, dr)]_{(r,<)}$

## Beispiel

### Lernen über Trainingskollektion

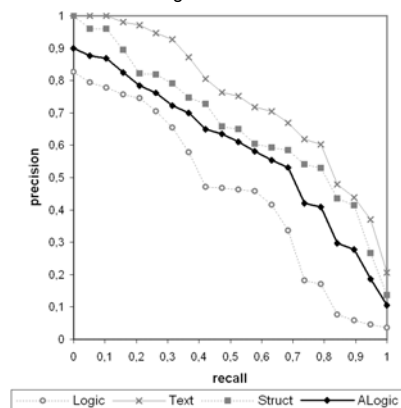


### Selektion für Dienstanfrage Q



## Experimentelle Evaluierung von iSeM

Macro-Averaged R/P over OWLS-TC4:



Download @:  
[projects.semwebcentral.org/projects/isem](http://projects.semwebcentral.org/projects/isem)

### Präzision der Filtervarianten

- Logikbasiert: 0.41
- Textähnlichkeit: 0.74
- Ontologisch strukturell: 0.65
- Logic  $\wedge$  Text  $\wedge$  Struct: 0.76
- **Adaptiv gewichtet kombiniert: 0.92**

Antwortzeit: 2.34 sec (MX2: 5.4 sec)



**Best Service  
Matchmaker  
in Precision**



## Semantische Komposition von Diensten



### ■ Automatisch

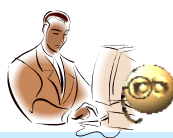
- Reduktion auf KI Planungsproblem
- Anwendung von KI Planverfahren

- Classical planning [Klusch 05, et al.]
- Model-checking [Traverso+ 01, et al.]
- HTN-planning [SHOP2: Sirin+ 02]

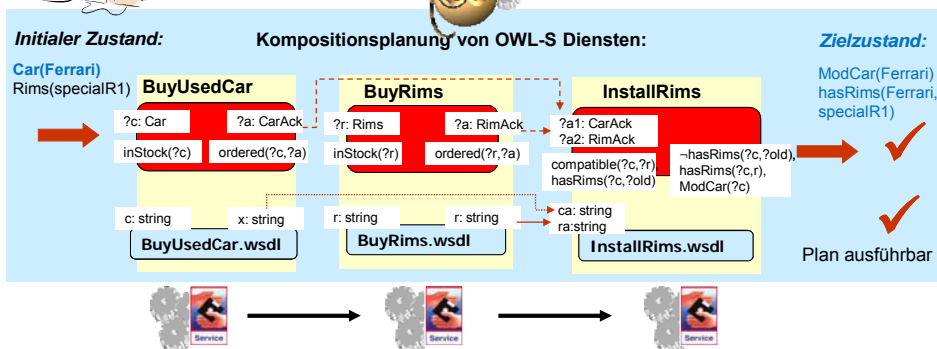
### ■ Interaktiv

- Vorwärts- oder Rückwärtskombination durch Benutzer mit Vermittleragenten [Sirin & Hendler 01, Henocque & Kleiner 07]

## Beispiel: Semantische Komposition mit OWLS-XPlan



„Kaufe einen gebrauchten Ferrari mit Felgen SpecialR1!“



@ [projects.semwebcentral.org/projects/owl-s-xplan](http://projects.semwebcentral.org/projects/owl-s-xplan). Top-20 (>5000 d/I)

## Anwendungen

- **Semantische Dienstvermittlung für Suchmaschinen mit OWLS-MX**

- Selektion von relevanten Raumfahrtinformationsdiensten

[NASA search engine SPASE, since 2007]



- Selektion von relevanten Werbungsdiensten zu Anfragen

[Pilotprojekt der U Zagreb mit Google, 2006]



- **Medizinische Dienstplanung mit OWLS-XPlan**

- Planung von medizinisch betreuten Repatriierungsdiensten

[Emergency Medical Assistance Group Ltd., Helsinki, Finland, since 2007]



