Karlsruher Institut für Technologie

Semantische Integration: Grundlagen, RDF & OWL

Informationsintegration und Web Portale
Wintersemester 2013/2014

FZI Forschungszentrum Informatik am Karlsruhe Institut für Technologie (KIT)
Dr. Stefan Zander
Haid-und-Neu-Straße 10-14
76131 Karlsruhe
Kontakt: zander@fzi.de

…oder

Ontologie-basierte Informationsintegration

Grundlagen und Technologien

What you will hear in this Lecture...

- Dimensions of information integration
- Different types of heterogeneity
- Ontologies
- Ontology Languages
- Ontology-based Information Integration

We will not talk about...

- Formal Semantics of RDF and OWL
- Foundations of Logics
- Details of XML Technologies
- Different Ontology Languages and Systems
- Inferencing in Detail
- Linked Data

Overview of the Lecture

- Grundlegende Begriffe und Konzepte
  - Integriertes Informationssystem
  - Heterogenität
- Ontologien
  - Definitionen
  - Eigenschaften
  - Formale Sprachen für Ontologien
- Ontologie-Sprachen
  - RDF
  - RDF/S
  - OWL
- Beispiel – Semantische Integration
  - Semantisches Mapping von Wissensbasen

Overview of the Lecture

- **Grundlegende Begriffe und Konzepte**
  - **Integriertes Informationssystem**
  - **Heterogenität**
- Ontologien
  - Definitionen
  - Eigenschaften
  - Formale Sprachen für Ontologien
- Ontologie-Sprachen
  - RDF
  - RDF/S
  - OWL
- Beispiel – Semantische Integration
  - Semantisches Mapping von Wissensbasen

Grundlegende Begriffe und Konzepte

For the integration of IT systems, we distinguish between two classes:

- **Information Integration**
  - Integration of existing data sources
  - Related concepts are
    - Information fusion
    - Data consolidation
    - Data cleansing
    - Data Warehousing
  - Focus: Query languages, information retrieval, and processing of huge amounts of data

- **Application Integration** (aka *Enterprise Application Integration*)
  - Integration of IT processes
  - Some similarities to information integration:
    - Mapping and consolidation of heterogeneous structures
    - Consolidation of terms and their semantics
  - Focus: Exchange of messages + harmonization of interfaces

Integrated Information Systems

- **Objective**
  Grant and **simplify access** to a number of **existing information systems** by means of a centralized and **integrated component** the exhibits a common, standardized **interface** for human and machine **consumption**.

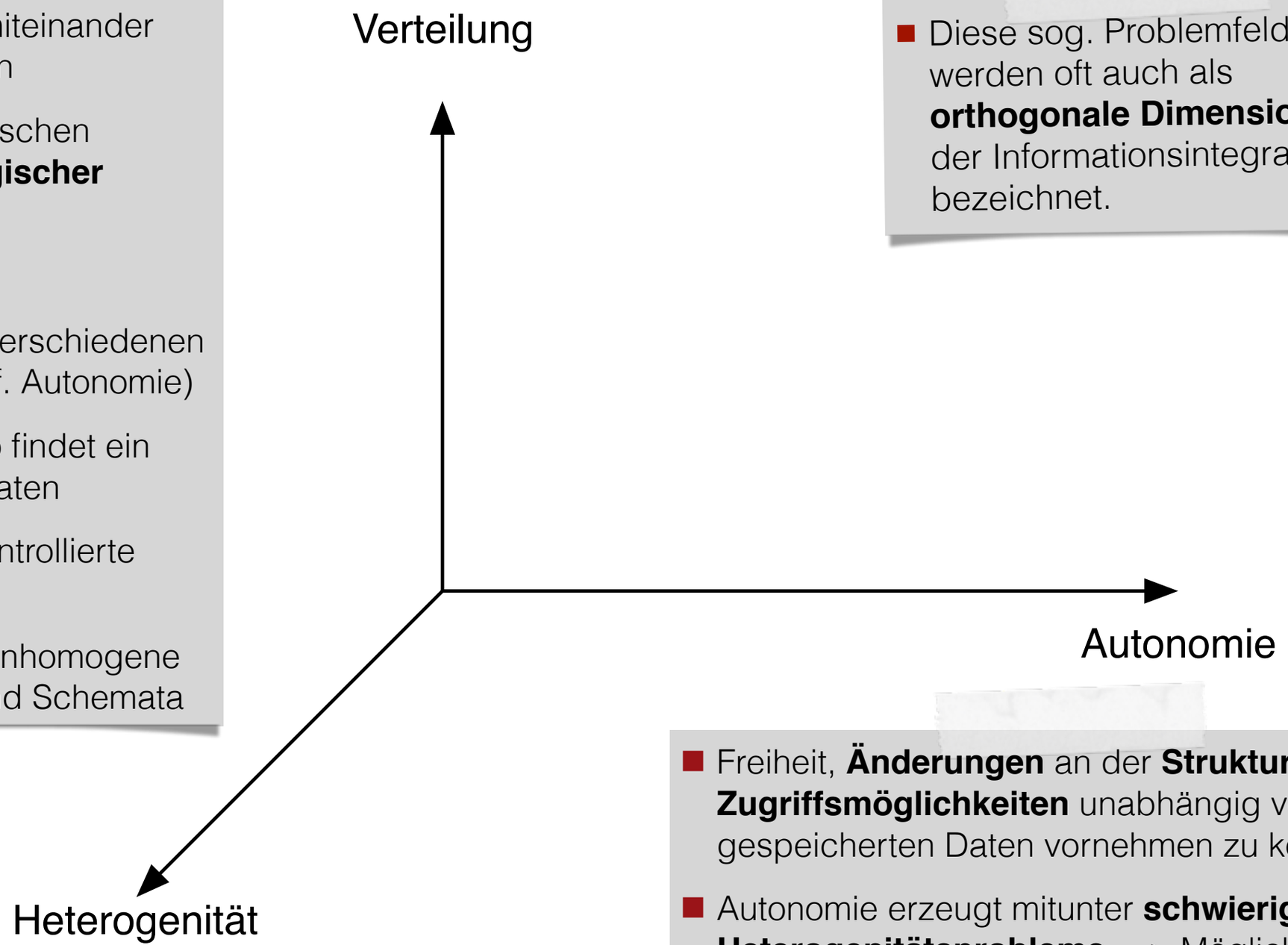
- Integrated information systems…
  - …offer a **uniform** and **coherent view** on the data and the data sources being wrapped
  - …employ **declarative query languages** or **query tools** to support query formulation
  - …exhibit a number of **well-defined technical interfaces** that allow for information retrieval or programmatic data access
  - …integrate over **existing system infrastructures** and distributed data sources

Positive Aspects of Integrated Information Systems

- Enable the realization of **structured information retrieval**
- Unified and simplified **access** for users and machines
- Increase **Recall** (IR)
- Record of **provenance** and **meta information** (technical meta data and annotations – cf. meta data in Web search)
- Increase **reliability** through redundancy
- Ensure **completeness** of results in ad-hoc information retrieval
- Controlled **redundancy** (sometimes)

Dimensionen der Informationsintegration

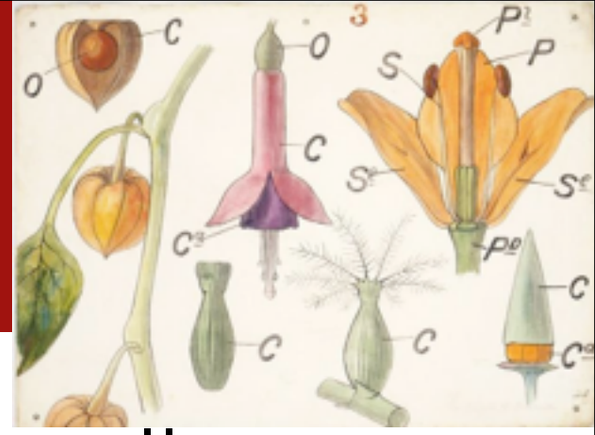
- **Verteilung der Daten** auf unterschiedlichen miteinander vernetzten Systemen
- Unterscheidung zwischen **physischer** und **logischer** Verteilung
- Probleme
  - Daten liegen in verschiedenen Schemata vor (cf. Autonomie)
  - Lokalisation – wo findet ein Nutzer welche Daten
  - Duplikate – unkontrollierte Redundanz
  - Widersprüche – inhomogene Datenmodelle und Schemata



- Diese sog. Problemfelder werden oft auch als **orthogonale Dimensionen** der Informationsintegration bezeichnet.

- Freiheit, **Änderungen** an der **Struktur** und den **Zugriffsmöglichkeiten** unabhängig von den gespeicherten Daten vornehmen zu können
- Autonomie erzeugt mitunter **schwierige Heterogenitätsprobleme** —> Möglichkeit: Einschränkung von Datenquellenautonomie

Heterogenität



- Die Integration von heterogenen Datenquellen erfordert **Hintergrundwissen** zu den enthaltenen **Datenmodellen** und **Schemata**

- Hintergrund-, Datenmodell- und Schemawissen ist wichtig für eine **präzise** und **verlustfreie Informationsintegration**

- Definition nach Leser und Naumann:

  - Zwei Informationssystem, die nicht exakt die gleichen **Methoden**, **Modelle** und **Strukturen** zum Zugriff auf ihre Daten haben bezeichnet man als **heterogen**.

