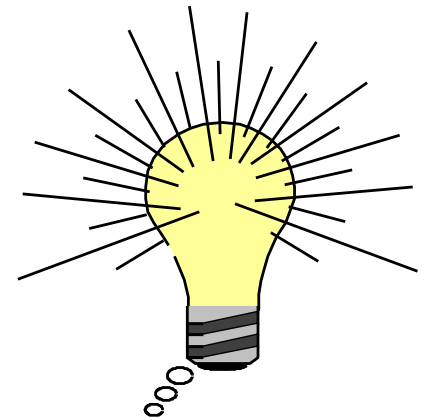


# Semantic Web

# Auf dem Weg zum Semantic Web

- Qualitative Verbesserung
  - ausgefeilte Such- und Bewertungstechniken
  - Häufigkeit, Position, Attribute des Suchbegriffs
  - Anzahl der Verweise auf eine Seite
- Beschreibung der Semantik von Seiteninhalten
  - Einsatz von Wissensrepräsentationstechniken
  - z.B. Konzeptgraphen oder Semantic-Web-Techniken (später)



# Ein besseres Web ...

- Das WWW ist ein Erfolg, gemessen an
  - der Anzahl Benutzer und
  - der Anzahl verfügbarer Webseiten
- Der Erfolg des WWW basiert auf der Einfachheit, jedoch
  - die Einfachheit hindert auch die Weiterentwicklung!

# Semantische Zusammenhänge



Wie entwickelte sich der  
Aktienkurs der Deutschen Bank  
in den letzten 3 Jahren?

Wie findet man diese Information?

# Googlen?

Nah dran,  
aber nicht  
das Gesuchte

Bezug zwischen  
den Begriffen?

The screenshot shows a Mozilla Firefox browser window with the title "Google-Suche: Deutsche Bank Aktienkurs 'letzten 3 Jahre' - Mozilla Firefox". The address bar contains the URL "http://www.google.de/search?hl=de&q=Deutsche+Bank+Aktienkurs+%2...". The search bar contains the text "Deutsche Bank Aktienkurs 'letzten 3 Jahre'". The search results are displayed under the heading "Web" and show "Ergebnisse 1 - 10 von ungefähr 58 für Deutsche Bank Aktienkurs 'letzten 3 Jahre'. (0,58 Sekunden)".

The search results include:

- [Die Frauenfinanzseite: Home](#)  
... Auch über die **letzten 3 Jahre** gehört er bei relativ moderaten ... Bildes richtet sich nach dem aktuellen **Aktien-Kurs** der Deutschen **Bank** (WKN:514000). ...  
[www.investorin.de/?section=Home&id=10153174 - 31k](#) - [Im Cache](#) - [Ähnliche Seiten](#)
- [Diskussionsforum von finanzen.net | Aktien | Börse | Wirtschaft ...](#)  
... Kleiner Tipp: Wenn man die Bilanzen der **letzten 3 Jahre** auseinander nimmt bekommt man es raus! ... 197. stand nicht die **Deutsche Bank** vor Gericht haha ! ...  
[finanzen.ariva.de/board/210396/ thread.m?a=&jump=1794575& - 114k](#) - [Im Cache](#) - [Ähnliche Seiten](#)
- [private Versicherungen, chat gratis test kostenlos preisvergleiche](#)  
... Berechnung: Monatsrendite der **letzten 3 Jahre** im Vergleich zum Top Quartil ... Kursgewinne ( bzw. minus der Kursverluste), bezogen auf den **Aktienkurs**. ...  
[www.aachenconsulting.de/t\\_gif/r\\_begriffe.htm - 134k](#) - [Im Cache](#) - [Ähnliche Seiten](#)
- [Aktienboard.com - Findet SAP den Weg aus der Krise?](#)  
... langsam begriffen hat, das man durch ständige Medienpräsenz den **Aktienkurs** pflegen kann. ... Die **Deutsche Bank** wird wegen der deutschen Steuerreform die ...  
[www.aktienboard.com/vb/archive/index.php/t-39855.html - 102k](#) - [Im Cache](#) - [Ähnliche Seiten](#)
- [\[PDF\] Bärfocus](#)  
Dateiformat: PDF/Adobe Acrobat - [HTML-Version](#)  
... issionen der **letzten 3 Jahre**) und eine 6 Mrd.- ... **Deutsche Bank**. 06.02.2004. Verizon. 13.02.2004. HBOS. 25.02.2004. Bulgari. 16.03.2004. Lafarge ...  
[www.juliusbaer.ch/runappl.cfm/Baerfocus%202004?wm=a\(135\)&dp=a\(d\)l\(1480\)&ext=.\\* - Ähnliche Seiten](#)
- [\[PDF\] Bärfocus](#)  
Dateiformat: PDF/Adobe Acrobat - [HTML-Version](#)  
... Resultat: Über die **letzten 3 Jahre** hat der Inve- ... **Deutsche Bank**. 01.07.2004. Richemont. 01.07.2004. Alcan. 01.07.2004. Siemens. 01.07.2004. Motorola ...  
[www.juliusbaer.ch/?wm=c\(565\)sp\(-1%2C75%2C417\) - Ähnliche Seiten](#)
- [|| Broker-Test.de || Interviews](#)  
... comdirect, - Cortal Consors, - CTS Brokerage, - DAB bank, - daytradeaustria.com ... Noch

On the right side of the search results, there are several "Anzeigen" (Ads) for "Deutsche Bank Konten", "wallstreet-online.de", "Aktienkurs", and "Schnelle Aktienkursuche".

The browser's status bar at the bottom shows "Fertig" and "Adblock".

# Suchprobleme im herkömmlichen WWW

- **Synonym:** bedeutungsähnliches, -gleiches Wort
  - Computer, Rechner, PC
- **Homonym:** das Wort, das ebenso wie ein anderes geschrieben und gesprochen wird, aber verschiedene Bedeutung hat. (Das Fremdwörterbuch, DUDEN)
  - Bank, Zug, springen
- **Begriff:** Umfang und Inhalt einer Vorstellung (Das Herkunftswörterbuch, DUDEN)
- Bislang eine Suche nach Wörtern, nicht nach Begriffen

# Das „Semantische Web“

- Bislang sind Informationen in HTML maschinell praktisch nicht auffindbar
- Integrieren von Bedeutungen in Ressourcen (Metadaten)
  - XML als geeignete Repräsentation
  - Ontologien: Aussagen über das Seiende (was existiert)
    - in der Informatik: eine explizite und formale Spezifikation einer Konzeptionalisierung
    - Konzepte, Relationen (Beziehungen)
- Nutzen aus der Künstlichen Intelligenz
  - dort Forschung an Ontologien, Wissensrepräsentationen, maschinellem Schlussfolgern

Maschinen müssen nicht nur die Informationen finden (durch Nutzung der Strukturierung mit XML), sondern auch verarbeiten – durch den Einsatz von Logik.

# Das „Semantische Web“

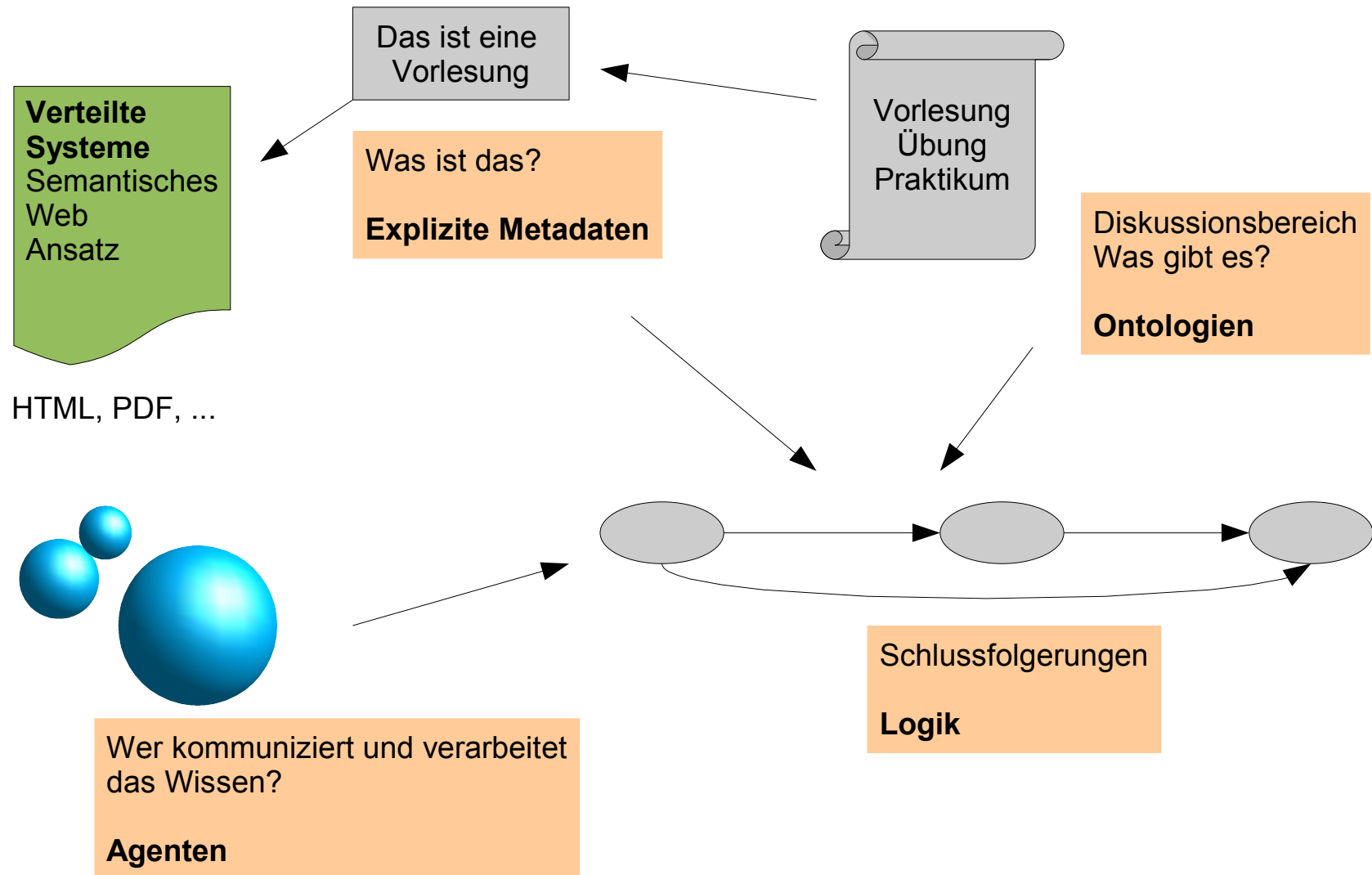
- Nach [www.w3.org](http://www.w3.org):

Das Semantische Web bietet ein allgemeines Rahmenwerk, welches das Austauschen von Daten und deren Weiterverwendung über Applikation, Unternehmen und Gruppen hinweg fördert. Es ist ein gemeinsame Unternehmung, geleitet durch das W3C unter Mitarbeit einer großen Zahl von Forschern und Industriepartnern.

