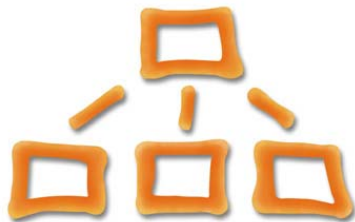




Peter Tomczyk

# Semantic Web und Ontologien



DBS

Informationsintegration und Web-Portale

WS 2003/04

- Web und Semantik
- Bausteine des „Semantic Web“
  - ▶ Begriffe
  - ▶ Aussagen
  - ▶ Ontologien
  - ▶ Regeln und Logik
- Werkzeuge für das Semantic Web
- Zusammenfassung

# Web und Semantik



## ■ Geschätzte Größe des World Wide Web:

### ▶ Lawrence & Glies, 1999:

“800 million pages (...) about 6 terabytes of text after removing HTML tags, comments, and extra whitespace.“

### ▶ NEC-RI & Inktomi, 2000:

„Größe des WEB übersteigt 1 Mrd indizierbare Webseiten.“

### ▶ 2001:

- Denic meldet den 4.000.000sten (.de) Eintrag.
- Spekulationen über ca. 550 Mio. Webseiten; täglich 7 Mio. neue

## ■ Google:

### ▶ „über 200 Mio. Suchvorgänge pro Tag“

### ▶ „Der Google-Suchindex enthält mehr als 4 Mrd.

Webdokumente.“

# ...und Semantik

- „The core meaning of the word ‚semantics‘ is: *meaning* itself.“
- Semantic Continuum:



Shared human  
consensus.

**Pump** : “a device for  
moving a gas or liquid  
from one place or  
container to another”

Text descriptions.



(pump has  
(superclasses (...))

Semantics hardwired;  
used at runtime.



Semantics processed  
and used at runtime.

**Implicit**

**Informal**  
(explicit)

**Formal**  
(for humans)

**Formal**  
(for machines)

[Uschold (2002), “Where are the Semantics in the Semantic Web?”]

# Die Geschichte von Lucy und Pete

“... (Pete’s) phone turned the sound down by sending a message to all other local devices that had a volume control...”

„Mom needs to see a specialist and then has to have a series of physical therapy session.“



“... (Lucy’s) agent (...) looked up several lists of providers, and checked for the ones in-plan for Mom’s prescribed insurance within a 20-mile radius of her home and with a rating of excellent (...) on trusted rating services...”

“...find a match between available appointment times and Pete’s and Lucy’s busy schedules...”

[Berners-Lee et al. (2001): “The Semantic Web”]

# Semantik: Das Problem des WWW

## ■ Was „sieht“ ein Rechner im Web?

**Handy „Siemens ME55“**

```
<html>
  <head><title>Siemens SL 55</title></head>
  <body>
    ...
    <a href=„http://www.siemens.de>Siemens</a>
    ...
    <td>1 €</td>
    ...
  </body>
</html>
```

<http://www.handy-shop.de/siemens-sl55.html>

**Hersteller „Siemens“**



**Relation  
„ist hergestellt von“**

<http://www.siemens.de>

## ■ XML ist nicht genug:

**Syntaktisch  
verschieden**

**Syntaktisch  
gleich**

Tags:

`<price>`

`<preis>`

`<preis>`

Bedeutung:

ohne Vertrag

ohne Vertrag

mit Vertrag

**Semantisch  
gleich**

**Semantisch  
verschieden**

# Prinzipielle Lösungsansätze

- 1) Wir lassen das Web, wie es ist, und bauen „intelligenterere“ Software:
  - ▶ Funktioniert z.T. in sehr begrenzten Bereichen, skaliert aber kaum bzgl. der Domäne
  - ▶ Beispiele: Preissuchmaschinen, klassische I<sup>3</sup>-Systeme
  
- 2) Wir ergänzen das Web um beschreibende Informationen, die für Maschinen „verständlich“ sind:
  - ▶ Beschreibende Informationen → Metadaten
  - ▶ Wie das geht, sehen wir gleich...

# Semantic Web: Die Vision



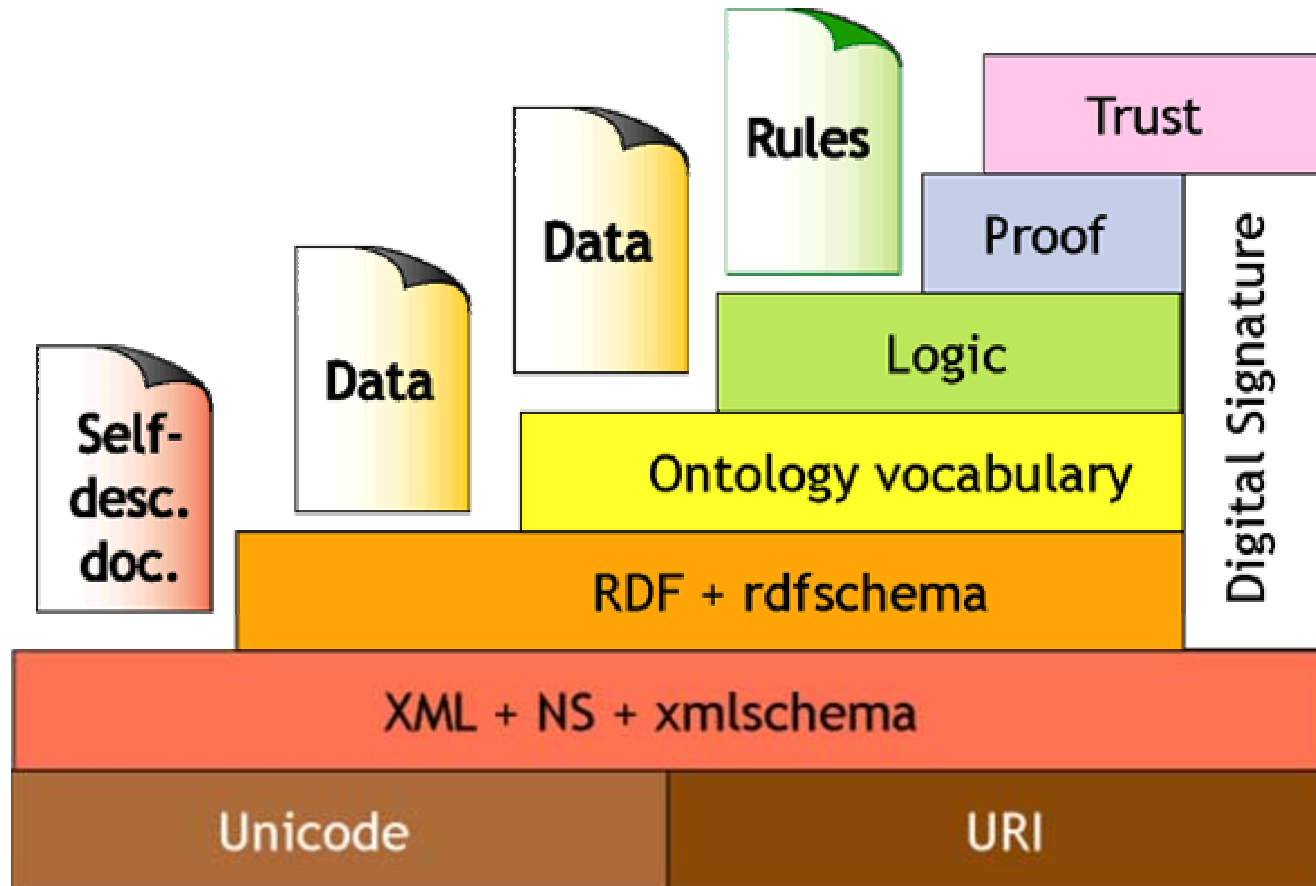
*“The Semantic Web is a vision: the idea of having data on the Web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications.”*

[W3C 2001]

*“The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.”*

[Berners-Lee et al 2001]

# Schichten des Semantic Web



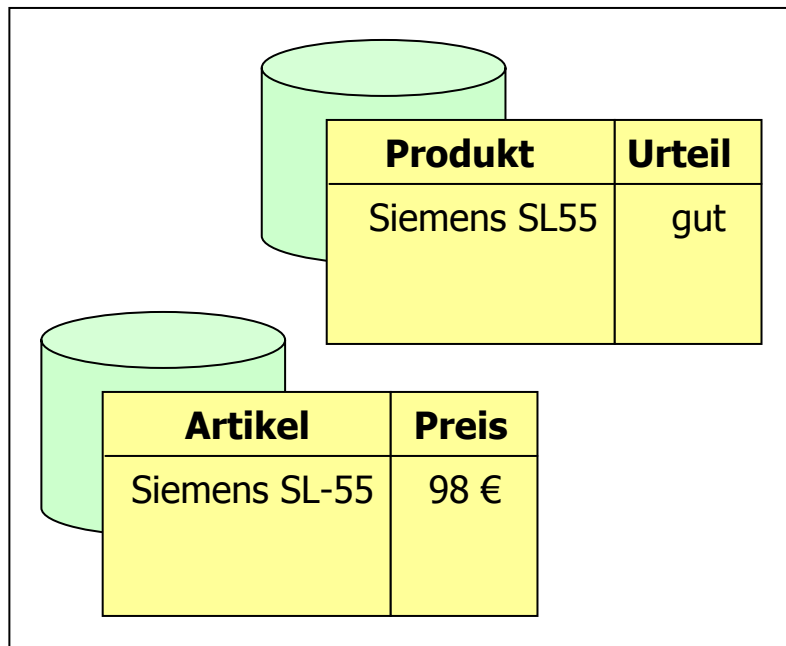
Quelle: Berners-Lee (1999)