Arten von Heterogenität

- **Technische Heterogenität**
  - Unterschiede in der technischen Realisierung des Zugriffs auf Daten einer Datenquelle
- **Syntaktische Heterogenität**
  - Unterschiede in der Darstellung (Serialisierung) von Informationen (Zeichenkodierung etc.)
- **Datenmodellheterogenität**
  - Unterschiedliche Datenmodelle zur Datenrepräsentation
- **Strukturelle Heterogenität**
  - Unterschiede in der strukturellen Repräsentation von Informationen
- **Schematische Heterogenität**
  - Spezialfall der strukturellen Heterogenität
  - Unterschiede in den verwendeten Datenmodellelementen
- **Semantische Heterogenität**
  - Unterschiede in der Bedeutung der verwendeten Begriffe und Konzepte (intensionale Überlappung)

Overview of the Lecture

- Grundlegende Begriffe und Konzepte
  - Integriertes Informationssystem
  - Heterogenität
- **Ontologien**
  - **Definitionen**
  - **Eigenschaften**
  - Formale Sprachen für Ontologien
- Ontologie-Sprachen
  - RDF
  - RDF/S
  - OWL
- Beispiel – Semantische Integration
  - Semantisches Mapping von Wissensbasen

Ontology-based Information Integration

“People can’t share knowledge if they don’t speak a common language.”

[T. Davenport]

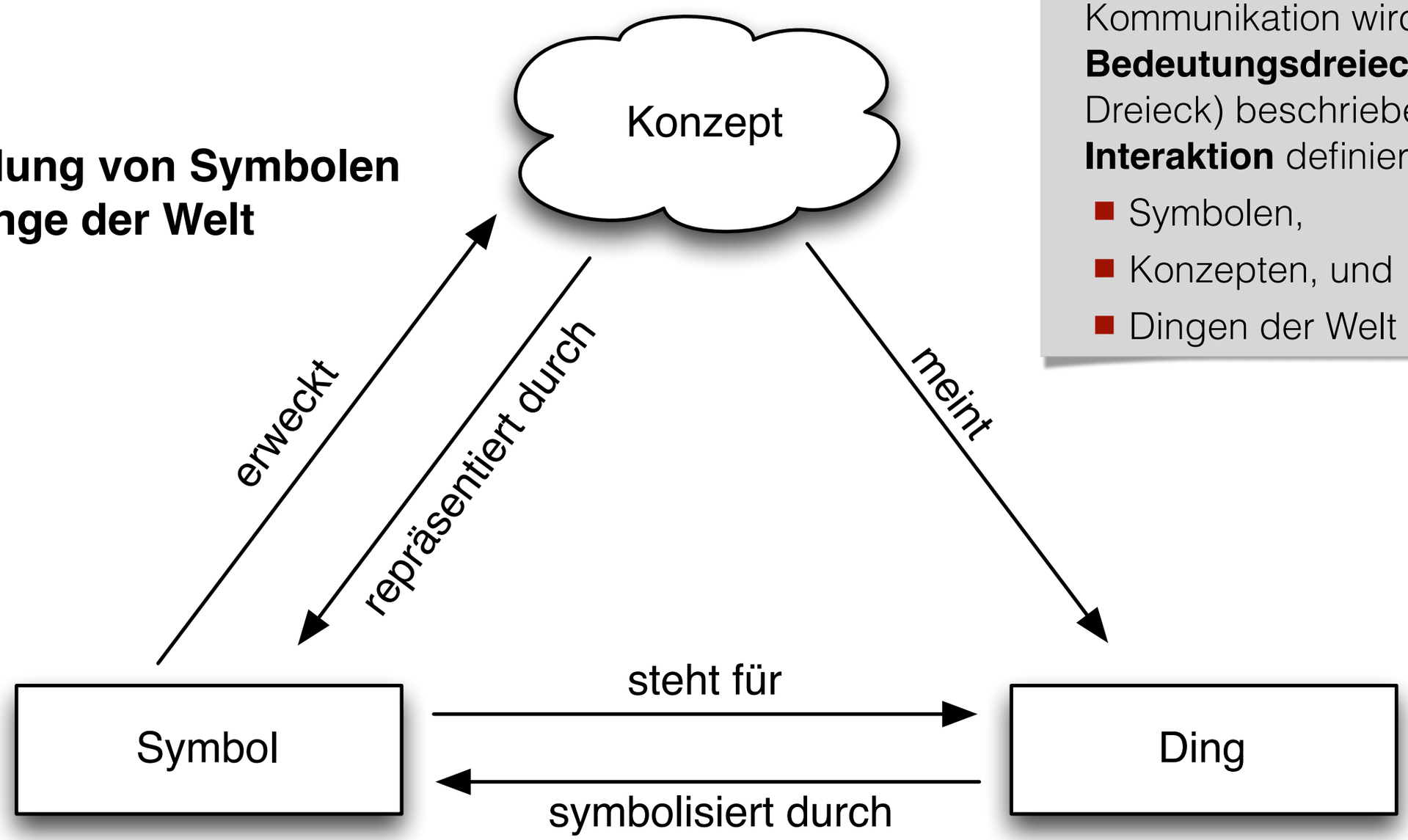
- Ontologien schaffen eine **gemeinsame Sprache** für die Integration von Informationen

- Ein **gemeinsames Verständnis** des **Anwendungsbereichs** ist wichtig für eine funktionierende Informationsintegration

Das Semiotische Dreieck

- Die Beziehung zwischen einem **Symbol** und einem **Ding** ist indirekt.

Abbildung von Symbolen auf Dinge der Welt



Der allgemeine Kontext der Kommunikation wird durch das **Bedeutungsdreieck** (semiotisches Dreieck) beschrieben, welches die **Interaktion** definiert zwischen:

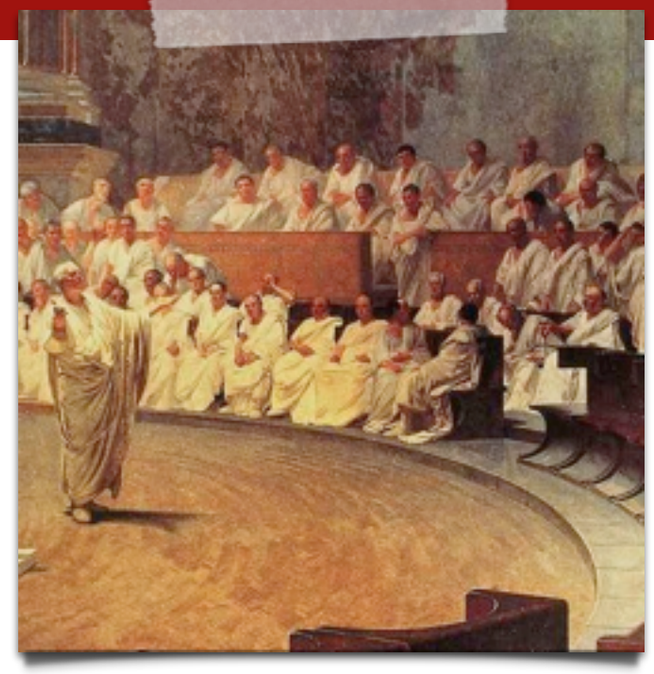
- Symbolen,
- Konzepten, und
- Dingen der Welt

Beispiele für “gemeinsame Sprache”

- **Konzepte**:
  - Person, Arbeitnehmer, Manager, Berater, Projekt
  - Firma, Hersteller, Finanzunternehmen, Versicherer, Bank
- **Relationen**:
  - MitgliedVon, TeilnehmerAn
- **Lexikoneinträge**: (*)
  - „ArbeitnehmerIn“, „SachbearbeiterIn“
  - „Firma“, „Unternehmen“, „Business“, „Arbeitgeber“
  - „Mitglied“, „Mitgliedschaft“, „Teilnehmen“, „Arbeit“

(*) **Lexikon:**
Ontologien erlauben – teilweise durch spezielle Mechanismen und teilweise durch Nutzung generischer Mechanismen – sehr gut, die **konzeptuelle Ebene** durch **n-zu-m-Beziehungen** mit einer **lexikalischen Ebene** (verschiedene Benennungen für ein Konzept/ eine Relation) zu verbinden. Das ermöglicht auch Umgang mit **mehrsprachigen Informationsquellen**.

Vorteile einer “gemeinsamen Sprache”



- ermöglicht den **Zugriff** auf abgespeichertes Wissen
  - integriert und harmonisiert
    - verschiedene Quellen
    - verschiedene Darstellungen
    - verschiedene Detailstufen und Abstraktionsebenen
- bietet **verschiedene Sichten** auf abgespeichertes Wissen
  - unter Berücksichtigung von
    - Benutzern
    - Nutzungskontext
- Fokus auf die **relevanten Aspekte**
- angemessene **Abstraktionsebenen**
- bestimmtes (aufgabenspezifisches) **Vokabular**

Was sind Ontologien und welche Rolle spielen sie für die Informationsintegration?