Es basiert auf dem Resource Description Framework (RDF), welches eine Vielzahl von Anwendungen durch die Verwendung von XML für die Syntax und URIs für die Namensgebung integriert.

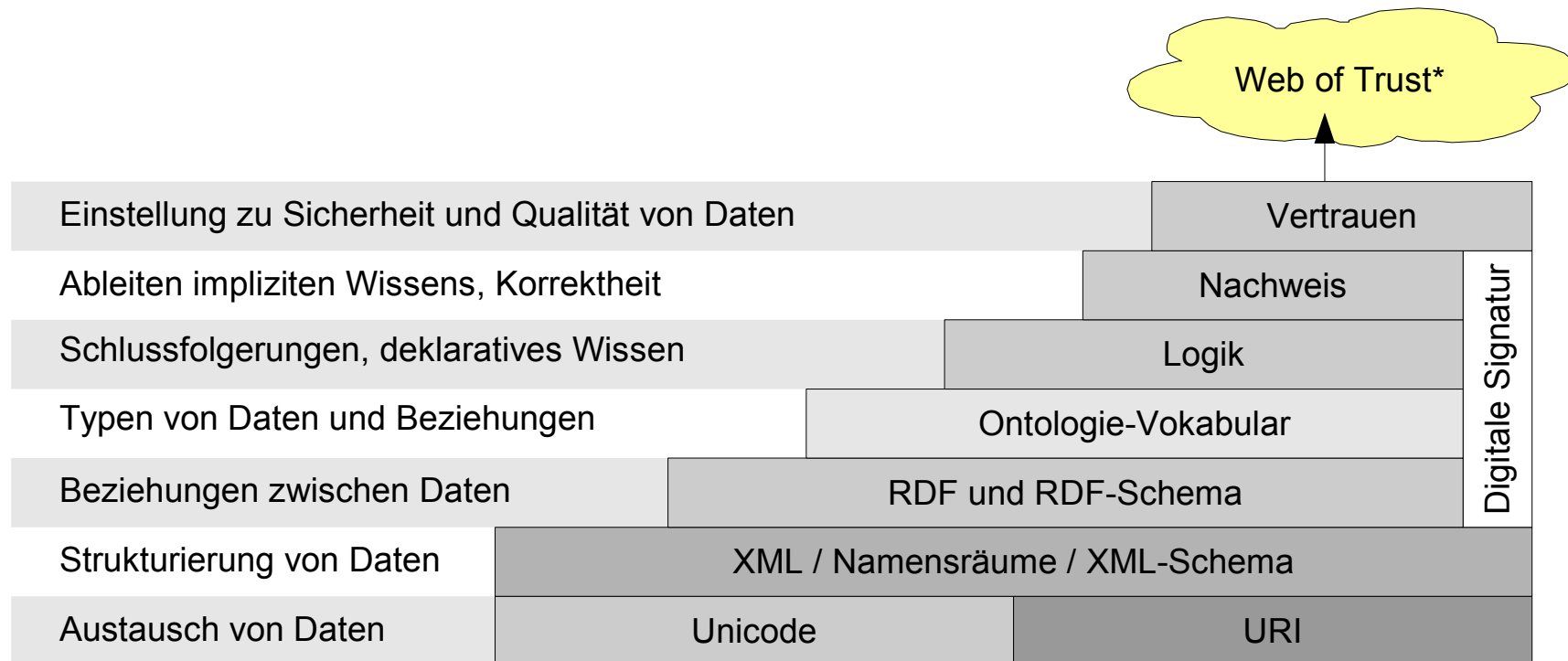


# Ansatz



# Das „Semantische Web“

- Vision des Semantic Web nach Tim Berners-Lee



\*nicht zu verwechseln mit dem Web of Trust im Kontext von PGP

# XML-basierte Wissensrepräsentation

- Simple HTML Ontology Extension (SHOE)
  - XML-konforme HTML-Erweiterung zum Sammeln und Repräsentieren von semantischen Informationen bezüglich Webressourcen (XML-Erweiterung)
- Ontology Markup Language (OML)
  - Erweiterung von SHOE mit verschiedenen Schichten, einschließlich einer Abbildung auf RDF
- Ontology Interchange Language (OIL)
  - von RDF abgeleiteter Standard zur Wissensrepräsentation
- DAML+OIL (Darpa Agent Markup Language / OIL)
  - auf Beschreibungslogik basierender Standard zur Erzeugung von Ontologien, auf RDF aufsetzend
- Topic Maps (ISO/IEC 13250)
  - Generischer Standard zur Annotation von Dokumenten

# XML-basierte Wissensrepräsentation

- RDF (Resource Description Framework/Schema Language)
  - Beschreibung von Metadaten für Webressourcen; verschiedene Beschreibungsschemata oder Ontologien können importiert oder erzeugt werden
- Web Ontology Language (OWL)
  - Beschreibungen von Ontologien

# Resource Description Framework (RDF)

# Fakten repräsentieren

- XML als praktisches Mittel, um Fakten zu repräsentieren

```
<vorlesung name="Techniken und Dienste des Internets">  
  <dozent>Michael Zapf</dozent>  
</vorlesung>
```

```
<dozent name="Michael Zapf">  
  <veranstaltet>Techniken und Dienste des Internets</veranstaltet>  
</dozent>
```

```
<lehrangebot>  
  <dozent>Michael Zapf</dozent>  
  <vorlesung>Techniken und Dienste des Internets</vorlesung>  
</lehrangebot>
```

oder doch nicht?

Alle drei Blöcke drücken dasselbe aus!

# RDF

- Resource Description Framework
  - eher Datenmodell anstatt Sprache
  - **Aussage** (Statement) als zentrale Struktur: <Objekt, Attribut, Wert> oder auch interpretiert als <Subjekt, Prädikat, Objekt>
  - XML ist **eine** mögliche Notation
- RDF-Schema
  - unglückliche Begriffswahl – suggeriert Parallele zu „XML-Schema“
  - Definition der Terminologie
    - Wie lautet das Vokabular des RDF-Datenmodells?
  - Definition der Beziehungen von Begriffen

Dozent ist eine Subklasse von Wissenschaftlicher Angestellter

# RDF Schema

```
<wissenschaftlicherAngestellter>Peter Müller</wissenschaftlicherAngestellter>  
<professor>Kurt Geihs</professor>  
<vorlesung name="Techniken und Dienste des Internets">  
  <veranstaltetVon>Michael Zapf</veranstaltetVon>  
</vorlesung>
```

- Wer sind die wissenschaftlichen Angestellten?
  - XPath: `//wissenschaftlicherAngestellter` → Peter Müller
- Unbefriedigende Antwort
  - Alle Professoren sind angestellt (Detail ignorierend, dass die meisten Professoren Beamte sind)
  - Vorlesungen werden nur von wiss. Angestellten gehalten
  - Es fehlen also Antworten, die dem „semantischen Modell“ entspringen