**Begriffe**

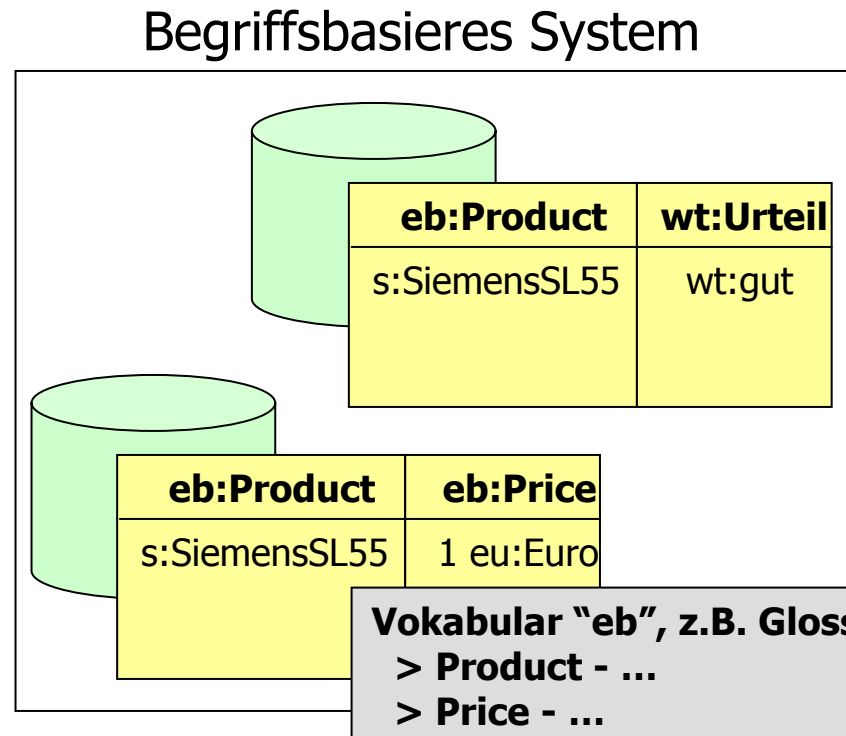


# Begriffe statt Werte

- Verwende eindeutige Begriffe statt (bedeutungsloser) Werte:



„Konventionelles“ System



Vokabular "eb", z.B. Glossar:  
> Product - ...  
> Price - ...

# Verbreitete Begriffsregelungen

- Die Idee des *kontrollierten Vokabulars* ist nicht neu:
  - ▶ ISO 3166 Länderkennzeichen
    - cx Christmas Island
    - cy Cyprus
    - cz Czech Republic
    - de Germany
    - dj Djibouti ...
  - ▶ ISO 639 Sprachkennzeichen
  - ▶ ISO 4217 Währungskennzeichen
  - ▶ Postleitzahlen
  - ▶ ...

# Resources und URIs

- Im Semantic Web spricht man von „Resources“:
  - ▶ beliebige physikalische oder abstrakte Objekte (Person, Buch, Web-Dokument ...)
  - ▶ identifiziert mittels Unified Resource Identifier (URI, RFC 2396)
- URIs im Semantic Web:
  - ▶ Wiederverwendung vorhandener URLs, E-Mail-Adressen etc.  
`http://www.fzi.de/dbs, mailto:ptomczyk@fzi.de`
  - ▶ URIs mit „Fragment Identifier“ (*Vokabular#Begriff*):  
`http://www.fzi.de/glossar#Abteilungsleiter`
- Probleme:
  - ▶ URIs können ihre Bedeutung ändern:

`http://www.wetter.de/wetter-heute.html`

**Aussagen**



# RDF: Datenmodell des Semantic Web

- „When looking at a possible formulation of a universal Web of semantic assertions, the principle of minimalist design requires that it be based on a common model of great generality. (...) The general model is the **Resource Description Framework**.” [Berners-Lee, 1998]
- „In RDF, a document makes assertions that particular things (people, Web pages or whatever) have properties (such as „is a sister of“, „is the author of“) with certain values (another person, another Web page).” [Berners-Lee et al. „The Semantic Web“]

## ■ Statement:

- ▶ „Peter Tomczyk arbeitet bei Prof. Lockemann.“

## ■ Struktur:

- ▶ Subjekt (Resource)

`http://www.fzi.de/home#ptomczyk`

- ▶ Prädikat (Property  $\subset$  Resource)

`http://www.uni-karlsruhe.de/home#arbeitetBei`

- ▶ Objekt (Resource | Literal)

`http://www.uni-karlsruhe.de/home#lockemann`

## ■ Graph:

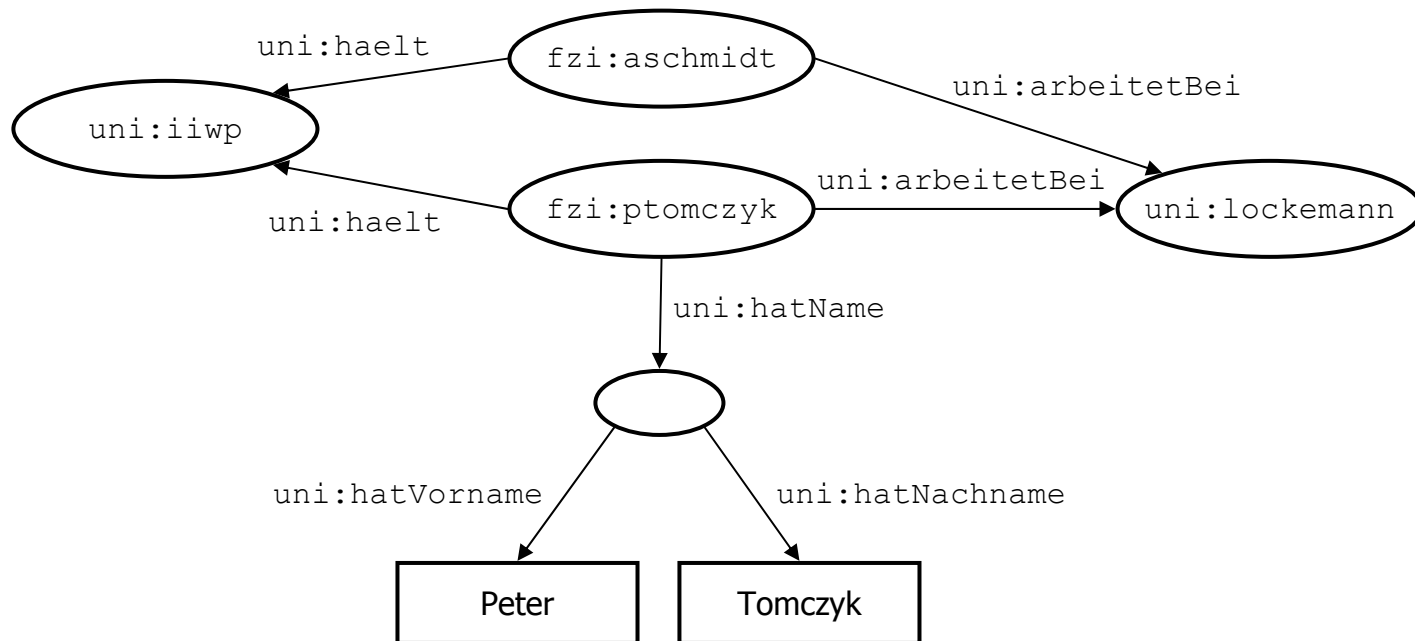
`http://www.uni-karlsruhe.de/home#arbeitetBei`

`http://www.fzi.de/home#ptomczyk`

`http://www.uni-karlsruhe.de/home#lockemann`

# RDF: Komplexe Aussagen

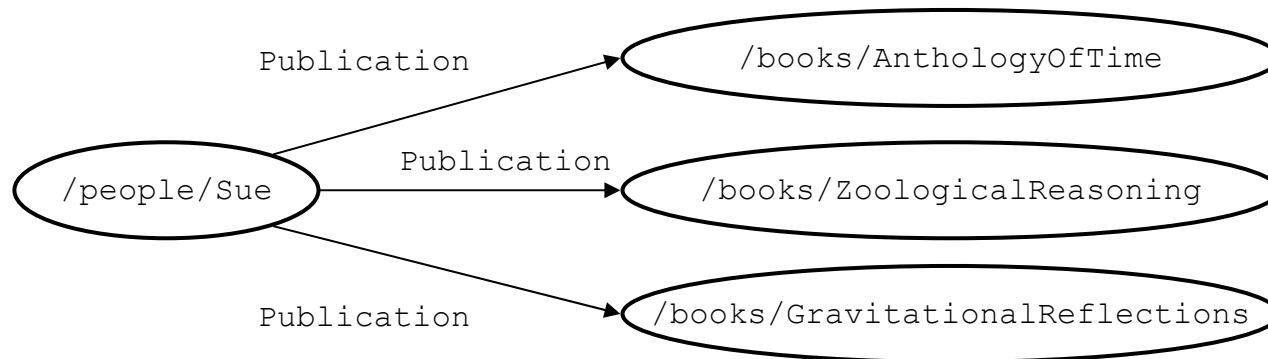
- Komplexe Aussagen entstehen durch Verwendung anonymer Ressourcen:



# RDF: Kollektionen (1)

- Mehrfaches Auftreten einer Eigenschaft desselben Typs stellt keine inhaltliche Verbindung zwischen den Werten her:

*Sue has written "Anthology of Time", "Zoological Reasoning", "Gravitational Reflections".*