- Ursprünglich eine **philosophische Disziplin**
  - Die Ontologie (griechisch von, „seiend“ und λόγος, Logos, „Lehre, Wort“)
  - Analyse und Beschreibung der **Wirklichkeit**
  - Die **Lehre vom Sein** (z.B. Aristoteles “Metaphysik” IV, 1)
    - Dabei wird eine Systematik grundlegender Typen von Entitäten (Gegenstände, Eigenschaften, Prozesse) und ihrer strukturellen Beziehungen diskutiert... (Quelle: Wikipedia)
- Die **Informatik** definiert häufig den Begriff **Ontologie** wie folgt:

“An Ontology is an explicit specification of a shared conceptualisation.”

[Gruber 1993]

Ontologien zur Definition einer gemeinsamen Sprache für die Informationsintegration

"Eine Ontologie ist eine **formale**, **explizite Spezifikation** einer **geteilten Konzeptualisierung**."

- Explizite **Spezifikation**
  - formal, mathematisch fundiert, maschieneninterpretierbar
  - Logische **Schlussfolgerungen** sind berechenbar (meistens)
  - Konzepte, Relationen usw. sind **explizit** definiert
- Geteilte **Konzeptualisierung**
  - **gemeinsames Verständnis** eines Anwendungsbereiches – konsensuales Wissen
  - Erarbeitet von einer Gruppe von Personen, z.B. einer Abteilung
  - intensionale Charakterisierung der relevanten **Konzepte** und **Relationen** eines Anwendungsgebiets – des sog. *"Universe of Discourse"*
    - **Relationen** führen zu weiteren relevanten Konzepten
    - **Attribute** innerhalb einer Konzeptualisierung kennzeichnen zusätzliche relevante Charaktereigenschaften für ein domänenspezifisches Konzept
- Definiert intensionale **Regeln** und **Beschränkungen**

Weitere populäre Ontologie-Definitionen

- "An ontology defines the **basic terms** and **relations** comprising the **vocabulary** of a **topic area**, as well as the **rules** for combining terms and relations to **define extensions** to the vocabulary."
  - Neches, R.; Fikes, R.; Finin, T.; Gruber, T.; Patil, R.; Senator, T.; Swartout, W.R. Enabling Technology for Knowledge Sharing. AI Magazine. Winter 1991. 36-56

- "An ontology is a **hierarchically structured set of terms** for describing a **domain** that can be used as a **skeletal foundation** for a knowledge base."
  - B. Swartout; R. Patil; k. Knight; T. Russ. Toward Distributed Use of Large-Scale Ontologies. Ontological Engineering. AAAI-97 Spring Symposium Series. 1997. 138-148

- "An ontology provides the means for **describing explicitly** the **conceptualization** behind the **knowledge** represented in a knowledge base."
  - A. Bernaras; I. Laresgoiti; J. Correra. Building und Reusing Ontologies for Electrical Network Applications. ECAI96. 12th European Conference on Artificial Intelligence. Ed. John Wiley & Sons, Ltd. 298-302

Abstrakte und konkrete Eigenschaften von Ontologien

… in der Informatik

- **Abstrakte Eigenschaften**
  - Spiegelt das **gemeinsame Verständnis** einer Domäne wieder
    - Wird gemeinschaftlich erstellt oder erarbeitet
  - Explizites **Modell** der Domäne
    - Enthält wenig verstecktes Wissen
  - Formale **Spezifikation**
    - Das modellierte Modell kann von einem Computer interpretiert werden

- Aus den vorangegangenen Definitionen lassen sich einige **Eigenschaften** von Ontologien ableiten…

- **Konkrete Eigenschaften**
  - **Klassen**/(Konzepte/Begriffe), **Instanzen** von Klassen (Individuals/Objekte), **Eigenschaften** von Instanzen einer Klasse (Properties/Attribute) sind häufig Elemente von Ontologien
  - Zusammenhänge von Klassen, Eigenschaften und Instanzen werden häufig über **Axiome** postuliert
  - Die Bedeutung einer Ontologie basiert oft auf **Prädikatenlogik**
  - In der Interpretation wird oft eine „**offene Welt**“ unterstellt (*Open World Assumption*)

Ontologie-basierte Informationsintegration

- Eine Ontologie ist ein **Engineering-Artefakt**, das gebildet wird durch:
  - einen **spezifischen Wortschatz** zur Beschreibung einer „bestimmten Realität“ (Sichtweise, Ausschnitt, Zweckorientierung)
  - plus eine Menge von **expliziten Annahmen** bezüglich der **intendierten Bedeutungen** des Wortschatzes in **formaler Sprache**, z.B. als logische Theorie

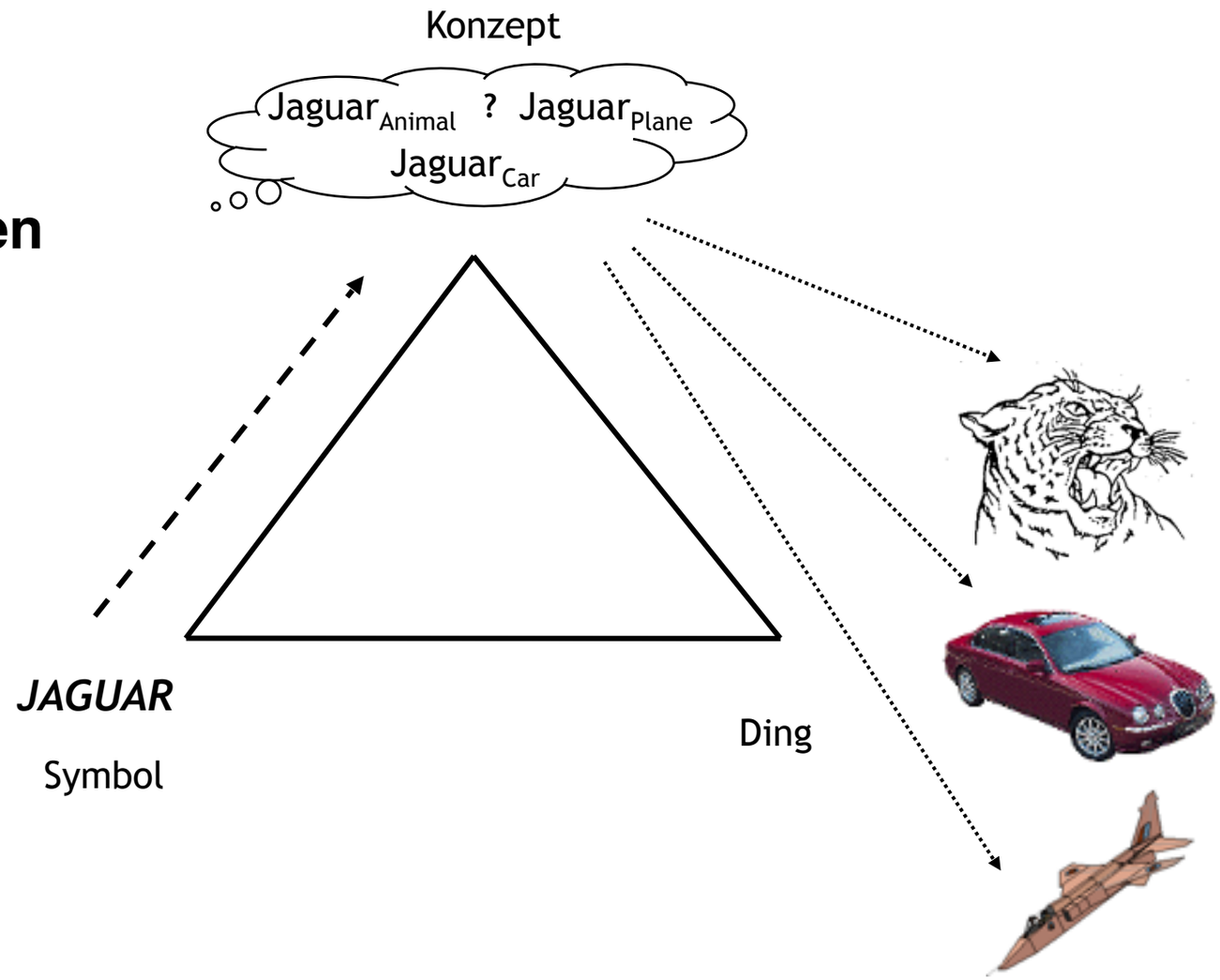
- Beides, **Wortschatz** und **Annahmen**, helfen Menschen und Maschinen, bei der Informationsintegration ein „**gemeinsames Verständnis**“ zu erreichen

- Die zugrundeliegende **logische Theorie** spezifiziert
  - **Relationen** zwischen den Symbolen
  - Verbindet Relationen mit einer **Semantik**, die die **Menge möglicher Interpretationen** des Symbols **beschränkt**