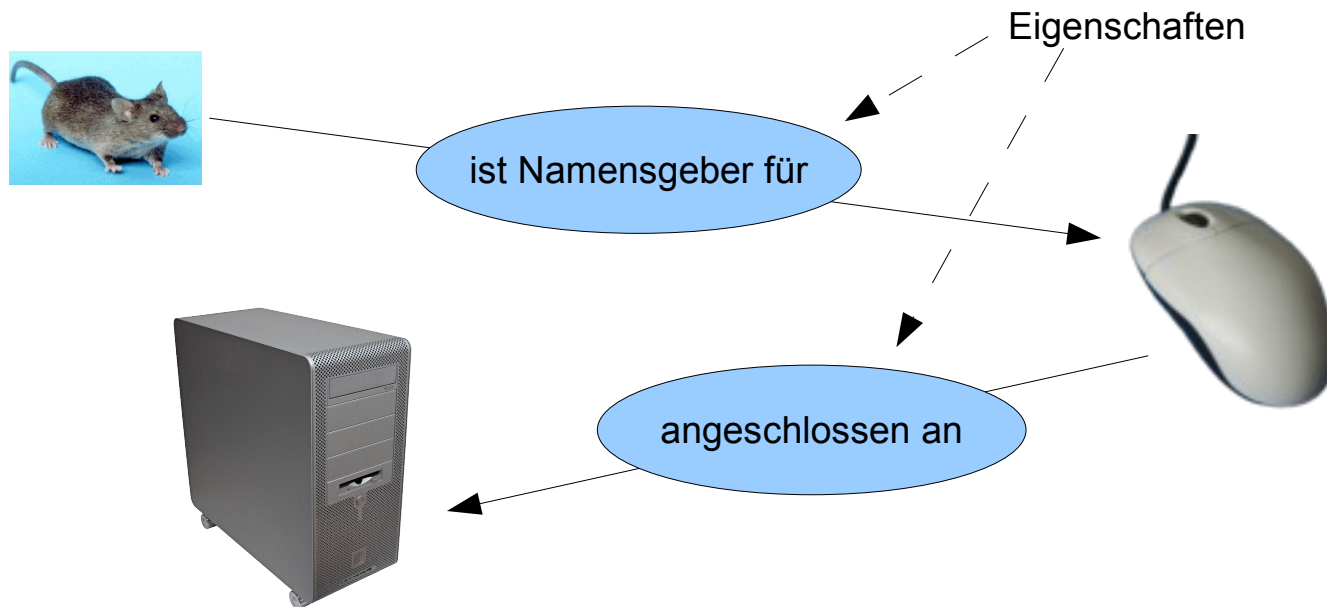
# Grundlagen von RDF

- Ressourcen
  - Objekt
  - Gegenstand, über den man reden kann (nicht nur greifbare Objekte, auch Vorgänge!)
  - Autor, Buch, Verleger, Ort, Suchanfrage
- Ressourcen werden mit URIs adressiert
  - Vermeidet Verwechslung von Bedeutungen mit gleichen Namen (Homonyme)



# Grundlagen von RDF

- Eigenschaften (properties)
  - spezielle Art von Ressource: beschreibt Beziehung zwischen Ressourcen
  - ebenfalls per URI adressierbar
    - erleichtert auch hier das Fixieren einer Bedeutung

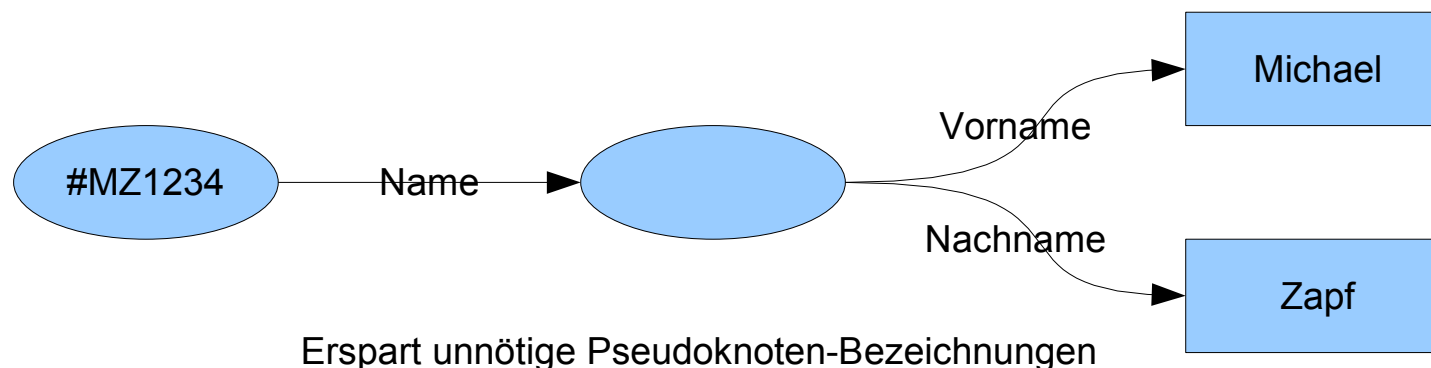


# Grundlagen von RDF

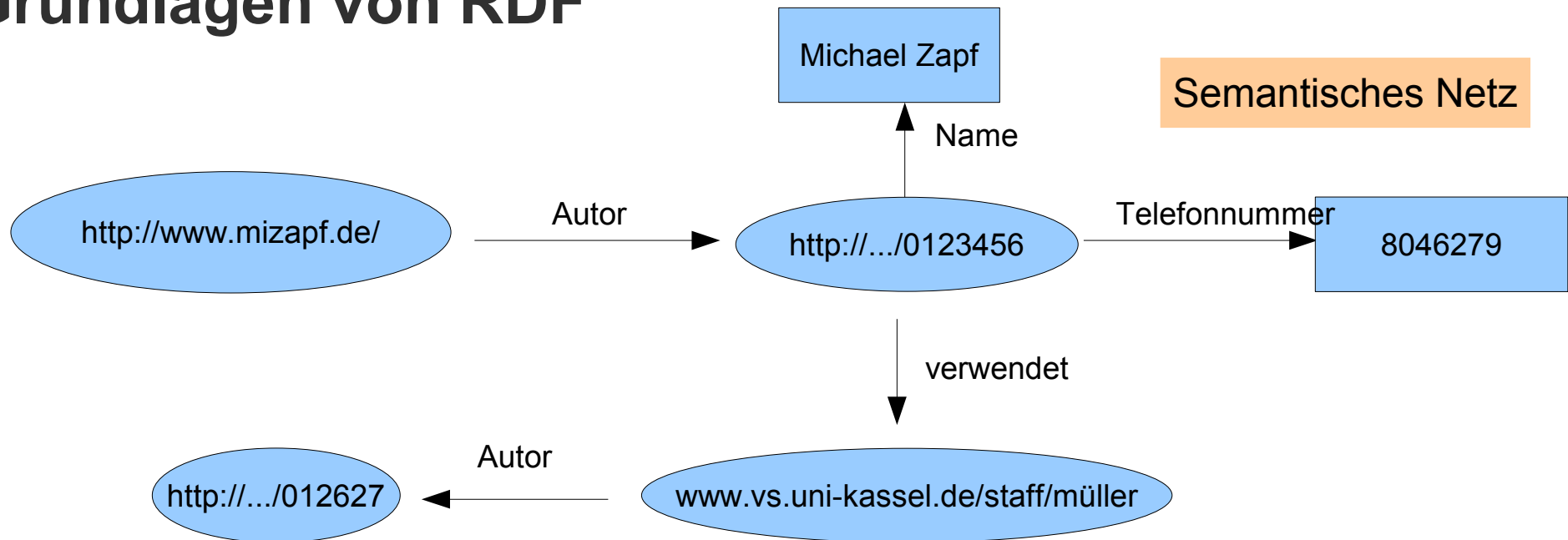
- Aussagen
  - bringen Ressourcen und Eigenschaften zusammen
  - haben Wahrheitsgehalt (wahr oder falsch)
- Beispiel
  - Michael Zapf ist der Autor der Webseite [www.mizapf.de](http://www.mizapf.de)
- etwas formaler
  - *<http://www.mizapf.de/, http://www.bedeutungen.dom/autor, Michael Zapf>* als logische Formel (x,P,y) zu interpretieren
- Nur zweistellige Prädikate in RDF
  - wenige Prädikate des täglichen Gebrauchs sind mehrstellig (z.B. „zwischen“)

# Grundlagen von RDF

- Aufbau als Sammlung von Tripeln
  - Subjekt
  - Prädikat
  - Objekt
- Subjekte sind immer eine URI-Referenz oder ein „leerer Knoten“
- Prädikate sind immer eine URI-Referenz
- Objekte sind URI-Referenzen oder Literale (z.B. Namen, Zahlen) oder ein „leerer Knoten“



# Grundlagen von RDF



<<http://www.mizapf.de/>, <http://www.bedeutungen.dom/autor>, <http://www.uni-kassel.de/personal/0123456>>  
<<http://www.uni-kassel.de/personal/0123456>, <http://www.bedeutungen.dom/name>, "Michael Zapf">  
<<http://www.uni-kassel.de/personal/0123456>, <http://www.bedeutungen.dom/telefonnummer>, "8046279">  
<<http://www.uni-kassel.de/personal/0123456>, <http://www.bedeutungen.dom/verwendet>,  
<http://www.vs.uni-kassel.de/staff/müller>>  
<<http://www.vs.uni-kassel.de/staff/müller>, <http://www.bedeutungen.dom/autor>,  
<http://www.uni-kassel.de/personal/012627>>

Grafische Darstellung ist intuitiv; man benötigt jedoch ein maschinenlesbares Format.