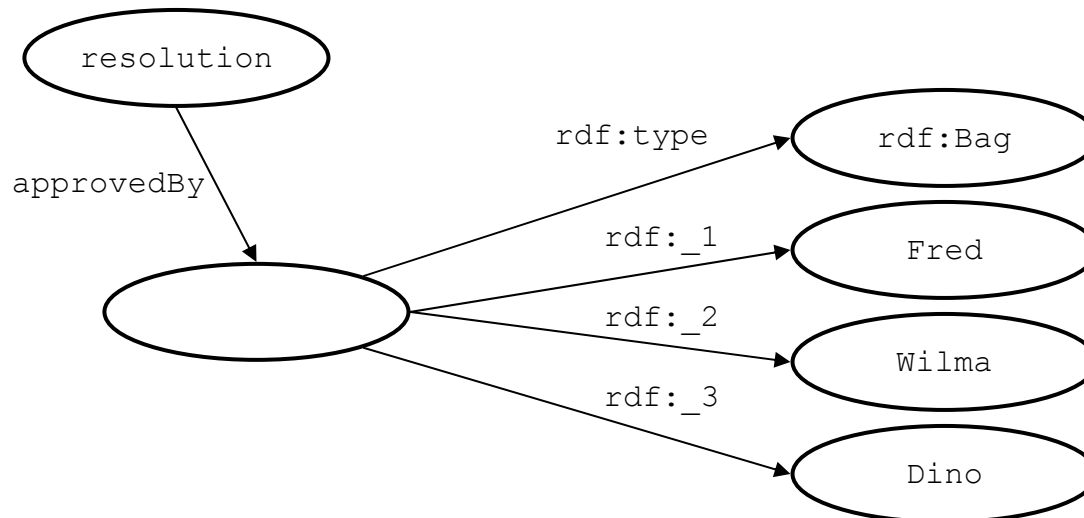


# RDF: Kollektionen (2)

## ■ Herstellung von Zusammenhängen durch Kollektionen:

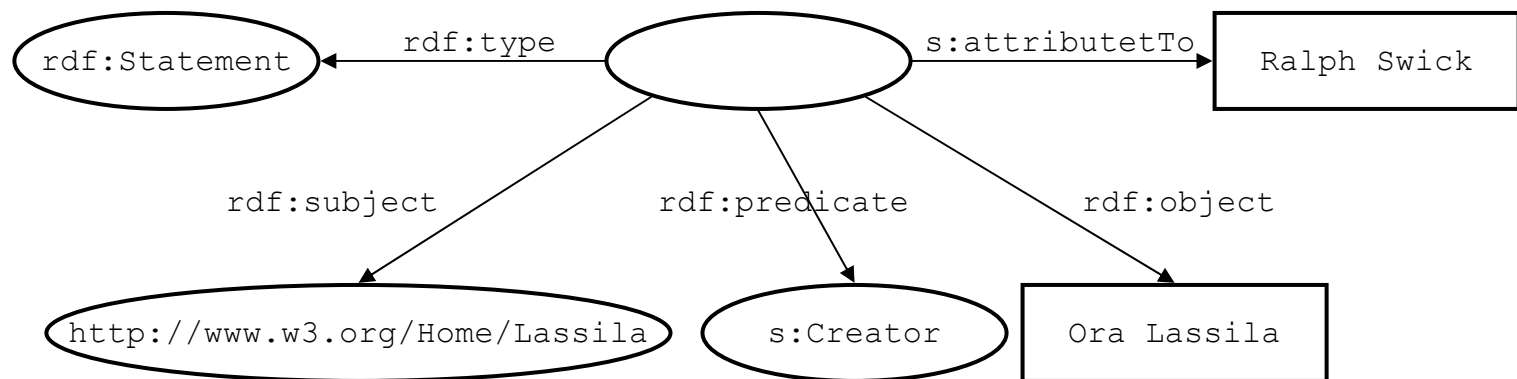
- ▶ `rdf:Seq` – Reihenfolge von Bedeutung
- ▶ `rdf:Bag` – Reihenfolge besitzt keinerlei Bedeutung
- ▶ `rdf:Alt` – Es handelt sich um mehrere Alternativen

*The committee of Fred, Wilma, and Dino approved the resolution.*



# RDF: Aussagen über Aussagen

- RDF erlaubt Aussagen über Aussagen (sog. Reifikation):  
*Ralph Swick says that Ora Lassila is the creator of the resource <http://www.w3.org/Home/Lassila>.*



# RDF: Formales Modell (1)

## ■ Grundlegende Definitionen:

- ▶ Resources
- ▶ Properties  $\subset$  Resources
- ▶ Literals
- ▶ Statements = Properties  $\times$  Resources  $\times$  { Resources  $\cup$  Literals }

## ■ Typisierung:

- ▶ `rdf:type`  $\in$  Properties
- ▶ {`rdf:type`, `sub`, `obj`}  $\in$  Statements  $\Rightarrow$  `obj`  $\in$  Resources

Quelle: <http://www.w3.org/TR/REC-rdf-syntax/>  
Hier nach: Wahlster (2000)

# RDF: Formales Modell (2)

## ■ Kollektionen:

- ▶ `{rdf:Bag, rdf:Seq, rdf:Alt}` ∈ Resources - Properties
- ▶ `Ord = (rdf:_1, rdf:_2, rdf:_3, ...)` ⊂ Properties

## ■ Reifikation (Aussagen über Aussagen):

- ▶ `rdf:Statement` ∈ Resources - Properties
- ▶ `{rdf:predicate, rdf:subject, rdf:object}` ∈ Properties
- ▶ **Reifikation eines Tripels** `{pred, sub, obj}` **von Statements ist ein Element** `r` ∈ Resources **und die Elemente** `s1, s2, s3, s4` ∈ Statements, **sodass:**
  - `s1: {rdf:predicate, r, pred}`
  - `s2: {rdf:subject, r, subj}`
  - `s3: {rdf:object, r, obj}`
  - `s4: {rdf:type, r, [rdf:Statement]}`

# RDF und XML: RDF Syntax

- XML ist eine mögliche Serialisierung eines RDF-Graphen:

```
<rdf:RDF xmlns:rdf="..." xmlns:rdfs="..." xmlns:uni="...">

  <rdf:Description rdf:ID="lockemann">
    <rdfs:label>Prof. Lockemann</rdfs:label>
    <uni:hatName rdf:parseType="Resource">
      <uni:hatVorname>Peter C.</uni:hatVorname>
      <uni:hatNachname>Lockemann</uni:hatNachname>
    </uni:hatName>
  </rdf:Description>

  <rdf:Description rdf:about="http://www.fzi.de/home#ptomczyk">
    <rdfs:label>Peter Tomczyk</rdfs:label>
    <uni:hatName rdf:parseType="Resource">
      <uni:hatVorname>Peter</uni:hatVorname>
      <uni:hatNachname>Tomczyk</uni:hatNachname>
    </uni:hatName>
    <uni:arbeitetBei rdf:resource="#lockemann"/>
    <uni:haelt rdf:resource="#iiwp"/>
  </rdf:Description>

</rdf:RDF>
```

# RDF und „herkömmliche“ Modelle

- Strikte Trennung von Aussagen und ihrem Gegenstand:
  - ▶ Jeder kann alles über alles sagen
  - ▶ Gegensatz zur Objektorientierung, in der ein Objekt alle seine Eigenschaften enthält
  - ▶ Duplikate und Widersprüche sind in RDF ganz „natürlich“
    - Web-gerechte Auffassung der Konsistenz
  - ▶ Frage: Wann vertraue ich einer Aussage?
    - „Trust“ ist die Spitze der RDF-Pyramide
- Keine “Closed World Assumption”
  - ▶ Lässt sich mit dem vorhandenen Wissen eine Aussage nicht begründen, so heißt es nicht, dass sie falsch ist
  - ▶ Gegensatz zu klassischen wissensbasierten Systemen

# Ontologien

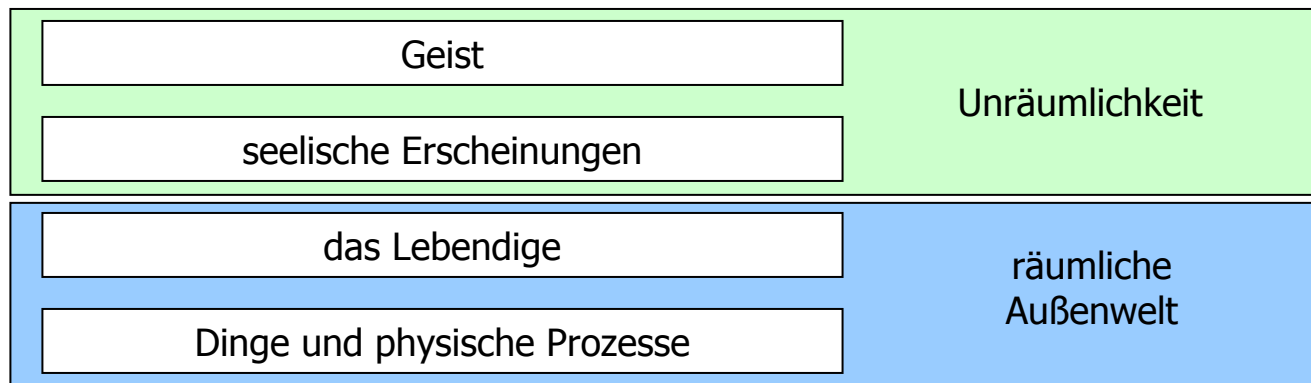


# Noch offene Fragen...

- Was sind die Konzepte einer Domäne?
  - ▶ Beispiel Universität: Lehrkräfte, Professoren, Assistenten, Vorlesungen,  $x$  ist Voraussetzung für  $y$ ,  $x$  arbeitet für  $y$ , ...
- Welche Aussagen sind in einer Domäne zulässig bzw. sinnvoll?
  - ▶ *Vorlesung\_1* ist Voraussetzung für *Vorlesung\_2*
  - ▶ *Assistent* arbeitet für *Professor*
- Welche weiteren Aussagen lassen sich aus vorhandenen Aussagen folgern?
  - ▶  $A$  arbeitet für  $P \Rightarrow P$  hat Mitarbeiter  $A$

# Ontologie in der Philosophie

- Die Lehre vom Seienden, seinen Kategorien und Eigenschaften
- Geht zurück auf Aristoteles
- wichtige Vertreter der Neuzeit:
  - Alfred North Whitehead (1861-1947)
  - Nikolai Hartmann (1882-1950):