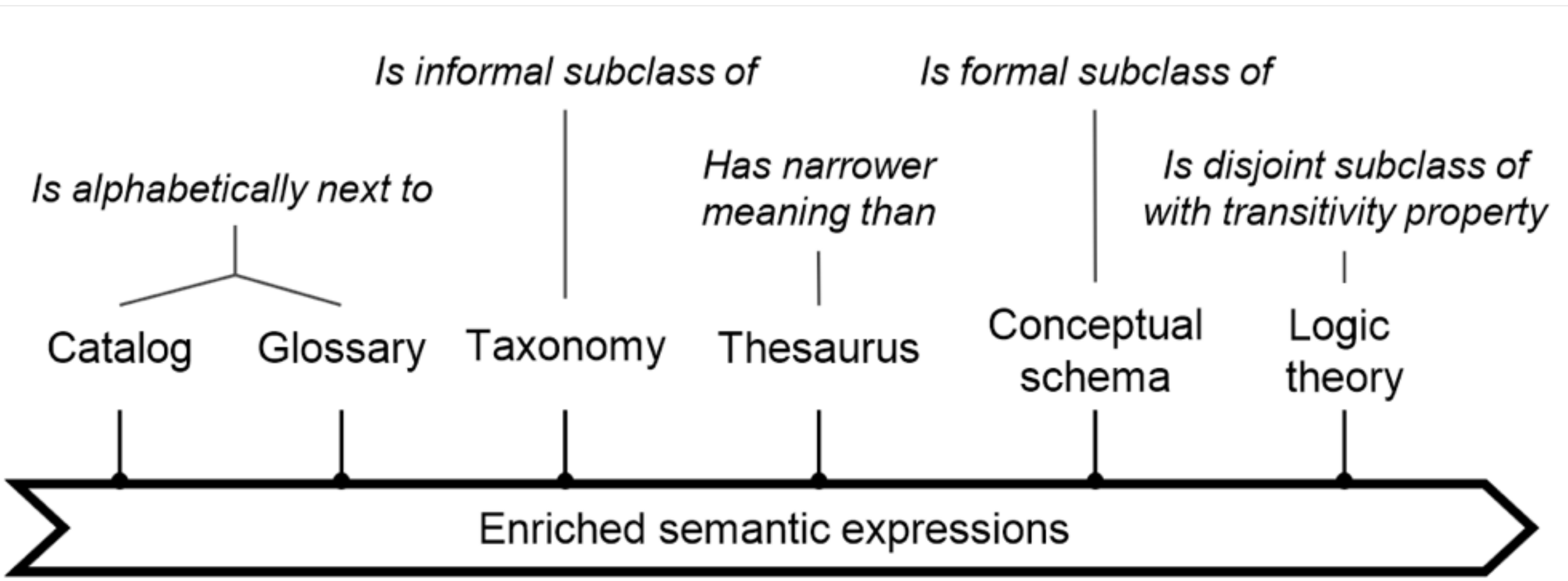
Ontologie-basierte Informationsintegration

- Eine Ontologie **reduziert die Anzahl der Mappings** von Symbolen zu Dingen der realen Welt

Abbildung von Symbolen auf Dinge der Welt

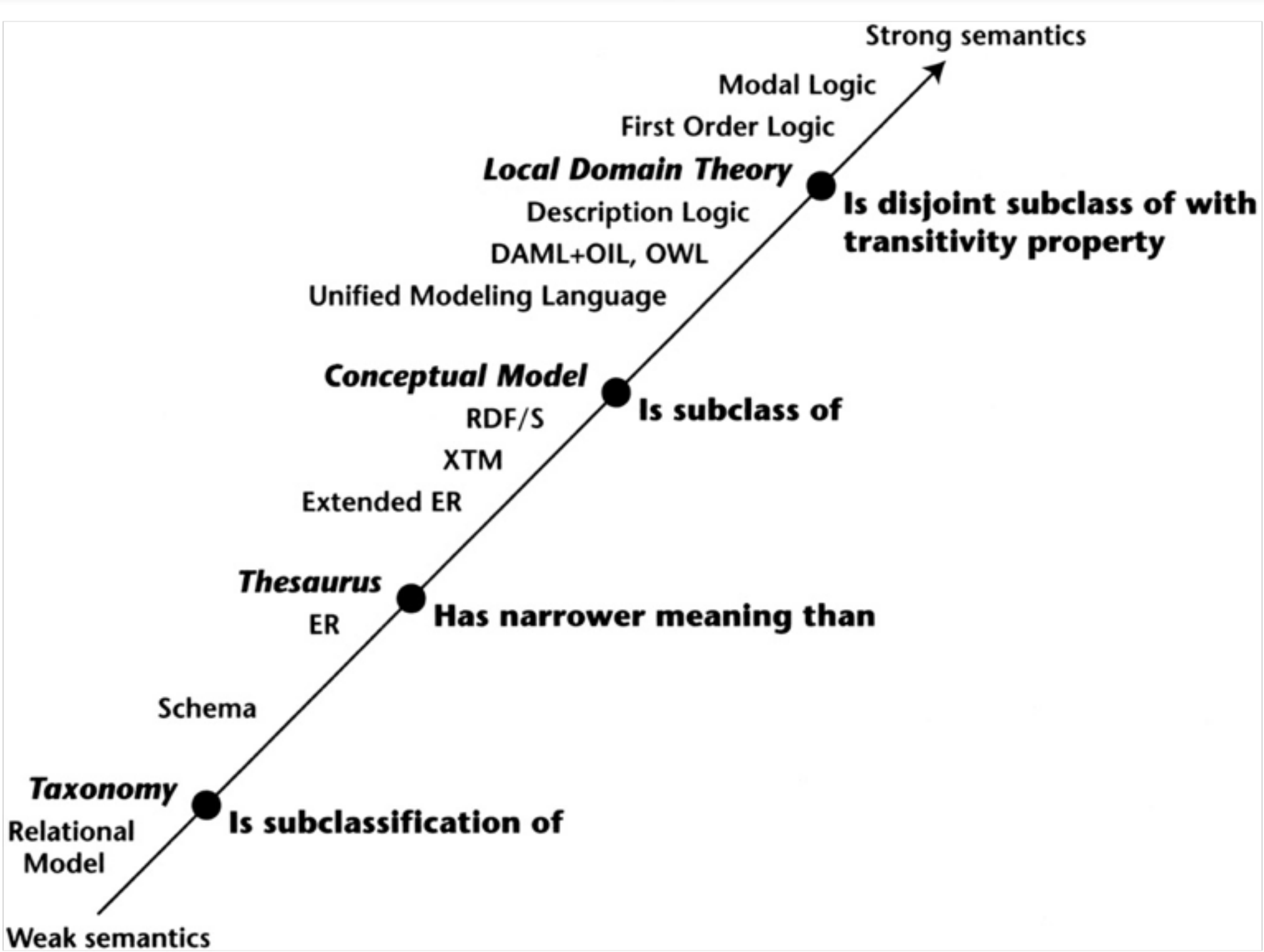


Ontology Spectrum (I)



Quelle: http://tw.rpi.edu/weblog/2013/12/19/what-is-ontology/

Ontology Spectrum – inkl. Datenmodellen



Quelle: http://edc.tversu.ru/elib/inf/0040/images/fig7-5_0.jpg

Fazit

- Menschen und/oder Software-Agenten können das **zur Integration von Informationen notwendige Wissen** nicht teilen, wenn sie keine **gemeinsame Sprache** sprechen

- Der **Rahmen** für dieses **gemeinsame Verständnis** wird durch **Ontologien** gebildet

Overview of the Lecture

- Grundlegende Begriffe und Konzepte
  - Integriertes Informationssystem
  - Heterogenität
- **Ontologien**
  - Definitionen
  - Eigenschaften
  - **Formale Sprachen für Ontologien**
- Ontologie-Sprachen
  - RDF
  - RDF/S
  - OWL
- Beispiel – Semantische Integration
  - Semantisches Mapping von Wissensbasen

Formale Sprachen für Ontologien

- **Definitionen**:
  - Eine **Ontologie** ist eine Menge von **Axiomen**.
  - Ein Axiom beschreibt die formale Semantik der **Konzepte**.
  - **Konzepte** werden durch ihre Symbole bezeichnet.
  - Die **Ontologie-Sprache** definiert die Ontologie-Sprachelemente, die in einem Axiom verwendet werden können.

- **Mögliche Formalismen**:
  - Terminological Logic
    (vgl. Vorlesung “Angewandte Informatik 1”)
  - Frame Logic (F-Logic)
    (vgl. Vorlesung “Semantic Web Technologies I”)
  - **RDF Schema (RDFS)**
    (vgl. Vorlesung “Semantic Web Technologies I”)
  - **OWL**
    (vgl. Vorlesung “Semantic Web Technologies I”)

- **Bemerkung:**
  In dieser Vorlesung werden nur **Ontologie-Sprachen** aus **Semantic Web Technologien** diskutiert. Es gibt auch andere.