# Grundlagen von RDF

- XML-Repräsentation

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:meinedef="http://www.bedeutungen.dom/my-rdf-ns">

  <rdf:Description rdf:about="http://www.mizapf.de/">
    <meinedef:autor>
      <rdf:Description rdf:about="http://www.uni-kassel.de/personal/0123456">
        <meinedef:name>Michael Zapf</meinedef:name>
      </rdf:Description>
    </meinedef:autor>
  </rdf:Description>
</rdf:RDF>
```

Subjekt

Prädikat

Objekt und Subjekt der nächsten Aussage

Prädikat mit Objekt-Literal

- Reihenfolge der Beschreibungen unwesentlich
- Auch Datentypen verwendbar
  - Zahlen können z.B. von Bezeichnungen unterschieden werden

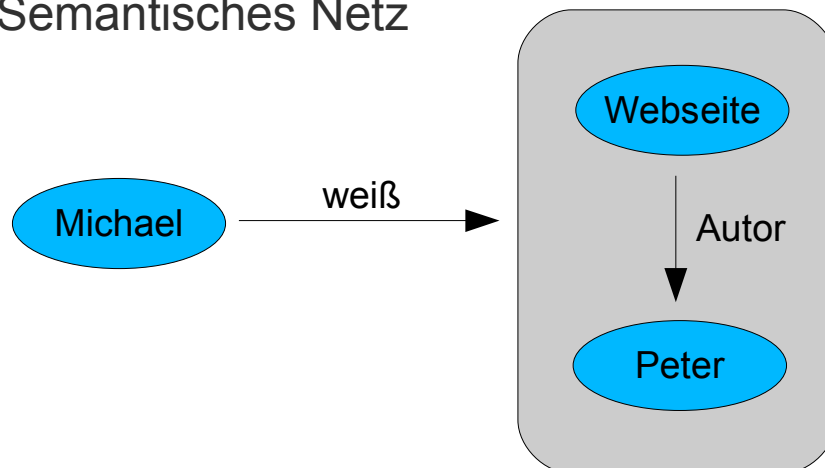
# Grundlagen von RDF

- Beschreibung von Subjekten mit `<rdf:Description>`
- Prädikate werden direkt als Kindknoten verwendet
- Objekte sind darin eingeschachtelte `<rdf:Description>`
  - Vereinfachung für Literale: Als Inhalt des Prädikatsknoten nennen
- Referenzierung von RDF-URIs
  - Vollversion: `http://www.irgendwo.dom/pfad#ressource`
  - Nutzung relativer Pfade (relativ zum Dokument)
    - `rdf:about="#ressource"` (Ressource ist hier in diesem Dokument definiert)
    - `rdf:ID="ressource"` (identisch zum vorangegangenen Beispiel)
    - Mit `xml:base` kann man eine andere Basisadresse als dieses Dokument setzen

`rdf:ID` dient zur Kennzeichnung lokaler Beschreibungen; `rdf:about` erlaubt die Referenzierung lokaler, aber auch beliebiger Ressourcen außerhalb des lokalen Kontexts (Ort des Dokuments)

# Reifikation\*

- aus lateinisch *res* (Gegenstand) und *facere* (machen), auch Reifizierung
  - etwas zu einem (Erörterungs-)Gegenstand machen
  - in der Sprache als Substantivierung (das Gelb, das Rennen) oder als Nebensatzkonstruktion
    - „Sie behauptet, dass sie den Mann nicht kennt“.
    - Frage: Was behauptet sie? - Antwort: Sie kennt den Mann nicht.
- In der Logik
  - Semantisches Netz



\*sprich: [reʔifika'tsio:n]

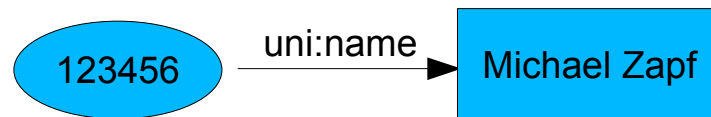


# Reifikation in RDF

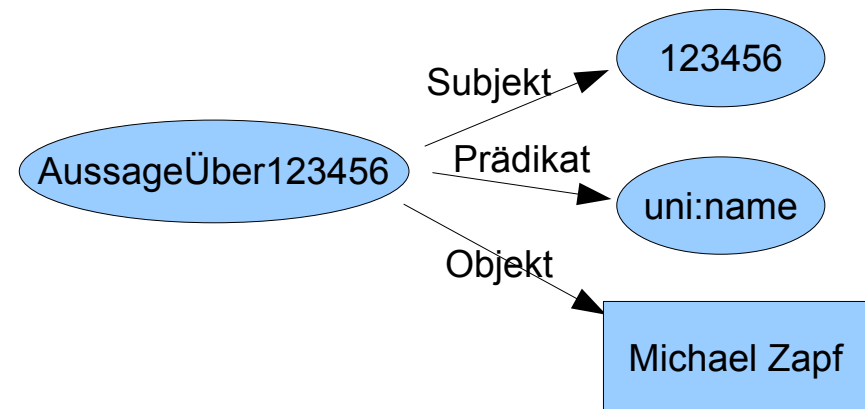
- Reifikation durch Deklaration von Aussagen

```
<rdf:Description rdf:about="123456">  
  <uni:name>Michael Zapf</uni:name>  
</rdf:Description>
```

Zuordnung einer Personalnummer  
zu einer Person  
123456 → Michael Zapf