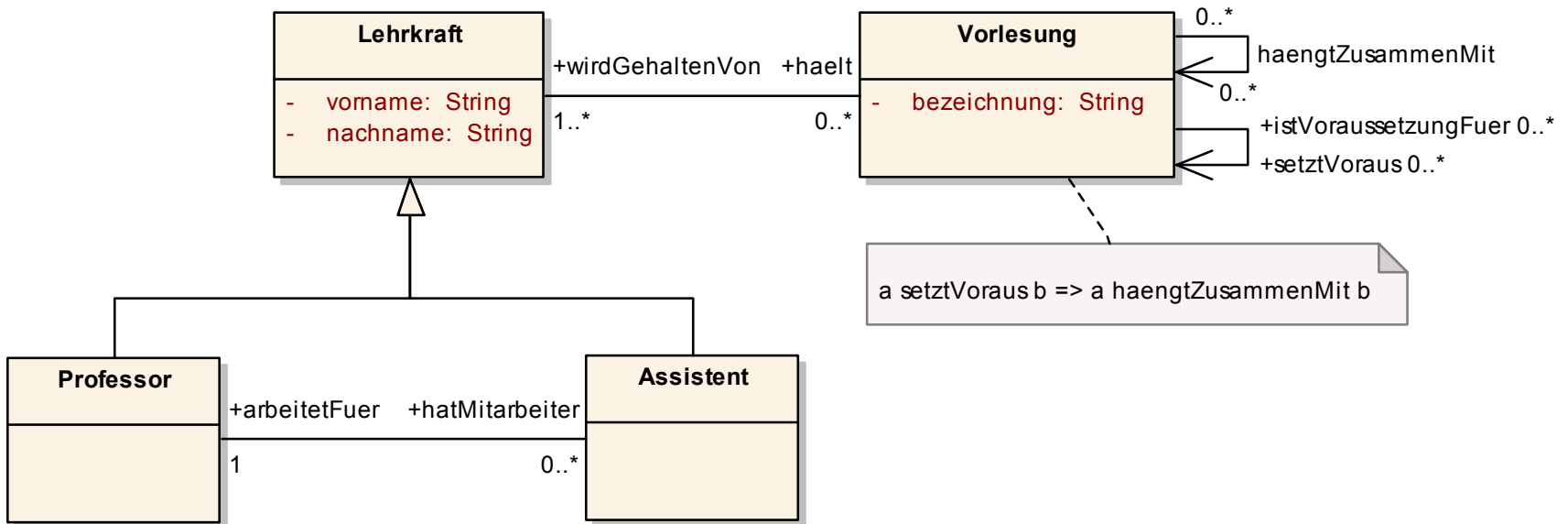


# Ontologie in der Informatik

- Zahlreiche unterschiedliche Definitionen:
    - ▶ „An Ontology is an explicit specification of a conceptualization.“ (Gruber, 1993)
    - ▶ „An ontology is a shared understanding of a domain of interests.“ (Uschold, Gruninger, 1996)
    - ▶ Ausführliche Diskussion des Begriffs: Guarino (1996)
  - wesentlich:
    - ▶ Ontologien sind Modelle von Realweltausschnitten
    - ▶ Ontologien bedienen sich einer formalen Sprache
    - ▶ Ontologien werden gemeinschaftlich genutzt
- ⇒ unverkennbare Verwandtschaft mit der konzeptuellen Modellierung in SWT (UML) und Datenbanken (ER)

# Beispiel: „Universität“

- Modellierung in UML (nach Uschold „explizit, informal“):



# RDF Schema: Modellierungskonstrukte

## ■ Kernklassen:

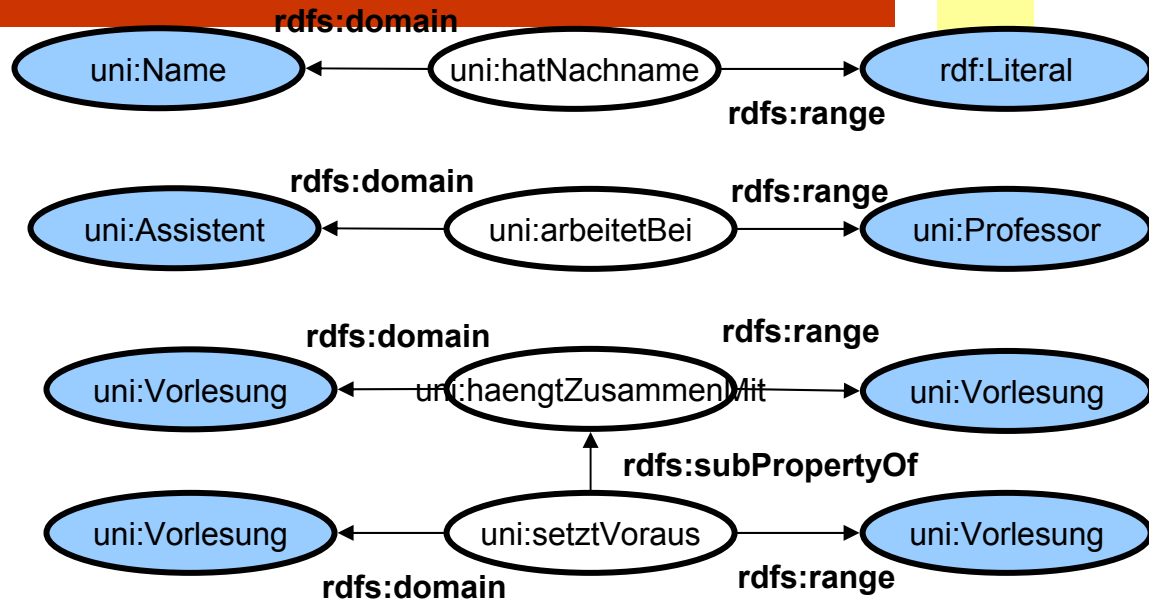
- ▶ `rdfs:Resource` – Klasse aller Resources
- ▶ `rdfs:Class` – Klasse aller Klassen (enthält sich selbst!)
- ▶ `rdfs:Literal` – Klasse aller möglichen Literale
- ▶ `rdf:Property` – Klasse aller Properties (aus RDF)

## ■ Kerneigenschaften:

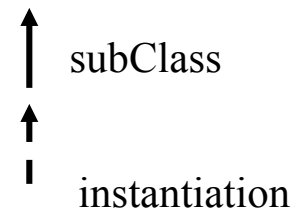
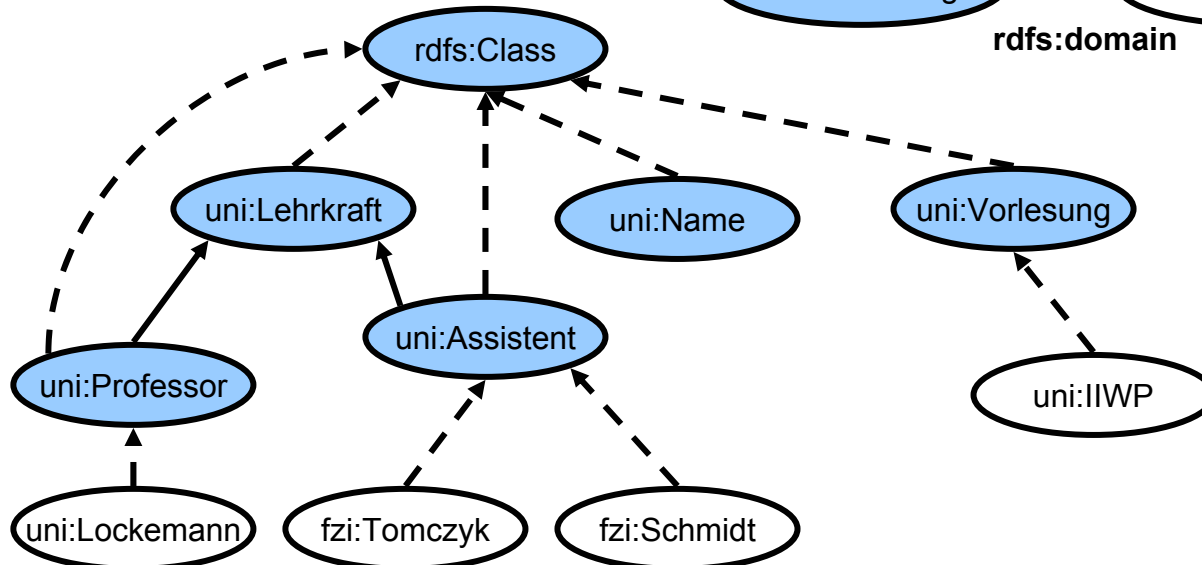
- ▶ `rdf:type` – „ist Instanz von“ (aus RDF)
- ▶ `rdfs:domain`, `rdfs:range` – Anwendbarkeit von Properties
- ▶ `rdfs:subClassOf`, `rdfs:subPropertyOf` – Spezialisierung
- ▶ `rdfs:label`, `rdfs:comment` – Informationen im Klartext