Anforderungen an Ontologie-Sprachen

- wohldefinierte **Syntax**
- formale **Semantik**
- adäquate **Ausdrucksmittel**
  - Ausdrucksmittel für eine konzeptuelle Modellierung
    - Klassen, Beziehungen, Vererbung etc.
  - Hinreichende Ausdrucksmächtigkeit
- effiziente **Verarbeitung**
  - Durchsetzung der formalen Semantik
  - Zugriffsmethoden
- Unterstützung **offener** (**Web**-)**Umgebungen**
  - Verteilte Ontologien
  - Autonomie und ihre Konsequenzen

Semantic Web Ontologie-Sprachen RDF, RDFS und OWL

Overview of the Lecture

- Grundlegende Begriffe und Konzepte
  - Integriertes Informationssystem
  - Heterogenität
- Ontologien
  - Definitionen
  - Eigenschaften
  - Formale Sprachen für Ontologien
- **Ontologie-Sprachen**
  - **RDF**
  - RDF/S
  - OWL
- Beispiel – Semantische Integration
  - Semantisches Mapping von Wissensbasen

Why XML is not a feasible technology for expressing rich data relationships...

- **Ambiguity** of XML Tags (*can be mitigated by namespaces*)
- Tree structure not optimal for...
  - representing **explicit relationships**
  - an **intuitive description** of data and their relationships
  - **1:n schema mappings** between an XML tree and equivalent directed graphs
- But:
  - good **tool support** and established technology
  - intuitive schema for **classification**

How to decode the following fact in XML?

- How to represent the following statement in XML:
  "Das Buch 'Semantic Web - Grundlagen' wird beim Springer-Verlag verlegt"

```
<Verlegt>
 <Verlag>Springer-Verlag</Verlag>
 <Buch>Semantic Web - Grundlagen</Buch>
</Verlegt>
```

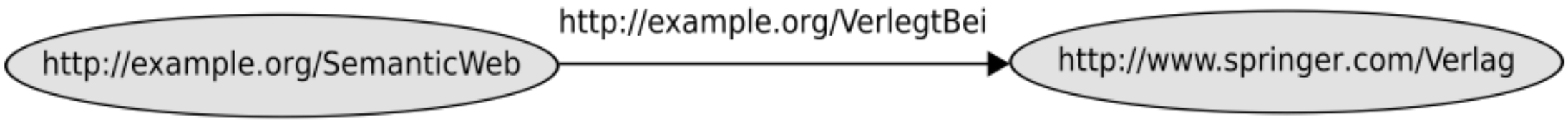
Modeling Ambiguity

```
<Verlag Name="Springer-Verlag">
 <Verlegt Buch="Semantic Web - Grundlagen" />
</Verlag>
```

```
<Buch Name="Semantic Web - Grundlagen">
 <Verleger Verlag="Springer-Verlag" />
</Buch>
```

Solution: Use Graphs for decoding Information and their Relationships

- The **logical model** behind a graph is intuitive and easy to comprehend

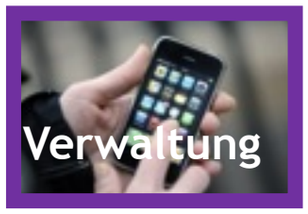


- The **one-to-many schema mapping problem** of trees and graphs can be circumvented
- Accurate representation of the **universe of discourse**
- **Relationships** are explicitly represented and typed

Basic Idea: Everything is a Resource and can be identified by an URI/IRI…

http://www.karlsruhe.info/
resources/districts#Hagsfeld

http://de.dbpedia.org/resource/
Karlsruhe-Hagsfeld



Das
selbstgestaltete
Stadtviertel

Das Null-Energie-
Stadtviertel

Das emissionsarme
Stadtviertel

http://www.die_mustermanns.de/
familienmitglieder#Tobias

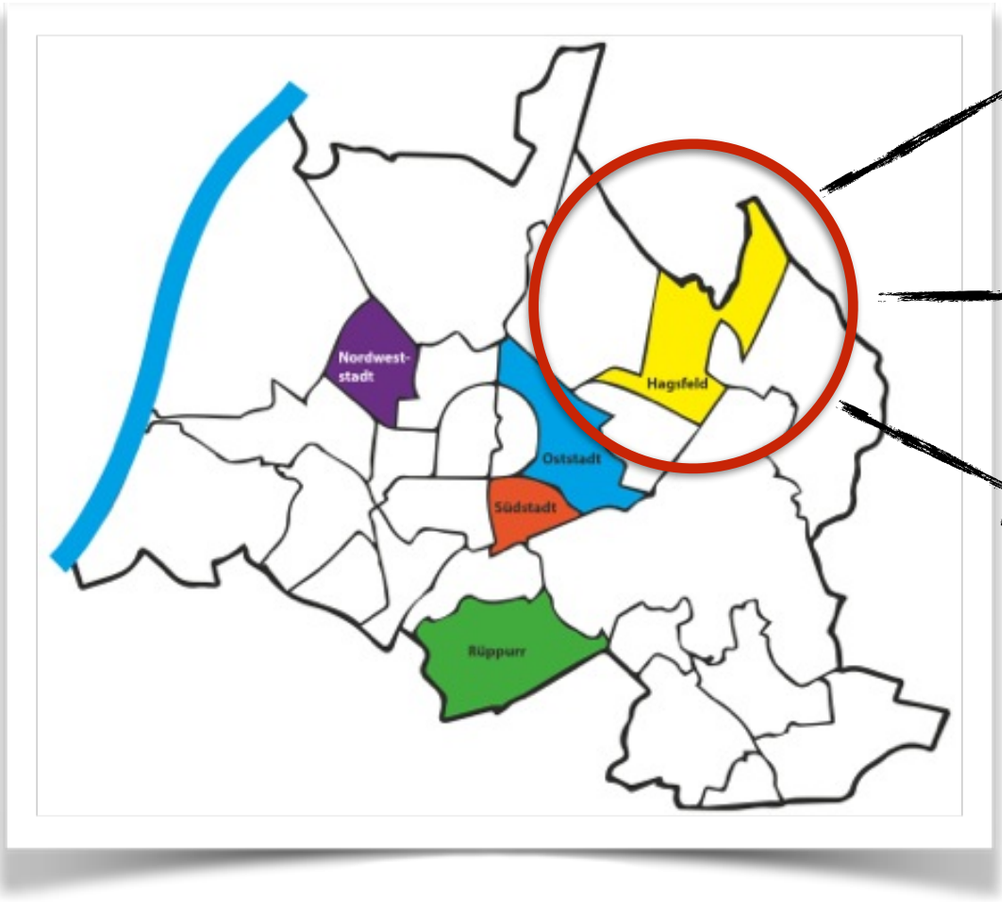
Das bürgerzentrierte
Stadtviertel

Das Stadtviertel der
kurzen Wege

http://www.die_mustermanns.de/
familienmitglieder#Opa_Herrman

http://www.dienstverzeichnis.info/
anbieter/xyz

Dereferencing HTTP URIs



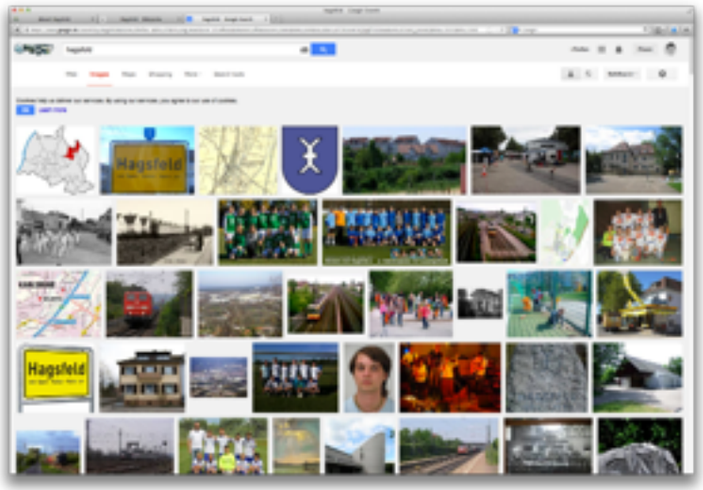
http://de.dbpedia.org/resource/
Karlsruhe-Hagsfeld

http://www.karlsruhe.info/
stadtteile/Hagsfeld

http://de.wikipedia.org/wiki/
Hagsfeld

Identifier

Representations



Structured Representation of an Information or Non-information Resource

de.dbpedia.org/page/Hagsfeld

dbpedia hagsfeld karlsruhe

About: Hagsfeld

An Entity of Type : PopulatedPlace, from Named Graph : http://de.dbpedia.org, within Data Space : de.dbpedia.org

DBpedia

Hagsfeld ist ein Stadtteil von Karlsruhe in Baden-Württemberg und grenzt an den Stadtteil Waldstadt an.