```
<rdf:Statement rdf:about="AussageÜber123456">  
  <rdf:subject rdf:resource="123456"/>  
  <rdf:predicate rdf:resource="&uni:name"/>  
  <rdf:object>Michael Zapf</rdf:object>  
</rdf:Statement>
```



*&uni;* ist eine Entitätsreferenz

# RDF – Kritik

- Dreiwertige Prädikate müssen in zweiwertige übersetzt werden
  - nicht dramatisch, da die meisten Prädikate zweiwertig sind
- Eigenschaften sind Ressourcen
  - was zu Verwirrung bei der Modellierung führen kann
- Reifikation ist zu mächtig
  - besser eine Ebene höher; lieber Fokus auf Basisausdrucksmöglichkeiten
- XML-basierte Notation
  - gut maschinenlesbar, aber anstrengend zu formulieren
  - jedoch gibt es Hilfsprogramme und das XML muss nicht selbst geschrieben werden

# RDF-Schema

# RDF-Schema: Grundlagen

- Erlaubt Beschreibung einer thematischen Domäne (ohne selbst spezifische Vorgaben zu machen)
- Höhere Logik
  - keine Aussagen über Individuen („Michael Zapf ist ein Dozent“) → RDF
  - sondern Aussagen über Klassen/Typen von Individuen
- Verhindern unsinniger Aussagen

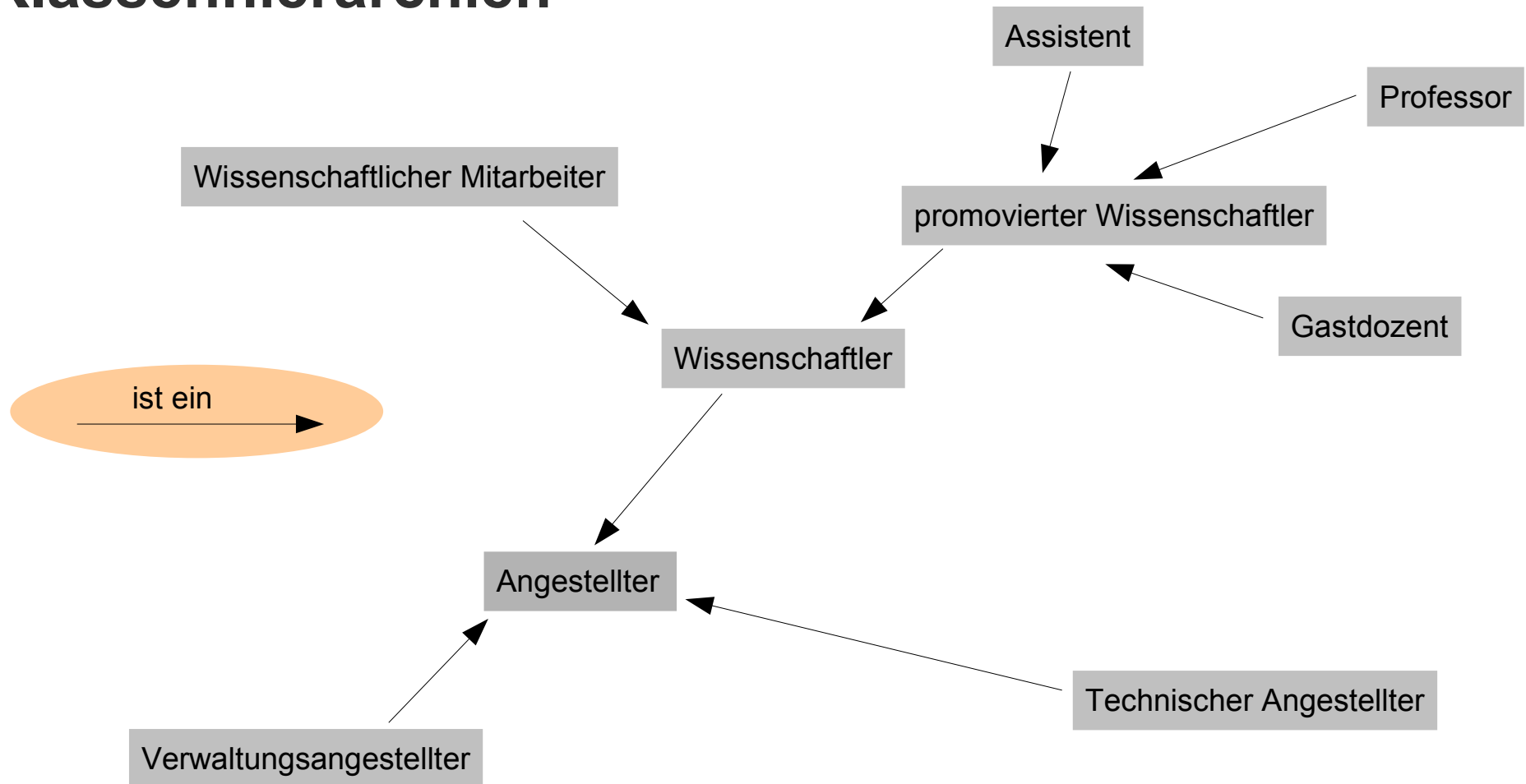
Die Vorlesung TDI wird gehalten von der Webseite <http://www.vs.uni-kassel.de/>

35 wird gehalten von Peter Müller

Festlegung der Domäne (Urbildbereich)  
der Eigenschaft

Festlegung des Zielbereichs (Bildbereich)  
der Eigenschaft

# Klassenhierarchien

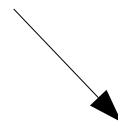


(Zum Wohle der einfachen Darstellung seien „Beamte“ auch „Angestellte“)

# Definition gültiger Werte und Vererbung

- Definition anhand der Klassen
  - Nutzung von Vererbung
- „Michael Zapf“ ist „Assistent“
  - nach RDF alleine wäre er damit nicht qualifiziert
  - entsprechend überall eintragen?
- RDF-Schema: Vererbung nutzen
  - Definieren der Eigenschaften einer ganzen Domäne

Vorlesungen können nur von **promovierten Wissenschaftlern** gehalten werden

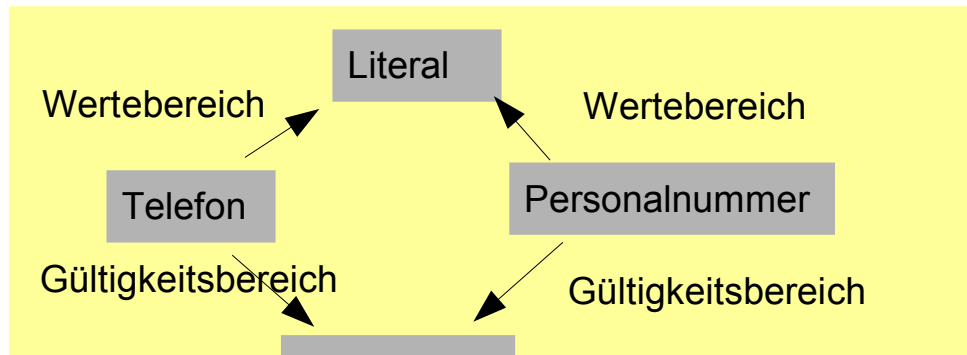


RDFS ist eine einfache **Ontologie-Definitionssprache.**

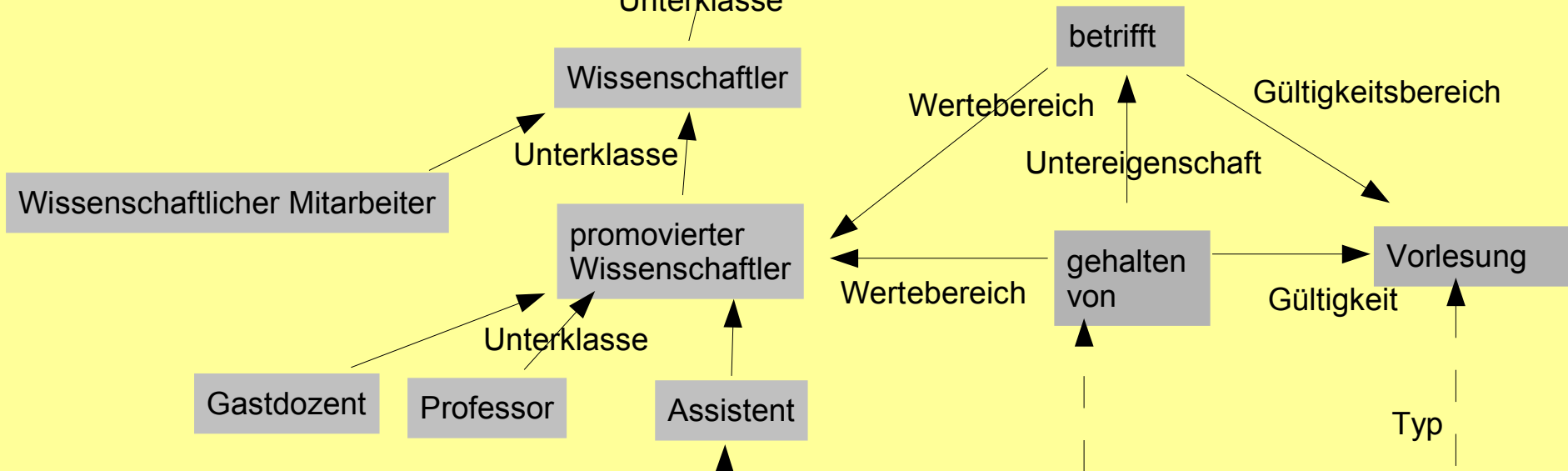
# Vererbung

- Konzept aus der Objektorientierung
- Unterschiede
  - Eigenschaften sind nicht „objektlokal“, sondern immer global
  - Man kann Eigenschaften definieren, die auf eine Klasse zutreffen, ohne die Klasse selbst zu ändern
- Auch eine Vererbung der Eigenschaften selbst möglich
  - Untereigenschaften (subproperties)
  - Q ist Untereigenschaft von P genau dann, wenn gilt:  $Q(x,y) \rightarrow P(x,y)$ 
    - „Dozent sein“ ist Untereigenschaft von „Vortragender sein“ (wiss. Mitarbeiter können auch vortragen)

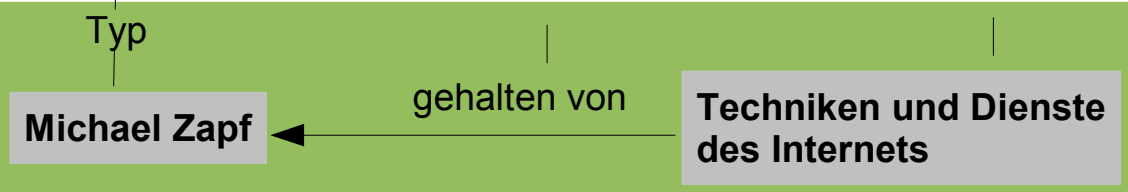
# RDF und RDFS



## RDFS



## RDF





# RDF-Schema

- Bild zeigt: RDF-Schema ist letztlich auch nur eine Anwendung von RDF
  - „Gültigkeitsbereich“, „Wertebereich“, „Untereigenschaft“, „Subklasse“ verbinden Subjekte mit Werten
- Kernklassen
  - rdfs:Resource (Klasse aller Ressourcen)
  - rdfs:Class (Klasse aller Klassen)
  - rdfs:Literal (Klasse aller Strings)
  - rdf:Property (Klasse aller Eigenschaften)
  - rdf:Statement (Klasse aller reifizierten Aussagen)
- Kerneigenschaften
  - rdf:type (Bezieht Ressource auf eine Klasse)
  - rdfs:subClassOf (Bezieht Klasse auf Superklasse)
  - rdfs:subPropertyOf (Bezieht Eigenschaft auf übergeordnete Eigenschaft)

# Weitere Merkmale

- Einschränkungen
  - rdfs:domain (Gültigkeitsbereich einer Eigenschaft)
  - rdfs:range (Wertebereich einer Eigenschaft)
  - rdfs:ConstraintResource (Klasse aller Einschränkungen)
  - rdfs:ConstraintProperty (Klasse aller Eigenschaften, welche Einschränkungen definieren)
- Reifikation
  - rdf:subject, rdf:predicate, rdf:object
- Container
  - rdf:Bag, rdf:Seq, rdf:Alt
  - rdfs:Container (Klasse aller Container)

# Formulierung in RDFS

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

  <rdfs:Class rdf:ID="assistent"> <!-- analog für Gastdozent, Professor -->
    <rdfs:Comment>Die Klasse der Assistenten. Alle Assistenten sind promovierte Wissenschaftler.
    </rdfs:Comment>
    <rdfs:subClassOf rdf:resource="#promovierterWissenschaftler"/>
  </rdfs:Class>

  <rdfs:Class rdf:ID="promovierterWissenschaftler"> <!-- analog für Wissenschaftlicher Mitarbeiter -->
    <rdfs:Comment>Die Klasse der promovierten Wissenschaftler.<rdfs:Comment>
    <rdfs:subClassOf rdf:resource="#wissenschaftler"/>
  </rdfs:Class>

  <rdfs:Class rdf:ID="wissenschaftler">
    <rdfs:Comment>Die Klasse der Wissenschaftler.<rdfs:Comment>
    <rdfs:subClassOf rdf:resource="#angestellter"/>
  </rdfs:Class>

  <rdfs:Class rdf:ID="angestellter">
    <rdfs:Comment>Die Klasse der Angestellten.<rdfs:Comment>
  </rdfs:Class>
```

# Formulierung in RDFS

```
<rdfs:Class rdf:ID="vorlesung">
  <rdfs:comment>Die Klasse der Vorlesungen.</rdfs:comment>
</rdfs:Class>

<rdf:Property rdf:ID="betrifft">
  <rdfs:comment>Stellt Beziehung zwischen Dozenten und Vorlesungen her</rdfs:comment>
  <rdfs:domain rdf:resource="#vorlesung"/>
  <rdfs:range rdf:resource="#dozent"/>
</rdf:Property>

<rdf:Property rdf:ID="gehalten_von">
  <rdfs:comment>Erbt die Eigenschaften der begrenzten Gültigkeit und Werte</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="#betrifft"/>
</rdf:Property>

<rdf:Property rdf:ID="telefon"> <!-- dasselbe für Personalnummer -->
  <rdfs:comment>Eigenschaft von Angestellten; Werte sind Literale</rdfs:comment>
  <rdfs:domain rdf:resource="#angestellter"/>
  <rdfs:range rdf:resource="&rdf;Literal"/>
</rdf:Property>
</rdf:RDF>
```

# Abfragen

# Abfragen in RDF

- Einsatzgebiete (use cases) für RDF-Abfragen
  - Finden einer Mailadresse
  - Finden von Informationen zu Ersatzteilen
  - Finden unbekannter Medienobjekte, etwa Neuerscheinungen
  - Finden spezieller Nachrichten über bestimmte Ereignisse im Fernsehen anhand eines RDF-EPG (Electronic Program Guide)
  - Geografische Daten sichten
  - Patientendaten durchsuchen

# Abfragen in RDF

- Candidate Recommendation (06.04.2006)
  - SPARQL = SPARQL Protocol And RDF Query Language  
<http://www.w3.org/TR/rdf-sparql-query/>
- Basiert auf dem Vergleich von Mustern im RDF-Graphen
  - einfachstes Muster: Tripel (Subjekt-Prädikat-Objekt)
  - Verwendung von Variablen anstelle von Termen