# Ontologie „Universität“ in RDF Schema

Declaration  
of properties

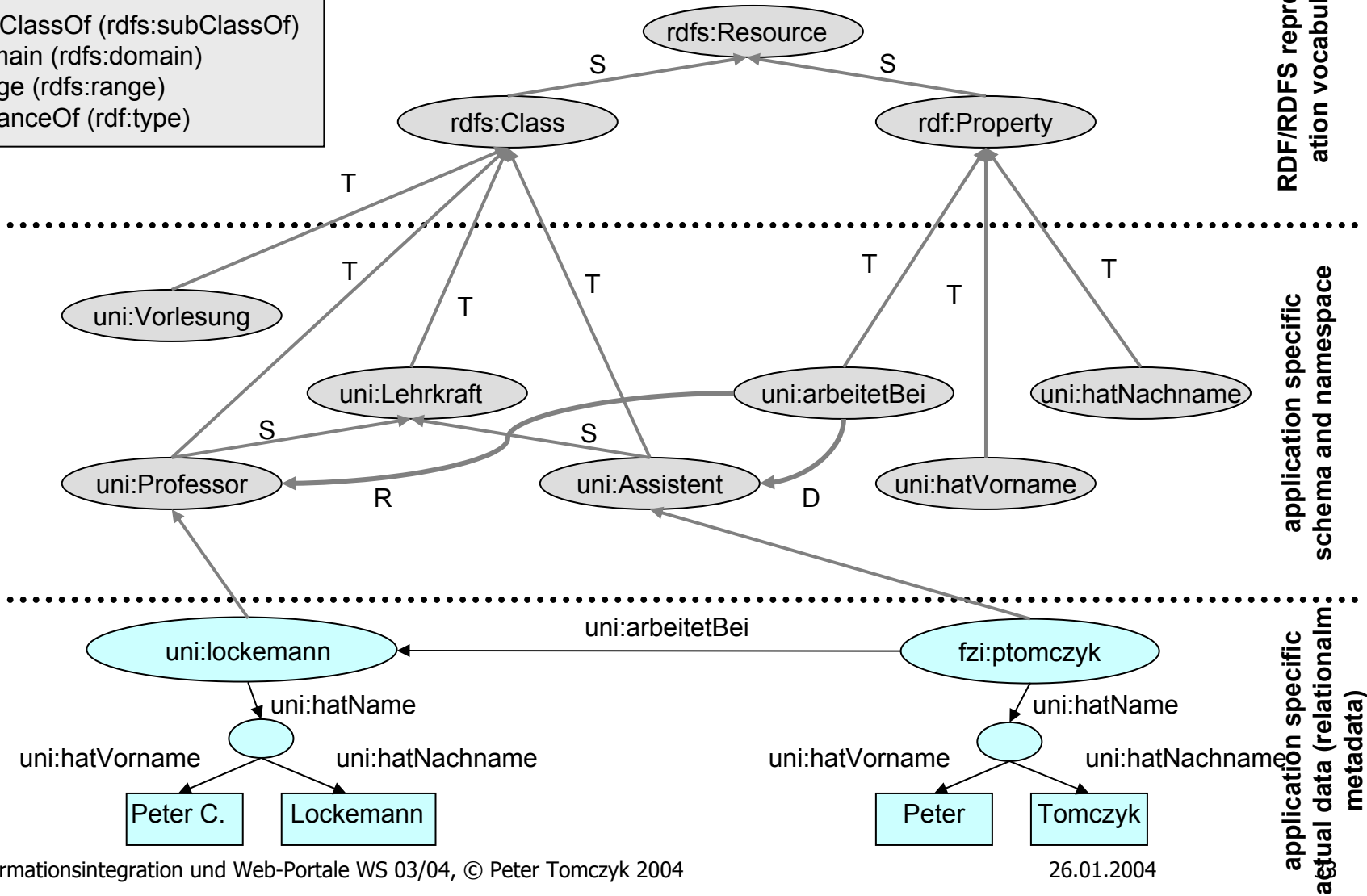


Declaration  
of classes



# RDF Schema: Schichtung

S subClassOf (rdfs:subClassOf)  
 R domain (rdfs:domain)  
 D range (rdfs:range)  
 T instanceOf (rdf:type)



# Axiome und logische Schlüsse (1)

## ■ rdfs:subClassOf und rdf:type

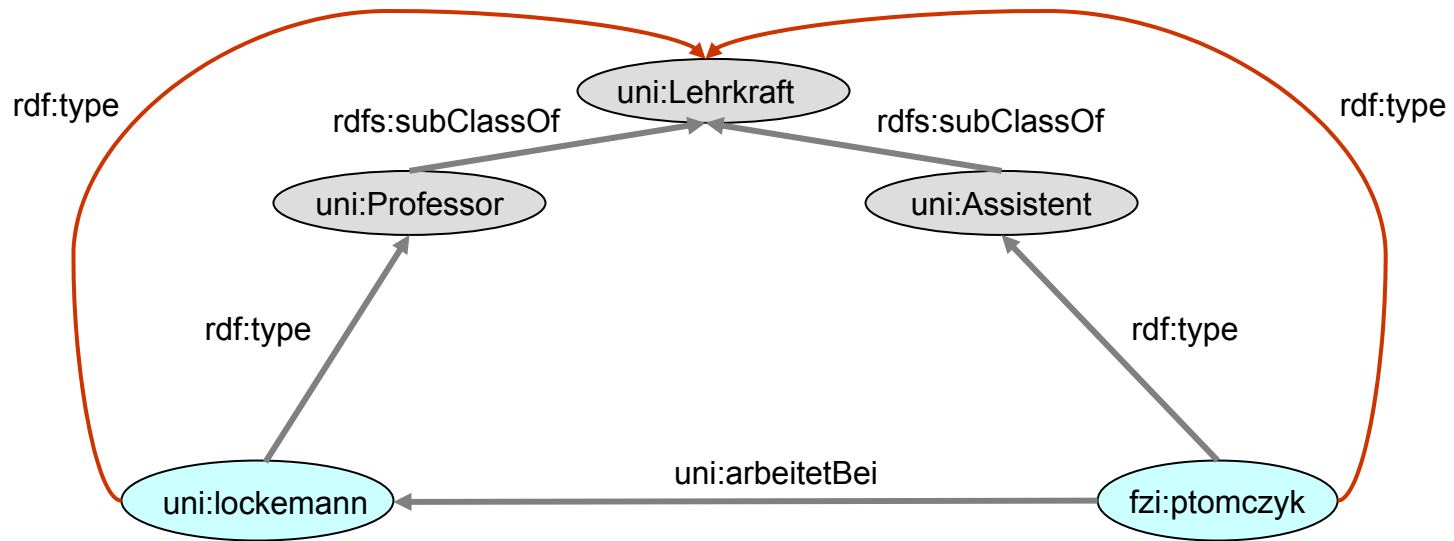
- ▶  $\forall i, c_1, c_2 \tau(i, \text{rdf:type}, c_1) \wedge \tau(c_1, \text{rdfs:subClassOf}, c_2) \Rightarrow \tau(i, \text{rdf:type}, c_2)$
- ▶  $\forall c_1, c_2, c_3 \tau(c_1, \text{rdfs:subClassOf}, c_2) \wedge \tau(c_2, \text{rdfs:subClassOf}, c_3) \Rightarrow \tau(c_1, \text{rdfs:subClassOf}, c_3)$

## ■ rdfs:subPropertyOf

- ▶  $\forall s, o, p_1, p_2 \tau(s, p_1, o) \wedge \tau(p_1, \text{rdfs:subPropertyOf}, p_2) \Rightarrow \tau(s, p_2, o)$
- ▶  $\forall p_1, p_2, p_3 \tau(p_1, \text{rdfs:subPropertyOf}, p_2) \wedge \tau(p_2, \text{rdfs:subPropertyOf}, p_3) \Rightarrow \tau(p_1, \text{rdfs:subPropertyOf}, p_3)$

Quelle: Champin (2000)

# Axiome und logische Schlüsse (2)



# XML-Serialisierung von „Universität“

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns"
         xmlns:rdfs=" http://www.w3.org/2000/01/rdf-schema"
         xmlns:uni="http://www.uni-karlsruhe.de/staff">
...
  <rdfs:Class rdf:ID="Lehrkraft">
  </rdfs:Class>

  <rdfs:Class rdf:ID="Professor">
    <rdfs:subClassOf rdf:resource="#Lehrkraft"/>
  </rdfs:Class>

  <rdfs:Class rdf:ID="Assistent">
    <rdfs:subClassOf rdf:resource="#Lehrkraft"/>
  </rdfs:Class>

  <rdfs:Class rdf:ID="Name">
  </rdfs:Class>

  <rdfs:Class rdf:ID="Vorlesung">
  </rdfs:Class>

...
```

# XML-Serialisierung von „Universität“

```
...
<rdf:Property rdf:ID="hatName">
  <rdfs:domain rdf:resource="#Lehrkraft"/>
  <rdfs:range rdf:resource="#Name"/>
</rdf:Property>

<rdf:Property rdf:ID="hatVorname">
  <rdfs:domain rdf:resource="#Name"/>
  <rdfs:range rdf:resource="http://www.w3.org/...rdfs...#Literal"/>
</rdf:Property>

<rdf:Property rdf:ID="arbeitetBei">
  <rdfs:domain rdf:resource="#Assistent"/>
  <rdfs:range rdf:resource="#Professor"/>
</rdf:Property>

<rdf:Property rdf:ID="haengtZusammenMit">
  <rdfs:domain rdf:resource="#Vorlesung"/>
  <rdfs:range rdf:resource="#Vorlesung"/>
</rdf:Property>

<rdf:Property rdf:ID="setztVoraus">
  <rdfs:domain rdf:resource="#Vorlesung"/>
  <rdfs:range rdf:resource="#Vorlesung"/>
  <rdfs:subPropertyOf rdf:resource="#haengtZusammenMit"/>
</rdf:Property>
...
```

# XML-Serialisierung von „Universität“

...

```
<rdf:Description rdf:ID="lockemann">  
  <rdfs:label>Prof. Lockemann</rdfs:label>  
  <uni:hatName rdf:parseType="Resource">  
    <uni:hatVorname>Peter C.</uni:hatVorname>  
    <uni:hatNachname>Lockemann</uni:hatNachname>  
  </uni:hatName>  
</rdf:Description>
```

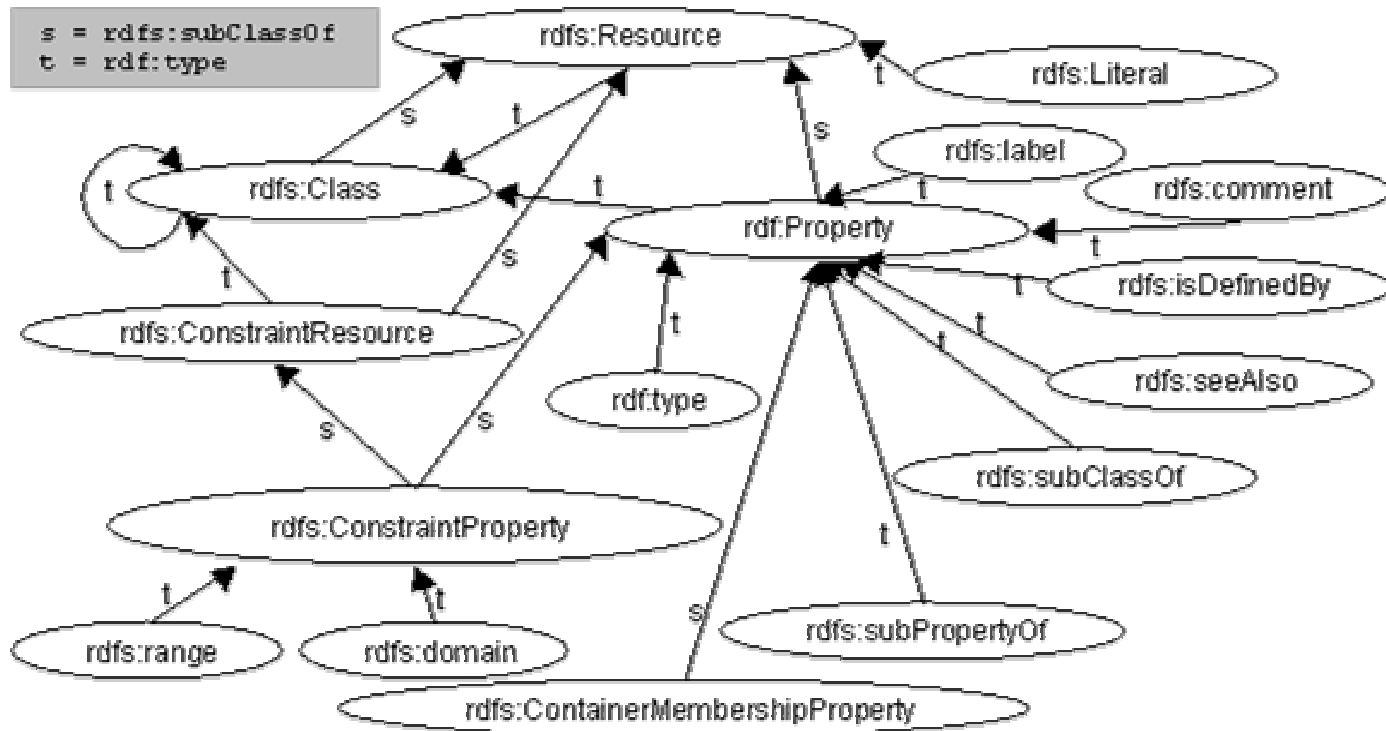
```
<rdf:Description rdf:about="http://www.fzi.de/home#ptomczyk">  
  <rdfs:label>Peter Tomczyk</rdfs:label>  
  <uni:hatName rdf:parseType="Resource">  
    <uni:hatVorname>Peter</uni:hatVorname>  
    <uni:hatNachname>Tomczyk</uni:hatNachname>  
  </uni:hatName>  
  <uni:arbeitetBei rdf:resource="#lockemann"/>  
  <uni:haelt rdf:resource="#iiwp"/>  
</rdf:Description>
```

...