Property	Value
dbpedia-owl:PopulatedPlace/area	▪ 7176.0
dbpedia-owl:abstract	▪ Hagsfeld ist ein Stadtteil von Karlsruhe in Baden-Württemberg und grenzt an den Stadtteil Waldstadt an.
dbpedia-owl:area	▪ 7176000000.000000 (xsd:double)
dbpedia-owl:areaCode	▪ 0721
dbpedia-owl:censusYear	▪ 2010-01-01 00:00:00 (xsd:date)
dbpedia-owl:city	▪ dbpedia-de:Karlsruhe
dbpedia-owl:cityType	▪ Stadt
dbpedia-owl:elevation	▪ 114.000000 (xsd:double)
dbpedia-owl:populationAsOf	▪ 2010-02-20 (xsd:date)
dbpedia-owl:populationUrban	▪ 7023 (xsd:integer)
dbpedia-owl:postalCode	▪ 76139
dbpedia-owl:state	▪ dbpedia-de:Baden-Württemberg
dbpedia-owl:thumbnail	▪ http://upload.wikimedia.org/wikipedia/commons/thumb/c/c5/Wappen_Hagsfeld.svg/200px-Wappen_Hagsfeld.svg.png
dbpedia-owl:wikiPageExternalLink	▪ http://www1.karlsruhe.de/Stadtteile/BK-Hagsfeld/
dbpedia-owl:wikiPageID	▪ 1097844 (xsd:integer)
dbpedia-owl:wikiPageInterLanguageLink	▪ http://nl.dbpedia.org/resource/Hagsfeld ▪ http://ru.dbpedia.org/resource/Харсфельд
dbpedia-owl:wikiPageRevisionID	▪ 102406695 (xsd:integer)
dbpprop-de:breitengrad	▪ 49 (xsd:integer)
dbpprop-de:bundesland	▪ Baden-Württemberg
dbpprop-de:eingemeindungsdatum	▪ 1938 (xsd:integer)
dbpprop-de:einwohner	▪ 7023 (xsd:integer)
dbpprop-de:einwohnerStandDatum	▪ 2010 (xsd:integer)
dbpprop-de:fläche	▪ 71762 (xsd:integer)
dbpprop-de:gemeindeart	▪ Stadt
dbpprop-de:gemeindename	▪ Karlsruhe
dbpprop-de:höhe	▪ 114 (xsd:integer)

Resource Description Framework

- **R**esource
  - can be or refer to everything that is **uniquely identifiable** by an URI and must be **referencable**
- **D**escription
  - representation of **properties** and **relationships** among resources in form of **directed graphs**
- **F**ramework
  - conflation of **Web technologies**, **protocols**, and **standards** (URI, HTTP, XML) and **formal logics** (semantics)

Resource Description Framework

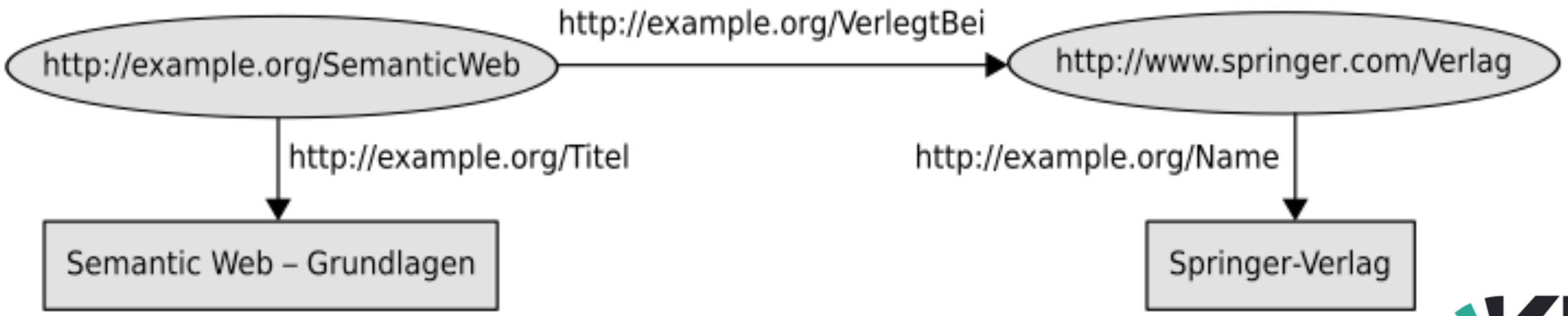
- Facts in RDF are expressed as **statements** in the form of **<subject> <predicate> <object>**-triples

- Statement (i.e., RDF-triple):
  - Subject i.e. *Resource* ---> **URI**
  - Predicate i.e. *Property* ---> **URI**
  - Object i.e. *Object / Value* ---> **URI / Literal**

- All RDF statements follow the same **logical schema**

Representation of Data in RDF

- Intuitive representation as a **list of triples** (i.e. *RDF statements*)
- An RDF document can be **represented graphically** in different forms
- Common representation form is a **Node-Edge-Node** graph
- Due to the **uniqueness** of node and edge identifiers, an RDF graph can be **reconstructed** from the list of triples



Elements of RDF Graphs

- **URIs**
  - enable the unique identification of resources
- **Literals**
  - describe data values that do not have a specific existence
- **Blank Nodes**
  - enable statements about the existence of individuals and their properties without naming them explicitly

Uniform Resource Identifier

- A **Uniform Resource Identifier** (URI) defines a simple and extensible schema for worldwide unique identification of abstract or physical resources (RFC 3986).

- Resources can identify **any distinguishable object** with a clear identity

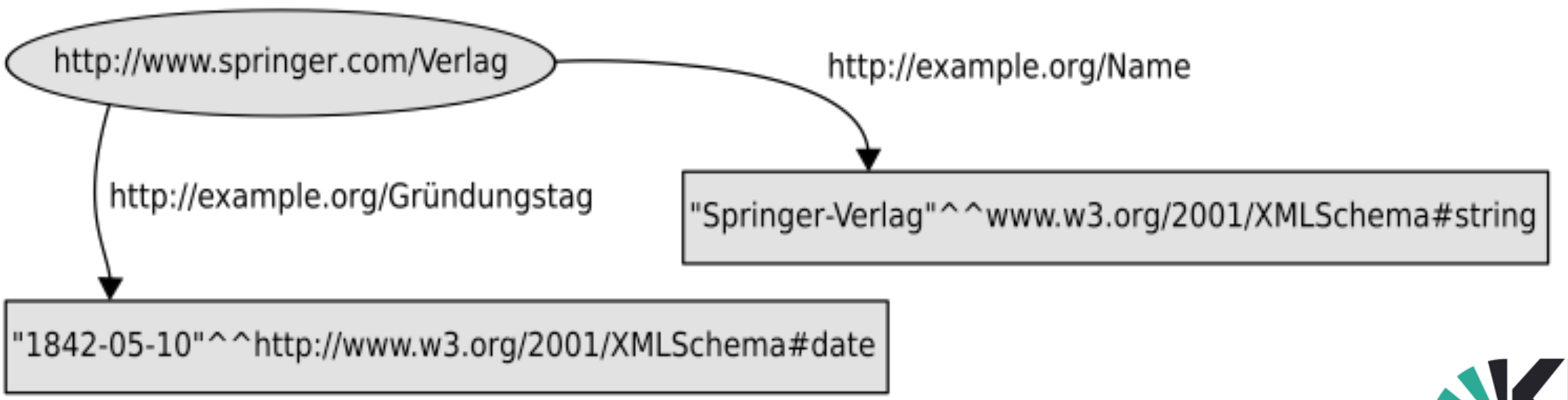
- In the semantic Web, we distinguish between **two types** of resources:
  - **Information Resources**
  - **Non-information Resources**

Literals*

- Used for the representation of **data values**
- Representation as **strings**
- Interpretation depending on the **data type** associated with a Literal
- Literals without type information are **untyped** and treated as **plain strings**
- Represented as **boxes** in visualized RDF graphs

Literals

- Used for the representation of **data values**
- Interpretation depends on the **data type** associated with a Literal
  - Common practice to express typed literals via **XML Schema data types**
- Literals without type information are **untyped** and treated as **plain strings**
- Represented as **boxes** in visualized RDF graphs