IRI (RFC 3987)  
Internationalized Resource Identifier  
in <...> eingeschlossen

- Beispiel

Gegeben: (<<http://example.org/book/book1>>, <<http://purl.org/dc/elements/1.1/title>>, "SPARQL-Anleitung")

Abfrage: 

```
SELECT ?title
WHERE
{
  <http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> ?title .
}
```

Ergebnis: title: „SPARQL-Anleitung“

# Syntax

- Literale (Stringkonstanten) in Anführungszeichen “...” oder '...'
  - optionaler Tag für die Sprache (@) oder für den Datentyp-IRI (^^)
- Zahlen direkt notieren
  - xsd:integer
  - xsd:decimal (wenn Punkt vorhanden)
  - xsd:double (wenn Exponent vorhanden)
  - xsd:boolean (für true/false)
- Variablen
  - ?Name oder \$Name (synonym)



# Weitere Beispiele

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>  
SELECT ?title  
WHERE { <http://example.org/book/book1> dc:title ?title }
```

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>  
PREFIX : <http://example.org/book/>  
SELECT $title  
WHERE { :book1 dc:title $title }
```

```
BASE <http://example.org/book/>  
PREFIX dc: <http://purl.org/dc/elements/1.1/>  
SELECT $title  
WHERE { <book1> dc:title ?title }
```

```
BASE <http://example.org/book/>  
PREFIX dcore: <http://purl.org/dc/elements/1.1/>  
SELECT ?title  
WHERE { <book1> dcore:title ?title }
```

Gleiche Abfragen



# Weitere Beispiele

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?book ?title
WHERE { ?book dc:title ?title }
```

← Abfragen mit zwei Variablen

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX : <http://example/ns#>
```

```
SELECT ?book ?title
WHERE
{ ?t rdf:subject ?book .
  ?t rdf:predicate dc:title .
  ?t rdf:object ?title .
  ?t :saidBy " Bob" .
}
```

← Abfrage einer Reifikation

# Ergebnisse

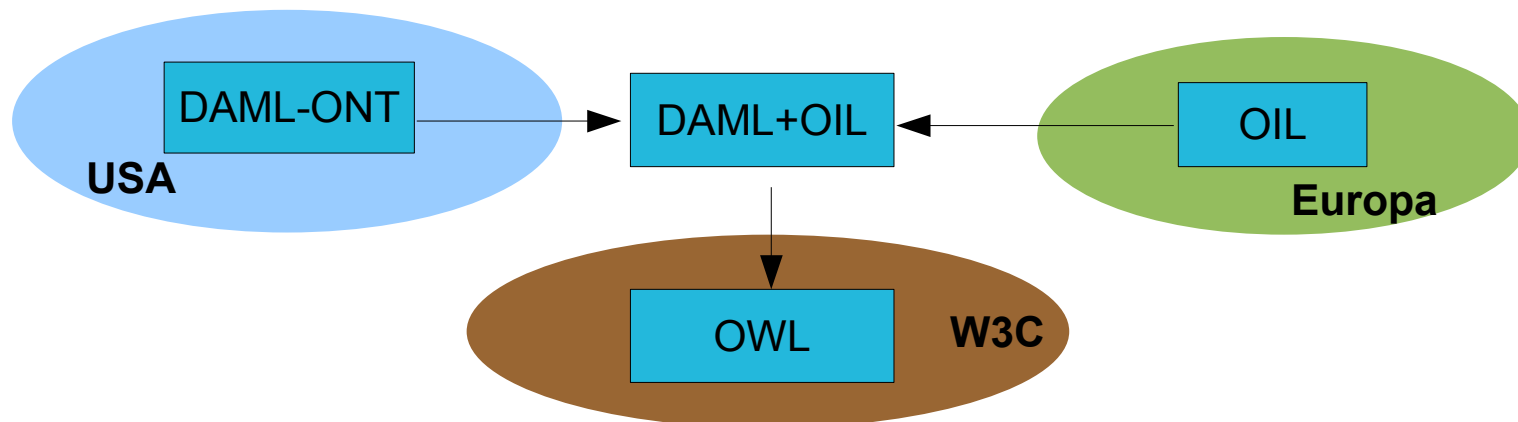
- Auch für die Suchergebnisse gibt es ein Format
  - SPARQL Query Results XML Format  
<http://www.w3.org/TR/rdf-sparql-XMLres/>
  - besteht aus Kopf und Resultatsliste

```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="x"/>
    <variable name="hpage"/>
    <variable name="name"/>
    <variable name="age"/>
    <variable name="mbox"/>
    <variable name="friend"/>
  </head>
  <results ordered="true" distinct="false">
    <result>
      <binding name="x">
        <bnode>r2</bnode>
      </binding>
      <binding name="hpage">
        <uri>http://work.example.org/bob/</uri>
      </binding>
      <binding name="name">
        <literal xml:lang="en">Bob</literal>
      </binding>
      <binding name="age">
        <literal
          datatype="http://www.w3.org/2001/XMLSchema#integer">30</literal>
      </binding>
      <binding name="mbox">
        <uri>mailto:bob@work.example.org</uri>
      </binding>
    </result>
    ...
  </results>
</sparql>
```

# OWL – Entwurf von Ontologien

# Web Ontology Language (OWL)

- Web Ontology Working Group hat weiter gehende Erfordernisse für ein „Semantic Web“ identifiziert
  - RDF ist auf binäre Prädikate beschränkt
  - RDFS ist auf Subklassen/Hierarchien von Gegenständen und Eigenschaften beschränkt
- Entwicklung einer Sprache zur Modellierung von Ontologien



# Untersprachen: Drei Geschmacksrichtungen

- OWL Full
  - Nutzt alle Eigenschaften von OWL, maximale Ausdrucksfähigkeit
  - Kombination mit RDF/RDFS
  - Selbst Änderung der vordefinierten Eigenschaften möglich (man könnte z.B. die Menge der Klassen beschränken)
  - Voll aufwärtskompatibel zu RDF: Jedes gültige RDF-Dokument ist auch ein gültiges OWL-Full-Dokument; jede Folgerung aus dem RDF-Schema ist auch eine Folgerung in OWL-Full
  - Nachteil: So mächtig, dass es nicht mehr entscheidbar ist → keine Automatisierung

# Drei Geschmacksrichtungen

- OWL DL (description logic)
  - Einschränkung von OWL Full, sodass Entscheidbarkeit gewahrt bleibt
    - so darf u.a. eine Klasse nicht Instanz einer anderen Klasse sein
  - Teilmenge der Prädikatenlogik erster Stufe
  - Entscheidbar: Alle Berechnungen terminieren in endlicher Zeit
  - Abgeschlossenheit: Alle Folgerungen sind entscheidbar
  - Nachteil: Nicht mehr voll kompatibel zu RDF/RDFS; jedoch ist jedes gültige OWL-DL-Dokument auch ein gültiges RDF-Dokument

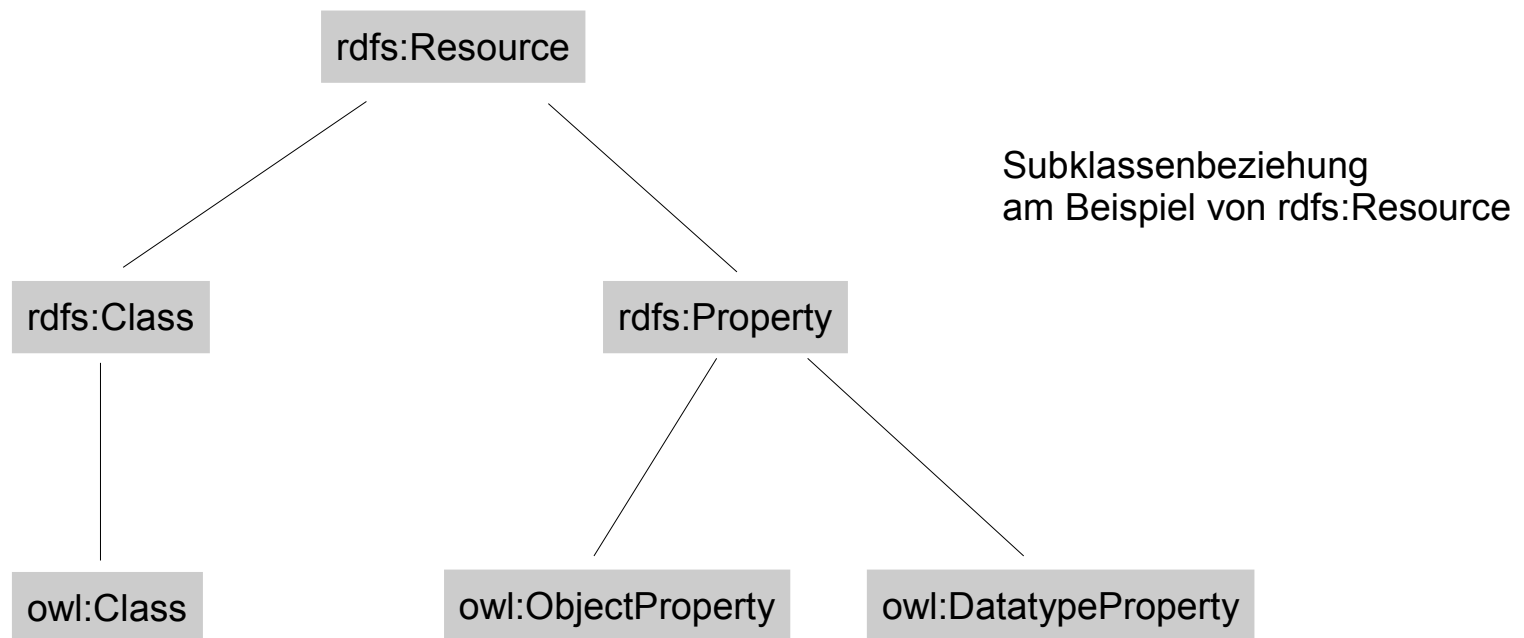
# Drei Geschmacksrichtungen

- OWL Lite
  - Weitere Einschränkung von OWL DL
    - Kardinalitäten nur noch 0 oder 1
    - keine aufgezählten Klassen
    - keine Disjunktheit
  - Vorteil
    - leicht zu lernen
    - leicht zu implementieren

<http://www.w3.org/TR/owl-features/>



# Beziehung zwischen OWL und RDF/RDFS



OWL definiert owl:Class neu – als Ableitung von rdfs:Class – da in OWL-DL/Lite Beschränkungen gelten: So darf es keine Klasse von Klassen geben. In OWL Full ist owl:Class gleich rdfs:Class.