```
</rdf:RDF>
```

# „Das Schema von RDF Schema“

- Das RDF Schema ist selbstbeschreibend (d.h. RDFS ist insbes. auch Schema für sich selbst):



# Kritik an RDF Schema

- Die wichtigsten Kritikpunkte:
  - ▶ Explizite Semantik nur für manche Primitive
  - ▶ Fragwürdiges Metamodell
  - ▶ Nur domain/range Einschränkungen von Properties
  - ▶ Keine Äquivalenz und Disjunktheit
  - ▶ Keine einfachen Axiommuster wie Reflexivität, Symmetrie und Transitivität
  - ▶ Keine allgemeinen Regeln
  - ▶ Keine Möglichkeit, Klassen automatisch zu berechnen (Subsumption)

Quelle: Maedche, IIWP WS2002/2003

# (Ein) formales Modell einer Ontologie

- Ontology  $O := (C, P, H_C, H_P)$ 
  - ▶  $C \subseteq E$  – set of concepts
  - ▶  $P \subseteq E$  – set of properties
  - ▶  $H_C$  – concept hierarchy
  - ▶  $H_P$  – property hierarchy
- Instance pool  $IP := (I, L, \text{instconc}, \text{instprop})$ 
  - ▶  $I \subseteq E$  – set of instances
  - ▶  $L$  is set of literal values,  $L \cap E = \emptyset$
  - ▶  $\text{instconc} : C \rightarrow 2^I$  – relates concepts with instances
  - ▶  $\text{instprop} : P \times I \rightarrow 2^{I \cup L}$  – relates instances through properties

Vereinfacht nach: Maedche, IIWP WS2002/2003

## ■ Ontology $O := (C, P, H_C, H_P)$

- ▶  $C = \{ \text{Lehrkraft, Professor, Assistent, Vorlesung} \}$
- ▶  $P = \{ \text{arbeitetBei, haelt, hatBezeichnung, setztVoraus, haengtZusammenMit} \}$
- ▶  $H_C = \{ (\text{Professor, Lehrkraft}), (\text{Assistent, Lehrkraft}) \}$
- ▶  $H_P = \{ (\text{setztVoraus, haengtZusammenMit}) \}$

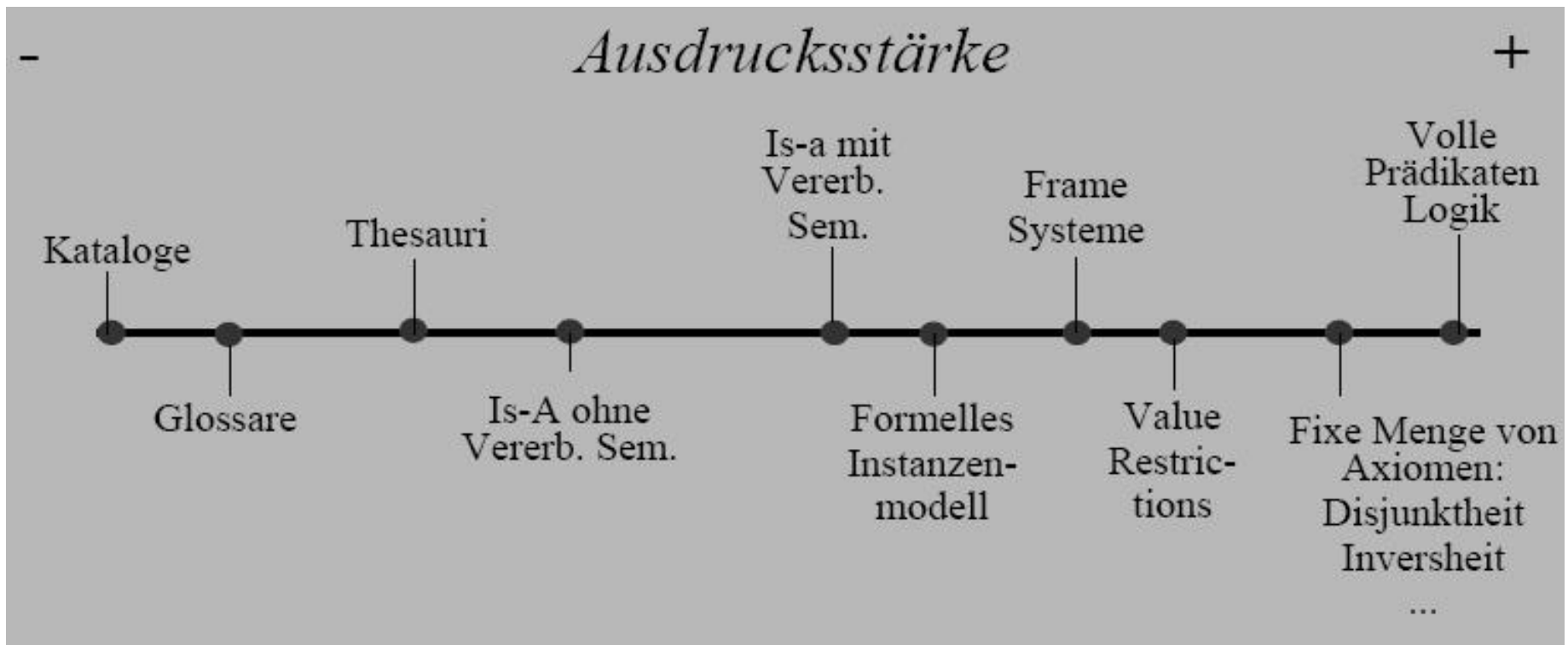
## ■ Instance pool $IP := (I, L, \text{instconc}, \text{instprop})$

- ▶  $I = \{ \text{Lockemann, Tomczyk, IIWP} \}$
- ▶  $L = \{ \text{"Informationsintegration und Web-Portale"} \}$
- ▶  $\text{instconc} : \{ (\text{Professor} \rightarrow \{ \text{Lockemann} \}), (\text{Assistent} \rightarrow \{ \text{Tomczyk} \}), (\text{Vorlesung} \rightarrow \{ \text{IIWP} \}) \}$
- ▶  $\text{instprop} : \{ ((\text{arbeitetBei, Tomczyk}) \rightarrow \{ \text{Lockemann} \}), ((\text{hatBezeichnung, IIWP}) \rightarrow \{ \text{"Informationsintegration und..."} \}), ((\text{haelt, Tomczyk}) \rightarrow \{ \text{IIWP} \}) \dots \}$

# Regeln und Logik



# Mächtigkeit von Ontologiesprachen



Quelle: Volz (2001)

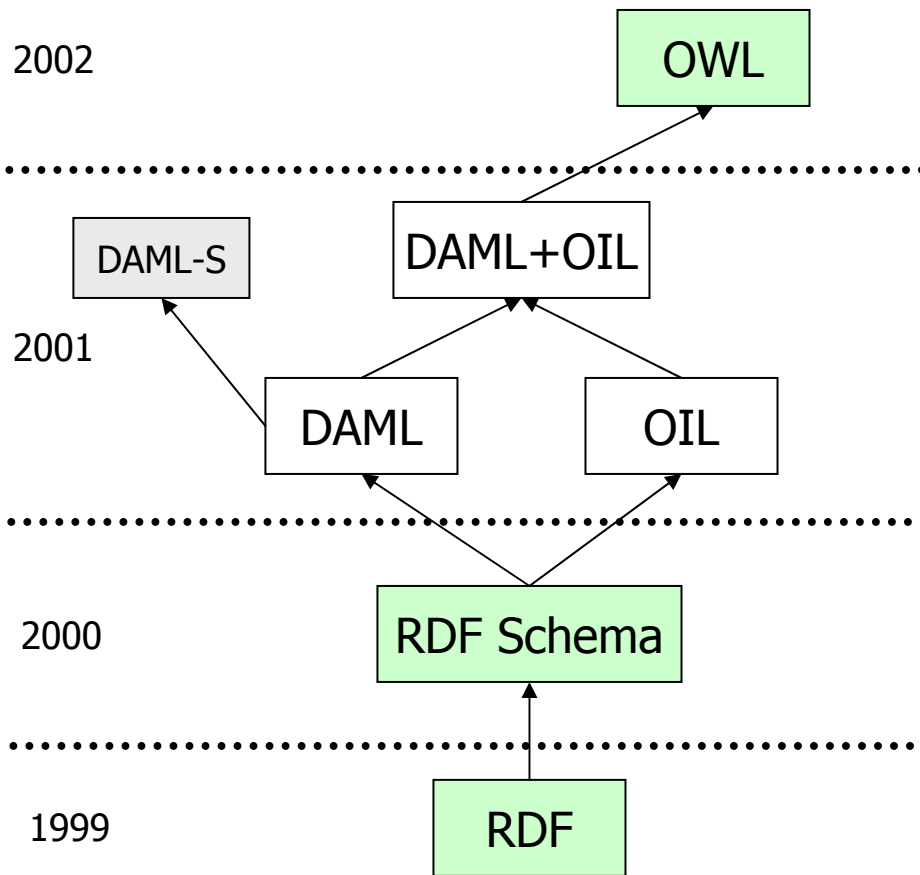
# Ontologiesprachen vor RDF(S)

	Ontol <sup>5</sup>	OKBC	OCML	LOOM	FLogic	XOL	SHOE	RDF(S)	OIL