- Planung von Proteinsequenzierungsdiensten

[U Rennes, Faculty of Medicine, Rennes, France, since 2008]



## Semantische Dienstbasierte Systeme

- **CASCOM System**

- Semantische Koordination von mobilen medizinischen Notfalldiensten
- European Commission



- **SoKNOS System**

- Semantische Koordination von Polizei-/Feuerwehr-/Ambulanzdiensten in Katastrophenszenarien
- Bundesministerium für Bildung und Forschung



- **THESEUS Texo System**

- Semantische Dienstvermittlung in Online-Handelsplattformen (B2B)
- Bundesministerium für Wirtschaft



## Intelligenter Einkaufsassistent

„Is *this pizza* good for my health? I am suffering from *lactose intolerance* !

Who offers the *cheapest party service* for my dinner? “



RDF/S Ontology



Attention! This pizza contains ingredients with lactose.  
Wanna try out fish, or visit store XYZ for lactose-free products ?  
Best party service for your special dinner is „CoolDining“.



Semantische Dienste:



Allergy



Party



Pricing

Produkte sind semantisch annotiert und per RFID und Barcode identifizierbar.

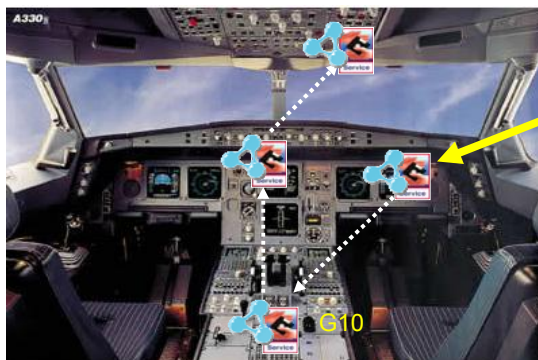
Suche nach relevanten semantischen Diensten für Anfrage zu Produkten.

@ DFKI Intelligent Retail Lab & BMBF project SemProM [www.semprom.org](http://www.semprom.org)

Klusch

69

## Intelligentes Cockpit Design



Show me: Which cockpit parts  
close to gear G10 are relevant for  
the standard landing procedure?  
How are switches A, B, C related?

Functional 3D simulation of aircraft handling in new cockpits @ design time.

Semantic annotation of gears, switches, buttons, panels (w/ services).

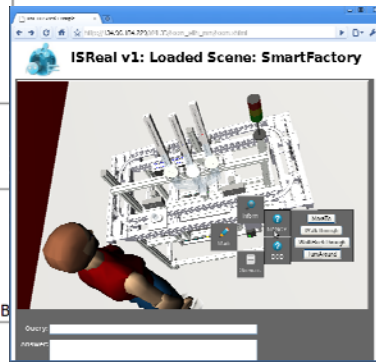
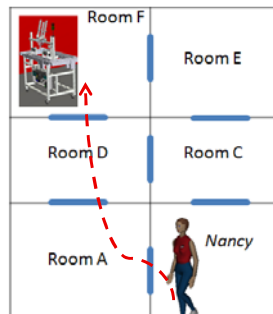
Intelligent agent roles: Expert (trainer) or novice (pilot).

13.12.2010, U Saarland

Klusch

70

## Semantik und Intelligente Avatare in 3D Welten



- *Semantisch annotierte XML3D* Szenen nativ simuliert im Browser
- Intelligente Agenten *verstehen* und *planen* mit der Semantik von Szenenobjekten: Intelligente Avatare

ISReal Project  
[www.dfki.de/~klusch/isreal](http://www.dfki.de/~klusch/isreal)



13.12.2010, U Saarland

Klusch

71

Thank you  
 for your attention !

Uni Saarland Campus, Bldg. D3.2, R +1.26  
 DFKI Research Group Multiagent Systems  
[klusch@dfki.de](mailto:klusch@dfki.de), [www.dfki.de/~klusch](http://www.dfki.de/~klusch)