# OWL-Notation

- RDF-Syntax
  - hier weiter verwendet, jedoch etwas unübersichtlich (schwierig zu lesen)
- XML-basierte eigenständige Syntax
  - OWL XML Presentation Syntax: <http://www.w3.org/TR/owl-xmlsyntax/>
  - leichter zu lesen als die RDF-Syntax
- Abstrakte Syntax
  - kompakter und leichter zu lesen als die RDF-Syntax
  - in Spezifikation benutzt
- Grafisch
  - Verwendung von UML

# Grundlagen von OWL

- OWL-Dokument wird „OWL-Ontologie“ genannt
- Wurzel ist ein rdf:RDF-Element; definiert auch den Kopf (Header)

```
<rdf:RDF
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
```

- Ontology-Element mit Kommentaren, Versionshinweisen, Einbindung weiterer Ontologien (transitiv)

```
<owl:Ontology rdf:about="">
  <rdfs:comment>Ein Beispiel einer OWL-Ontologie</rdfs:comment>
  <owl:priorVersion rdf:resource="http://www.meinedomäne.dom/uni-ns-old"/>
  <owl:imports rdf:resource="http://www.meinedomäne.dom/personal"/>
  <rdfs:label>Universitätsontologie</rdfs:label>
</owl:Ontology>
```

# Grundlagen von OWL

- Klassenelemente

```
<owl:Class rdf:ID="gastdozent">  
  <rdfs:subClassOf rdf:resource="#promovierterWissenschaftler"/>  
</owl:Class>
```

- Disjunktheit

```
<owl:Class rdf:about="#gastdozent">  
  <owl:disjointWith rdf:resource="#professor"/>  
  <owl:disjointWith rdf:resource="#assistent"/>  
</owl:Class>
```

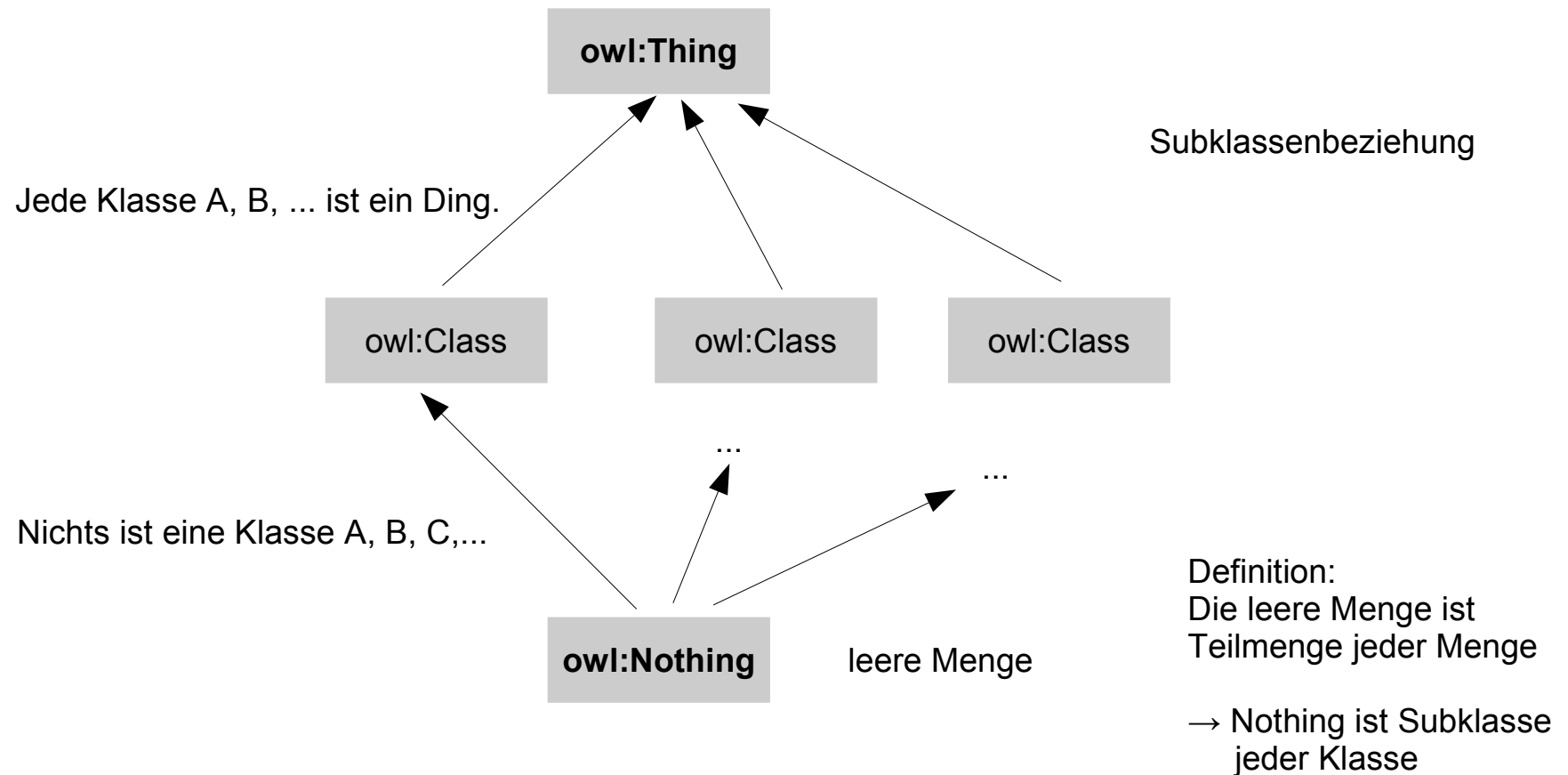
Ein Gastdozent ist kein Assistent und kein Professor (d.h. hier angestellt)

- Äquivalenz

```
<owl:Class rdf:ID="dozent">  
  <owl:equivalentClass rdf:resource="#promovierterWissenschaftler"/>  
</owl:Class>
```

# Grundlagen von OWL

- Vordefinierte Klassen



# Grundlagen von OWL

- Eigenschaftselemente (Property elements)
  - Objekteigenschaften: bringen Objekte untereinander in Beziehung
  - Datentypeigenschaften: bringen Objekte und Datentypen in Beziehung
    - Beispiel: Telefonnummer, Titel, Alter mit den entsprechenden Datentypen
  - OWL definiert keine eigenen Datentypen, sondern nutzt jene von XML-Schema.
- Weitere Merkmale
  - Eigenschaften können per Klasse eingeschränkt werden (<owl:Restriction>)
  - Existenz-/Allquantor (<owl:someValuesFrom>, <owl:allValuesFrom>)
  - Kardinalitäten (<owl:minCardinality>, <owl:maxCardinality>)
  - Vereinigung, Schnitt, Komplement von Klassen definierbar

# Beispiel

```
<!DOCTYPE owl [  
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">  
>  
  
<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:xsd="http://www.w3.org/2001/XMLSchema#"  
  xmlns:owl="http://www.w3.org/2002/07/owl#"  
  xmlns="http://www.meinedomäne.dom/drucker.owl#">  
  
  <owl:Ontology rdf:about="">  
    <owl:versionInfo>Mein Beispiel 1.0 vom 19.07.2007</owl:versionInfo>  
  </owl:Ontology>  
  
  <owl:Class rdf:ID="produkt">  
    <rdfs:comment>Produkte einer Klasse.</rdfs:comment>  
  </owl:Class>  
  
  <owl:Class rdf:ID="drscn">  
    <rdfs:comment>Drucker und Scanner</rdfs:comment>  
    <rdfs:label>Gerät</rdfs:label>  
    <rdfs:subClassOf rdf:resource="#produkt"/>  
  </owl:Class>
```



# Beispiel

(1) |  
(2) anonyme Klasse,  
geschnitten  
mit 1

```
<owl:Class rdf:ID="hpProdukt">
  <rdfs:comment>Produkte von HP werden von Hewlett-Packard hergestellt</rdfs:comment>
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#produkt"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hergestellt_von"/>
      <owl:hasValue rdf:datatype="&xsd:string">Hewlett-Packard</owl:hasValue>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>

<owl:Class rdf:ID="drucker">
  <rdfs:comment>Klasse der Drucker (verschiedene Typen)</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#drscn"/>
</owl:Class>

<owl:Class rdf:ID="bueroDrucker">
  <rdfs:comment>Klasse der persönlich zugeordneten Drucker</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#drucker"/>
</owl:Class>

<owl:Class rdf:ID="hpDrucker">
  <rdfs:comment>HP-Drucker sind spezielle Drucker und Produkte von HP</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#drucker"/>
  <rdfs:subClassOf rdf:resource="#hpProdukt"/>
</owl:Class>
```