Concepts	+	+	+	+	+	+	+	+	+
Relations	+	+/-	+	+	+/-	-	+	+	+
Functions	+	+/-	+	+	+/-	-	-	-	+
Procedures	+	+	+	+	-	-	-	-	-
Instances	+	+	+	+	+	+	+	+	-
Axioms	+	+/-	+	+	+	-	-	-	+
Production rules	-	-	+	+	-	-	-	-	+/-

Table 1. Definition of the main elements of domain knowledge.

Quelle: Corcho, Gómez-Pérez (2000)

# Ontologiesprachen für das Web



## ■ Erweiterungen:

- ▶ transitive, reflexive und symmetrische Properties
- ▶ disjunkte Klassen
- ▶ Kardinalitäten
- ▶ Deskriptive Definition von Klassen
- ▶ Datentypen (XML Schema)

## ■ Starker Fokus auf logische Entscheidbarkeit der Sprache

# (Ein) erweitertes formales Modell

- Ontology  $O := (C, P, S, T, INV, H_C, H_P, \text{domain}, \text{range}, \text{mincard}, \text{maxcard})$ 
  - ▶  $C \subseteq E$  – set of concepts
  - ▶  $P \subseteq E$  – set of properties
  - ▶  $S \subseteq P$  – set of symmetric properties
  - ▶  $T \subseteq P$  – set of transitive properties
  - ▶  $INV \subseteq P \times P$  – symmetric relation of inverse properties
  - ▶  $H_C$  – concept hierarchy
  - ▶  $H_P$  – property hierarchy
  - ▶  $\text{domain} : P \rightarrow 2^C \setminus \{ \emptyset \}$  – set of domain concepts
  - ▶  $\text{range} : P \rightarrow (2^C \setminus \{ \emptyset \}) \cup \{ L \}$  – set of domain concepts
  - ▶  $\text{mincard} : C \times P \rightarrow N_0$  – minimum cardinality
  - ▶  $\text{maxcard} : C \times P \rightarrow N_0$  – minimum cardinality

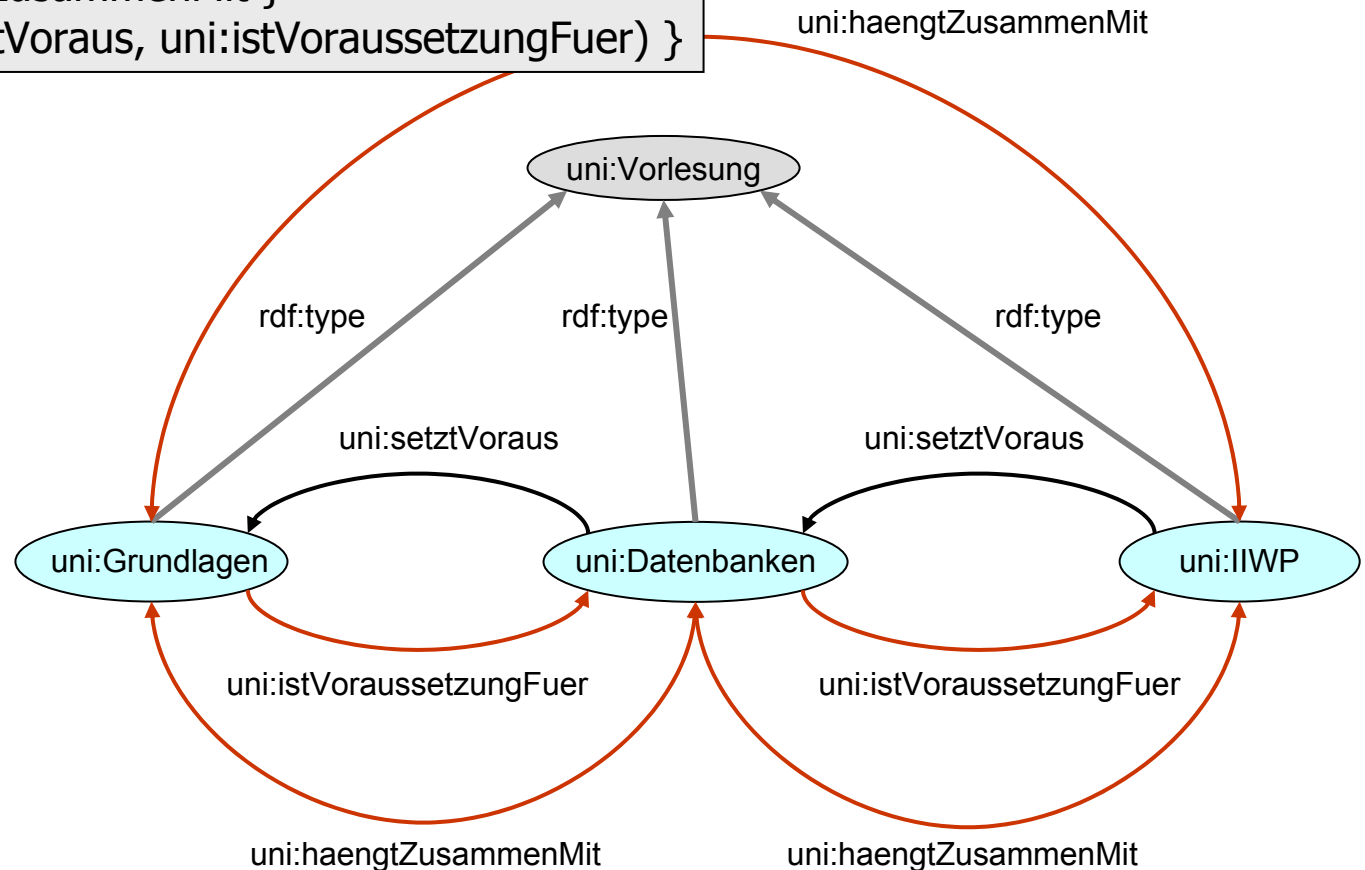
# Erweiterte logische Schlüsse

$H_p = \{ (uni:setztVoraus, uni:haengtZusammenMit) \}$

$S = \{ uni:haengtZusammenMit \}$

$T = \{ uni:haengtZusammenMit \}$

$INV = \{ (uni:setztVoraus, uni:istVoraussetzungFuer) \}$



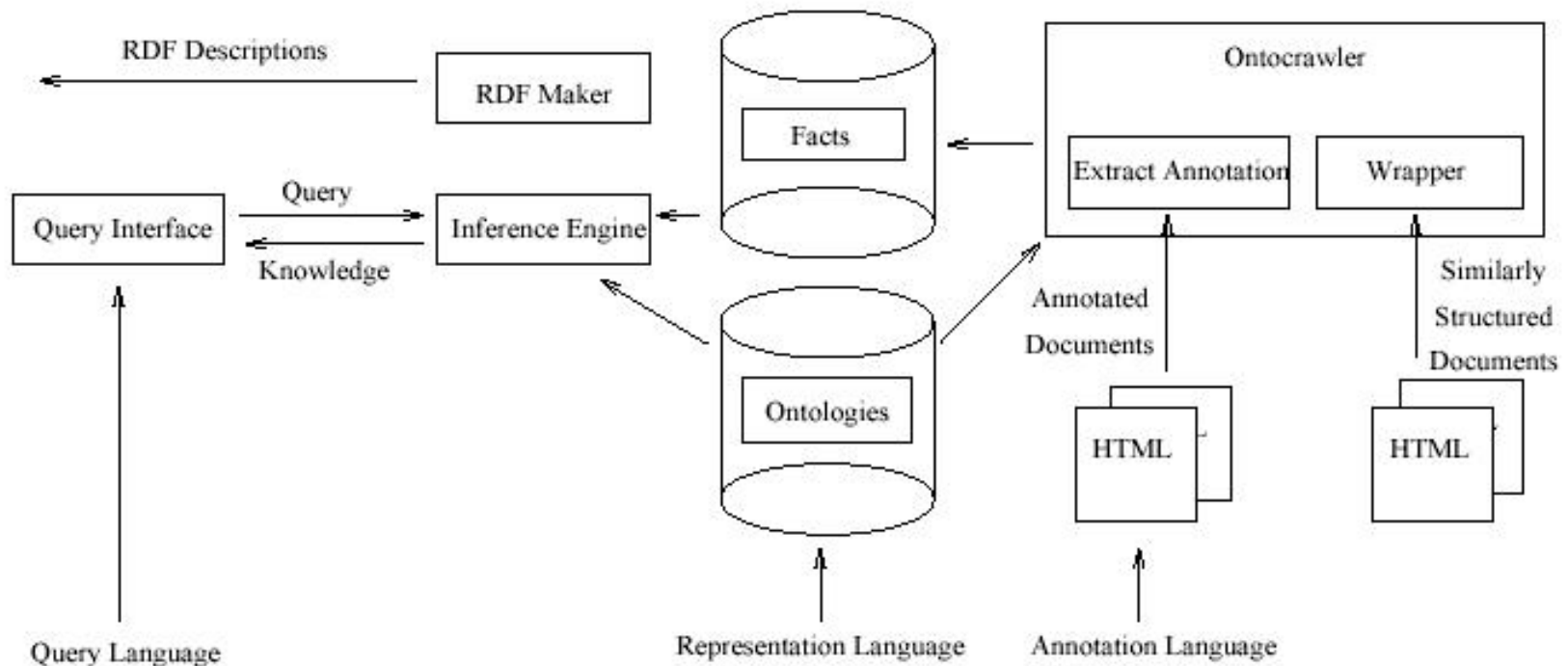
# Werkzeuge für das Semantic Web



# Onto\* und RDF\*: Die Werkzeuge

- OntoWeb (2001), „A survey on ontology tools“:
  - ▶ Ontology building tools
    - Protégé, OntoEdit, OIModeler
  - ▶ Ontology merge and integration tools
  - ▶ Ontology evaluation tools
  - ▶ Ontology based annotation tools
    - OntoMat, SHOE Knowledge Annotator
  - ▶ Ontology storage and querying
    - RDFPath, Triple, SquishQL, RQL, RDFQL, ...
    - Sesame, Jena, KAON

# Ontobroker: Der „Klassiker“



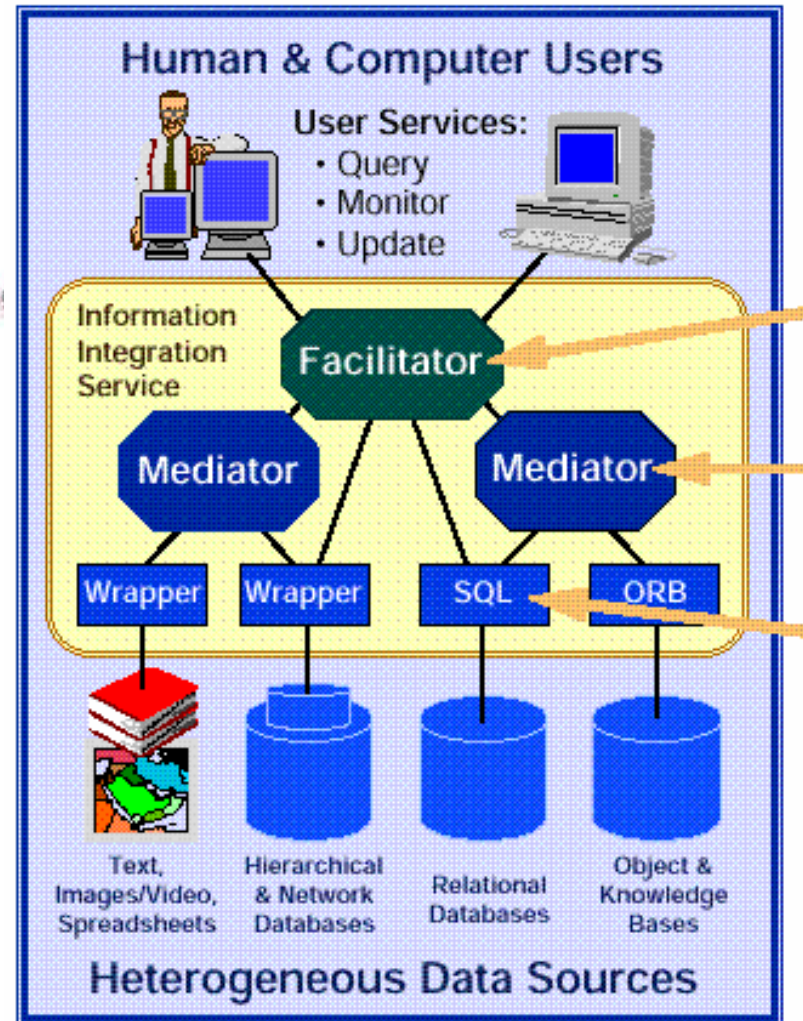
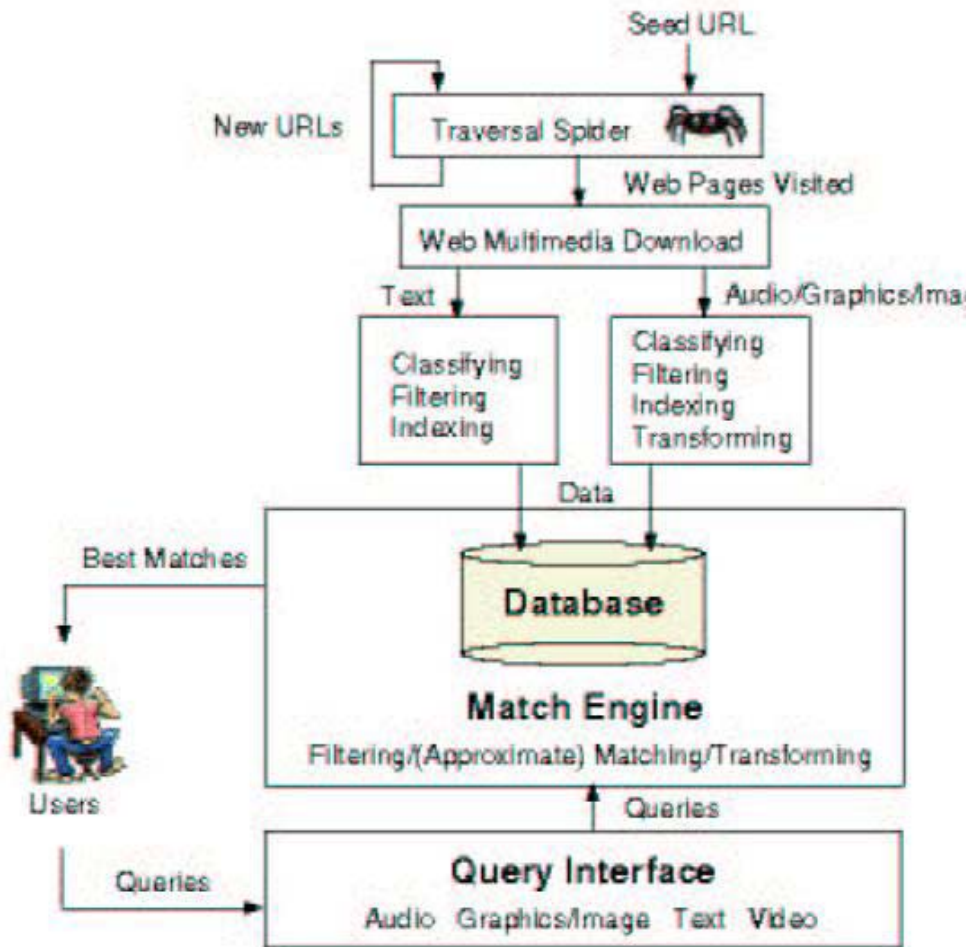
Quelle: Decker (1999)

# Annotation von HTML-Dokumenten