Blank Nodes

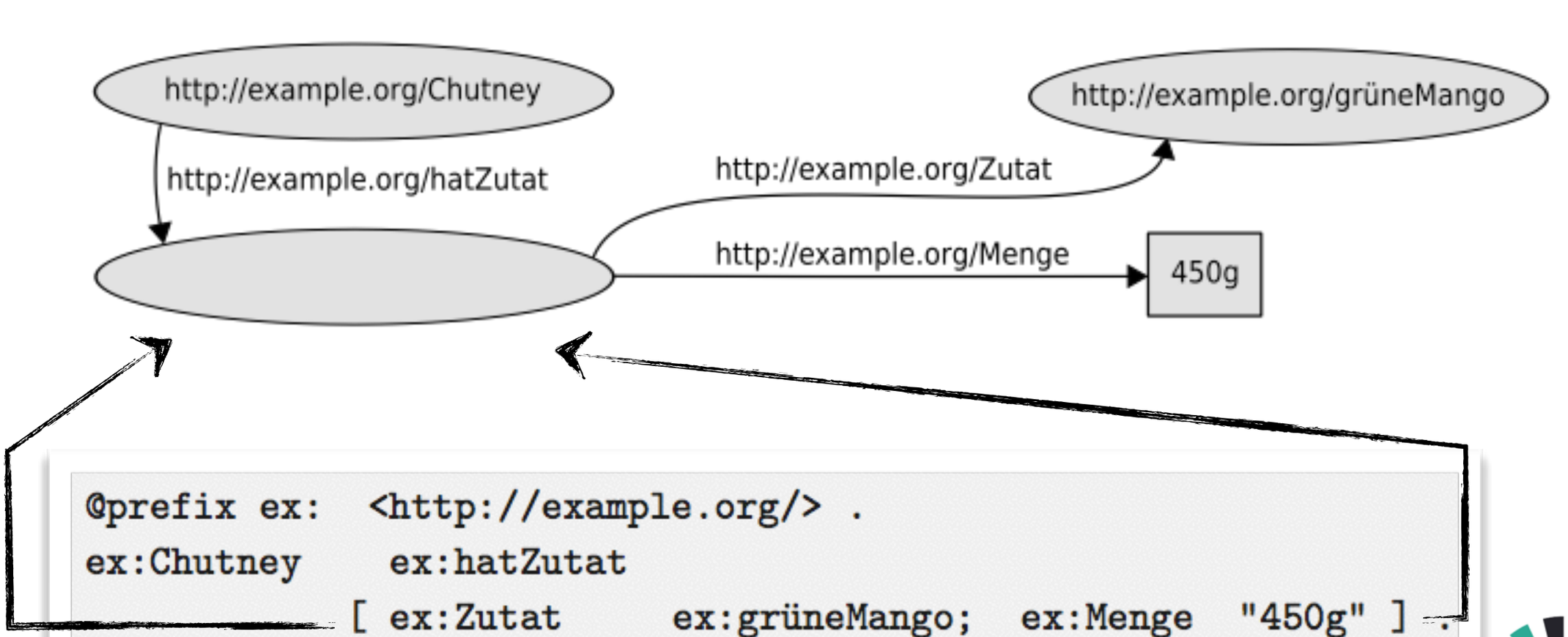
- Indicate the existence of an individual with specific attributes, but <u>without</u> providing external **identification** or **URI-based reference** information.

- Used to model **multi-valued** relations (e.g. `rdf:value`)
- Used for **auxiliary resources** that do not need a name
- Example:
  - *“A lecture takes place twice a week in two different rooms”*

Example: Cooking with RDF

- Consider the following recipe:

  “Für die Zubereitung von Chutney benötigt man 450g grüne Mango, einen Teelöffel Cayennepfeffer, …"




```
@prefix ex:   <http://example.org/> .
ex:Chutney     ex:hatZutat
               [ ex:Zutat     ex:grüneMango;   ex:Menge   "450g" ] .
```

RDF/XML Serialization Syntax

- Use of **namespaces** to disambiguate tag names
- RDF elements are denoted by namespace **<rdf:...>**

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

  <rdf:Description rdf:about="http://example.org/SemanticWeb">
    <ex:VerlegtBei>
      <rdf:Description rdf:about="http://springer.com/Verlag">
      </rdf:Description>
    </ex:VerlegtBei>
  </rdf:Description>

</rdf:RDF>
```

RDF/XML Syntax

- The `rdf:Description` element indicated the **subject** of a statement (the URI of which is specified by the `rdf:about` attribute)
- Every **embedded element** indicates a **predicate** (the URI of which denotes the element name) and contains the **object** of a statement (represented by the `rdf:Description` element).

```
<rdf:Description rdf:about="http://example.org/SemanticWeb">
  <ex:VerlegtBei>
    <rdf:Description rdf:about="http://springer.com/Verlag">
    </rdf:Description>
  </ex:VerlegtBei>
</rdf:Description>
```

RDF/XML Syntax Abbreviations

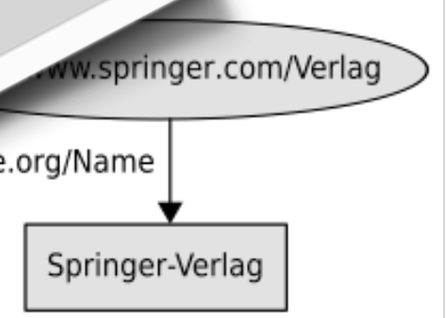
- Representations of **Literals** as XML attributes
- **Property URIs** are represented by attribute names
- **Object URIs** are values of rdf:resource attribute within a property tag


```
<rdf:Description rdf:about="http://example.org/SemanticWeb"
                 ex:Titel= "Semantic Web - Grundlagen">
  <ex:VerlegtBei rdf:resource="http://springer.com/Verlag" />
</rdf:Description>
<rdf:Description rdf:about="http://springer.com/Verlag"
                 ex:Name="Springer-Verlag"                  />
```

RDF/XML Syntax Abbreviations

- Representations of **Literals** as XML attribut
- **Property URIs** are represented b
- **Object URIs** are value erty tag

Bear complex applications with the XML syntax in mind!




```
<rdf:Descri                                    emanticWeb"
                                               Grundlagen">
  <ex:Verlegtb                                 pringer.com/Verlag" />
</rdf:Descripti
<rdf:Description                               ttp://springer.com/Verlag"
                                 e="Springer-Verlag"                    />
```

Other Serialization Formats

- **N3 Notation**
  - Powerful language with features for expressing variables, rules, etc.
  - Direct listing of triples; better readability than XML
  - N-Triples: part of N3

- **Turtle** (Terse RDF Triple Language)
  - Extension of N3 for RDF
  - URIS in < >
  - Literals in “...”
  - Triples end with .
  - Ignorance of whitespaces

```
<http://example.org/SemanticWeb>
    <http://example.org/VerlegtBei>  <http://springer.com/Verlag> .
<http://example.org/SemanticWeb>
    <http://example.org/Titel>       "Semantic Web - Grundlagen" .
<http://springer.com/Verlag>
    <http://example.org/Name>        "Springer-Verlag" .
```

Turtle-Syntax

- from...

```
@prefix ex: <http://example.org/> .
@prefix springer: <http://springer.com/> .

ex:SemanticWeb    ex:VerlegtBei   springer:Verlag .
ex:SemanticWeb    ex:Titel        "Semantic Web - Grundlagen" .
springer:Verlag   ex:Name         "Springer-Verlag" .
```

- to (by using abbreviations)...

```
@prefix ex: <http://example.org/> .

ex:SemanticWeb ex:Autor ex:Hitzler, ex:Krötzsch, ex:Rudolph, ex:Sure ;
               ex:Titel "Semantic Web - Grundlagen" .
```

Reification

- Question:

  How do we model propositions about propositions?

- This is problematic in RDF
- In the german language, such circumstance is often indicated by the word "*dass*"
  - Example:
    - *"Der Detektiv vermutet, dass der Butler den Gärtner ermordet hat."*

How to model the following fact?

- **Example**:

  The detective supposes that the butler killed the gardener

- **Solution 1**:

  ex:detective ex:supposes “The butler killed the gardener.” .

- **Shortcomings**:
  - Literal can not be referenced in other triples
  - Inherent meaning (semantics) of statement is lost

How to model the following fact?

- **Example**:

  The detective supposes that the butler killed the gardener

- **Solution 2**:

  ex:detective ex:supposes ex:theButlerKilledTheGardener .

- **Shortcomings**:
  - Does not capture the full meaning of the proposition
  - Semantics and inner structure of the assertion is lost

Reification