# Beispiel



```
<owl:Class rdf:ID="laser">
  <rdfs:comment>Jene Drucker mit Lasertechnik</rdfs:comment>
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#drucker"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#drucktechnik"/>
      <owl:hasValue rdf:datatype="&xsd:string">laser</owl:hasValue>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>

<owl:Class rdf:ID="hpLaserJet">
  <rdfs:comment>Laserdrucker von HP</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#laser"/>
  <rdfs:subClassOf rdf:resource="#hpDrucker"/>
</owl:Class>

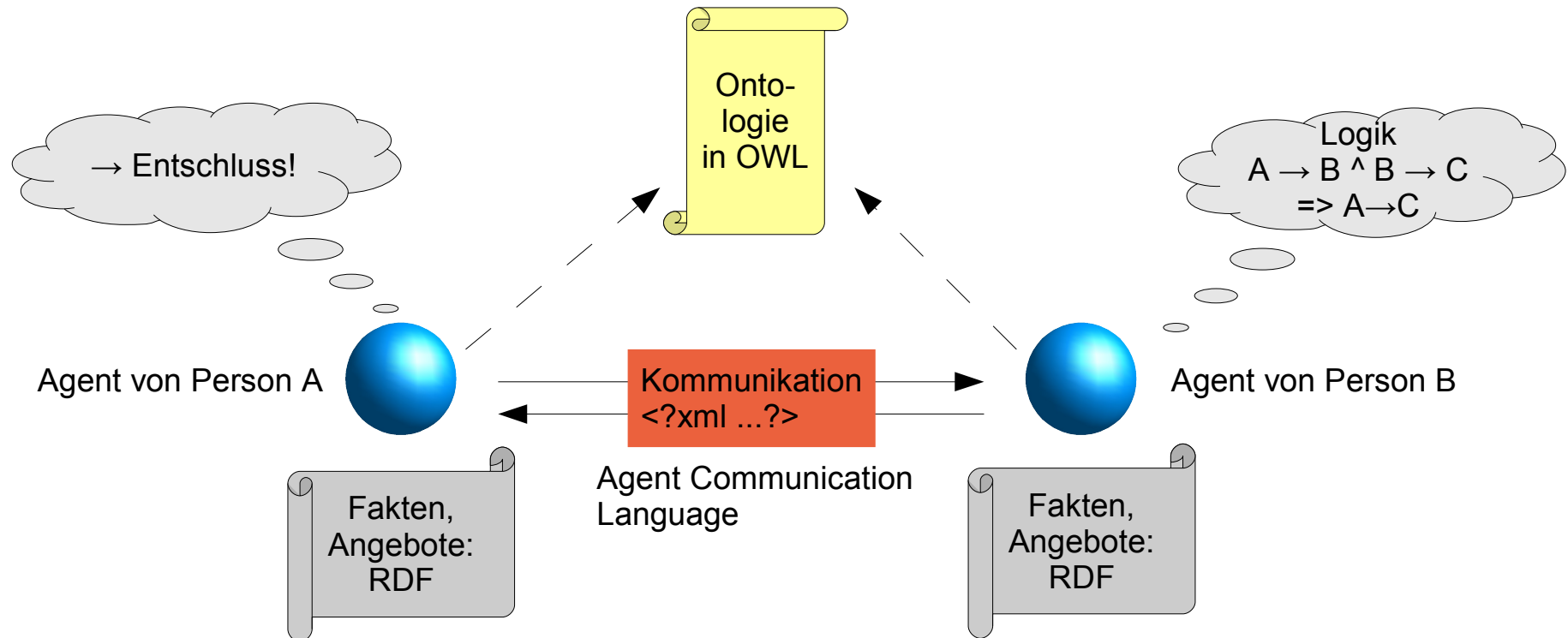
...
<owl:DatatypeProperty rdf:ID="hergestellt_von">
  <rdfs:domain rdf:resource="#produkt"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

...
```

# Anwendung

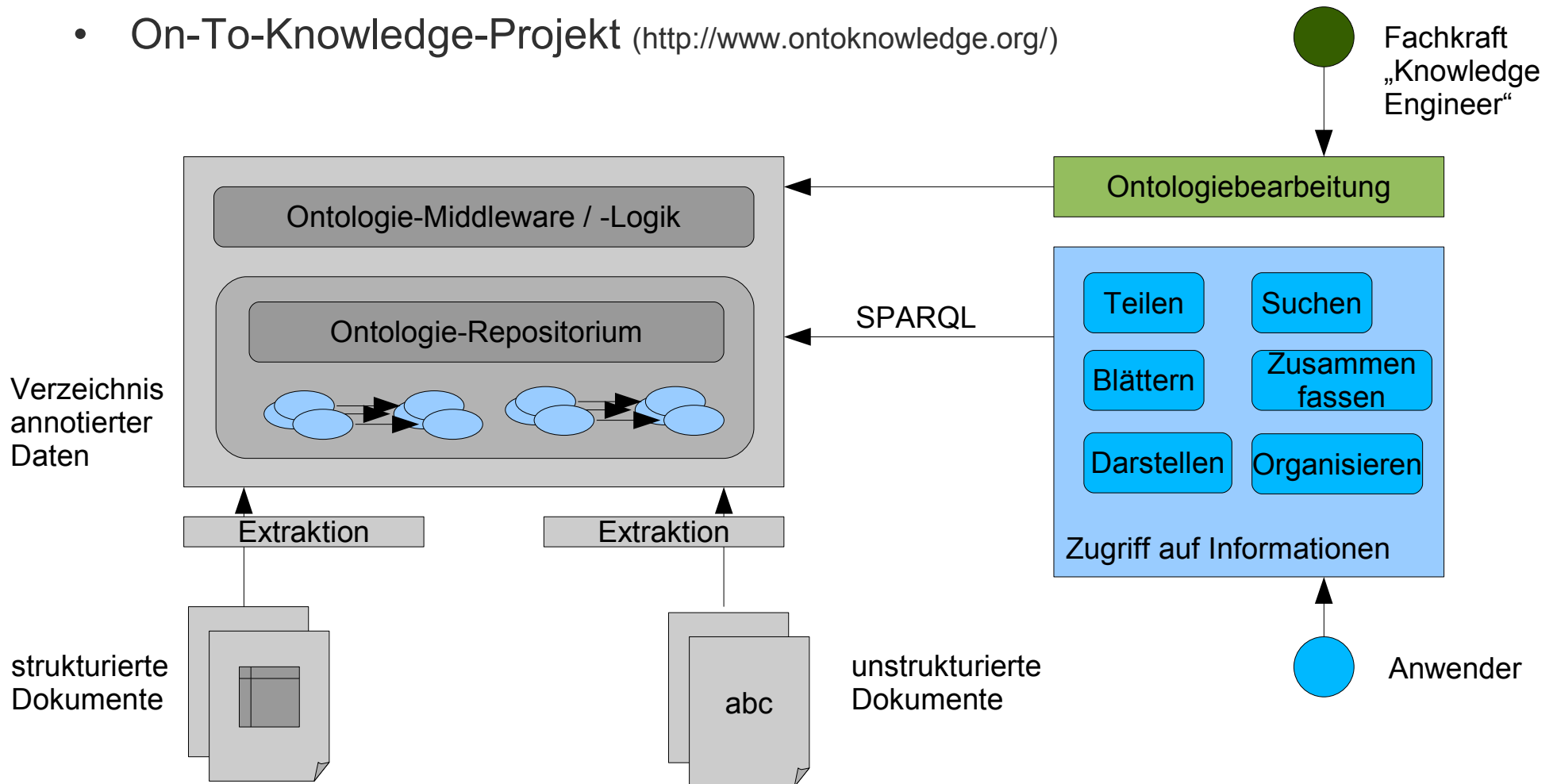
# Szenario

- Semantic Web: Nähe zu autonomen Softwareagenten



# Realisierung

- On-To-Knowledge-Projekt (<http://www.ontoknowledge.org/>)



# Einsatz im Semantic Web

- Wird das Semantic Web entstehen? Ist es schon da?
- Woher kommen die Ontologien?
  - Viele Ontologien selbstgeschrieben, einige durch maschinelles Lernen, andere durch Sprachanalyse
  - Nutzen von vorhandenen Datenbanken
- Woher kommt der Markup?
  - Nicht anzunehmen, dass dies per Hand gelingt
  - Analyse von Dokumenten, Nutzen von vorhandenen Datenbeständen
- Woher kommen die Hilfsprogramme?
  - Schon in Arbeit: Editorprogramme, Versionierung, Speicherung, Visualisierung usw.

# Einsatz im Semantic Web

- Wie geht man mit der Vielzahl von Ontologien um?
  - Abbildung von Ontologien?
  - Verhandlung zwischen Agenten?
  - Linguistische Analyse?

Aussicht: Wahrscheinlich nimmt das Semantic Web in der akademischen Welt seinen Anfang, ähnlich wie das WWW selbst.  
Später kommen Anwendungen wie E-Commerce hinzu.

# Literatur

- Semantic Web: <http://www.w3.org/2001/sw/>
- A Semantic Web Primer, Grigoris Antoniou und Frank van Harmelen, MIT Press
- RDF
  - Primer: <http://www.w3.org/TR/rdf-primer/>
  - Concepts and Abstract Syntax: <http://www.w3.org/TR/rdf-concepts/>
  - Vocabulary Description Language: <http://www.w3.org/TR/rdf-schema/>
  - RDF/XML Syntax Specification: <http://www.w3.org/TR/rdf-syntax-grammar/>
- OWL
  - Use cases and requirements, <http://www.w3.org/TR/webont-req/>
  - Web Ontology Language Reference, <http://www.w3.org/TR/owl-ref/>