```
<html>
  <head>
    <title>FZI: Peter Tomczyk</title>
    <!--
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns"
            xmlns:uni="http://www.uni-karlsruhe.de/home">
      ...
      <uni:Assistent rdf:about="http://www.fzi.de/home#ptomczyk">
        <rdfs:label>Peter Tomczyk</rdfs:label>
        <uni:hatName rdf:parseType="Resource">
          <uni:hatVorname>Peter</uni:hatVorname>
          <uni:hatNachname>Tomczyk</uni:hatNachname>
        </uni:hatName>
        <uni:arbeitetFuer
          rdf:resource="http://www.uni-ka.../home#lockemann"/>
        </uni:Assistent>
      </rdf:RDF>
    -->
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

```
<html>
  <head>
    <a onto="page:Assistent"></a>
  </head>
  <body>
    ...
    <a onto=page [vorname=body]>Peter</a>
    <a onto=page [nachname=body]>Tomczyk</a>
  </body>
</html>
```

# Vergleich der Architekturen



[Hu, Chen (2001): "An Overview on World Wide Web Search Technologies"]

[Quelle unbekannt]

# Zusammenfassung



- „Semantic Web“ ist die Vision eines erweiterten Web, in dem die Inhalte für Maschinen verständlich sind.
- Bausteine:
  - ▶ eindeutige Begriffe statt bedeutungsloser Werte (URI)
  - ▶ einfaches und flexibles Datenmodell (RDF)
  - ▶ Ontologien als explizite und formale Domänenmodelle (RDFS, OWL)
  - ▶ Möglichkeit logischer Schlussfolgerungen (Transitivität, Reflexivität, Symmetrie)
  - ▶ Mechanismen zum Nachvollziehen von Schlüssen und Schließen unter eingeschränkten Vertrauen

## ■ Herausforderungen:

- ▶ Größe der Ontologien
  - semi-automatische Erstellung von Ontologien
  - Wiederverwendbarkeit (generelle Gestaltungsrichtlinien und –muster, Anpassungsfähigkeit, Mapping)
  - Veröffentlichen/Finden einmal erstellter Ontologien
  - Prüfung der Vergleichbarkeit und Korrektheit von Ontologien
- ▶ Dynamik der Ontologien
  - Evolutionsstrategien, Versionsverwaltung
- ▶ Anbindung nicht annotierter Quellen
- ▶ Subjektivität und gegenseitiges Misstrauen der Akteure

- D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster (eds.): Semantic Web Technology, to appear in: MIT Press, Boston, 2002.
- J. Hjelm: Creating the Semantic Web with RDF: Professional Developer's Guide, Wiley, 2001.
- T. Berners-Lee, J. Hendler and O. Lassila, The Semantic Web, Scientific American, 2001. [online: <http://www.sciam.com/2001/0501issue/0501berners-lee.html>]
- J. Heflin and J. Hendler: A Portrait of the Semantic Web in Action. IEEE Intelligent Systems, 16(2):54-59, 2001.

## ■ URLs:

- ▶ <http://www.w3.org/2001/sw/>
- ▶ <http://www.w3.org/RDF/>
- ▶ <http://www.semanticweb.org>
- ▶ <http://www.ilrt.bris.ac.uk/discovery/rdf/resources/>
- ▶ Semantic Web Roadmap:  
<http://www.w3.org/DesignIssues/Semantic.html>
- ▶ Software:
  - <http://kaon.semanticweb.org/>
  - <http://protege.semanticweb.org>
  - <http://www.daml.org/tools/>