```
ex:butler ex:killed ex:gardener .
```

- Solution: **nested triples**
  - Object of the previous triple is a triple of its own
  - Draws the idea from **many-valued relations** (blank nodes)

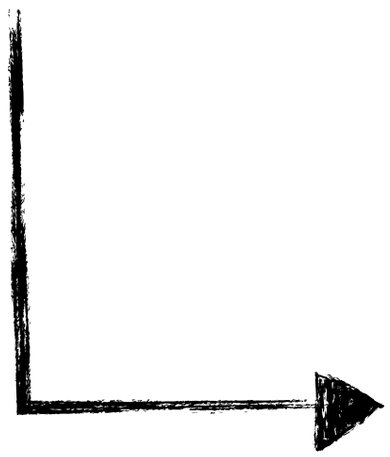
- **Auxiliary node** is used to refer to the whole statement
  - Access to inner structure of represented triple is connected via a **blank node** with
    - **rdf:subject**, referring to a statement's subject
    - **rdf:predicate**, referring to a statement's predicate
    - **rdf:object**, referring to the object properties
  - Corresponding triple is called **reified**

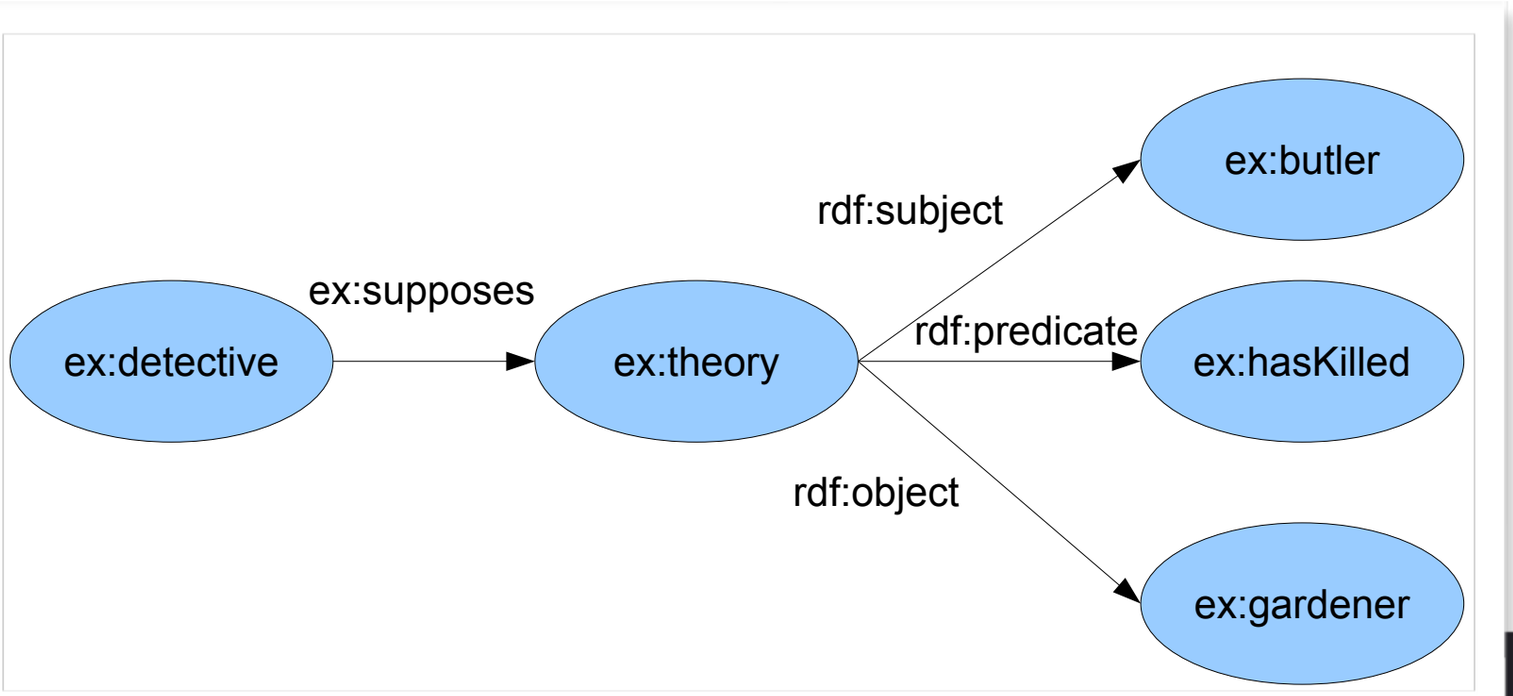
Reification

- **Correct model:**

```
ex:theory       rdf:subject      ex:butler .
ex:theory       rdf:predicate    ex:hasKilled .
ex:theory       rdf:object       ex:gardener .

ex:detective    ex:supposes      ex:theory .
```





Summary

- RDF allows for expressing **simple facts**
  - *"Anne is the mother or Merula"*
- It is desirable to express more **generic knowledge**
  - *"Mothers are female"*
  - *"If somebody has a daughter, this person is a parent"*
- Such kind of knowledge is called **schema knowledge** or **terminological knowledge**
- **RDF Schema** allows for schema knowledge modeling (although with less expressivity than OWL)

Overview of the Lecture

- Grundlegende Begriffe und Konzepte
  - Integriertes Informationssystem
  - Heterogenität
- Ontologien
  - Definitionen
  - Eigenschaften
  - Formale Sprachen für Ontologien
- **Ontologie-Sprachen**
  - RDF
  - **RDF/S**
  - OWL
- Beispiel – Semantische Integration
  - Semantisches Mapping von Wissensbasen

Motivation

- RDF allows for making arbitrary assertions about **individual resources** and their **relationships** on the Web

- Preferable:
  - To make assertions about **generic sets** of **individuals** (ie. *classes*) e.g. the class of all authors, organizations, books etc.
  - To explicitly specify the **logical relations** between classes, individuals, and properties to accurately describe the **universe of discourse**
    - e.g. publisher are organizations / authors are persons / etc.

- RDF Schema (RDFS) allows for the explicit specification of **schematic** and **terminological knowledge** (aka *factual knowledge*) about resources.

RDF Schema

- RDFS simply defines a **data model** and a **vocabulary** for the creation of RDF statements
- Official name: **“RDF Vocabulary Description Language”**

- RDF Schema allows:
  - Definition of **classes**
  - Definition of **properties** and **restrictions**
  - Definition of **hierarchies**
    - Subclasses and superclasses
    - Subproperties and superproperties

RDF Schema

- Part of the **W3C RDF Recommendation**
- RDFS is a specific RDF vocabulary, i.e., **every RDFS document** is also a **valid** RDF document
- Meta vocabulary: allows for the **specification of the semantics** of arbitrary RDF vocabularies (or parts of it)
- Every **software** with **RDFS capabilities** is able to comprehend the incorporated RDFS semantics correctly
- RDFS allows for defining **lightweight ontologies**

- Namespace: http://www.w3.org/2000/01/rdf-schema#

Classes and Instances

- Resources can be marked as **instances** of a class using the `rdf:type` property:

  `ex:semanticWeb  rdf:type  ex:Lehrbuch .`

  Assigns the object's URI to the subject as a new **type**, i.e., the resource `ex:semanticWeb` is a (new) instance of the class `ex:Lehrbuch`

- Class association or class assignment is <u>not</u> **exclusive**, i.e., a resource can be instance of *many* classes:

  `ex:semanticWeb  rdf:type  ex:Unterhaltsam .`

- **Problem**:
  - Syntactically **no inherent differentiation** between designators for classes and individuals, i.e., there is **no syntactic way** in RDFS to distinguish URIs representing *individuals* from URIs representing *class names*

Classes and Instances

- Resources can be marked as **instances** of a class using the `rdf:type` property:

  `ex:semanticWeb  rdf:type  ex:Lehrbuch .`

  Assigns the object's URI to the subject as a new **type**, i.e., the resource `ex:semanticWeb` is a (new) instance of the class `ex:Lehrbuch`

- Class association or class assignment is not **e** resource can be instance of *many* classes:

  `ex:semanticWeb  rdf:type  ex:`

- **Problem**:
  - Syntactically **no inherent differentiation** between design individuals, i.e., there is **no syntactic way** in RDFS to distinguish representing *individuals* from URIs representing *class names*

- **Solution**:
  Explicit designation of an URI as belonging to the class of all classes using `rdfs:Class`

Classes and Instances

- Preferable: **unique designation** of an URI as class
  - Designating (**typifying**) an URI as a class with **rdfs:Class**
  - `ex:Lehrbuch rdf:type rdfs:Class .`
    `ex:semanticWeb rdf:type ex:Lehrbuch .`

- `rdfs:Class` is the class of all classes and hence also **contains itself**, i.e., the following statement is valid:
  - `rdfs:Class rdf:type rdfs:Class .`

- **Notational conventions** for classes, properties, and individuals:
  - URIs representing classes are *capitalized*
  - Instance names and properties are written in *lower case*

Classes and Subcla

```
ex:Lehrbuch      rdf:type    rdfs:Class .
ex:semanticWeb   rdf:type    ex:Lehrbuch .
```

- **Problem**:
  - Search for **ex:Buch** only returns such publications that are this type, i.e, no instances of class **ex:Lehrbuch** are retrieved
  - Asserting **ex:semanticWeb rdf:type ex:Book** allows only for adding one specific resource
  - This explicit inclusion of all instances of a class is **cumbersome** and leads to **large RDF documents**

- **Solution**:
  - Asserting that every **ex:Lehrbuch** is also an **ex:Buch**, i.e., every instance of **ex:Lehrbuch** is also an instance of **ex:Buch**

```
ex:Lehrbuch  rdfs:subClassOf  ex:Buch .
```

Classes and Subclasses

- rdfs:subClassOf is **reflexive**, i.e., every class is subclass of its own;
  - Example:

    ```
    ex:Lehrbuch  rdfs:subClassOf  ex:Lehrbuch .
    ```

    is a valid statement.

- **Equality** of two classes can be expressed via mutual subclass relationships:
  - From

    ```
    ex:Hospital      rdfs:subClassOf  ex:Krankenhaus .
    ex:Krankenhaus   rdfs:subClassOf  ex:Hospital .
    ```

    follows that every instance of ex:Hospital is also an instance of ex:Krankenhaus and vice versa.

Class Hierarchies

- rdfs:subClassOf allows for defining complex **class hierarchies** (so-called *Taxonomies*):
  - ex:Lehrbuch rdfs:subClassOf ex:Buch .
    ex:Buch rdfs:subClassOf ex:Printmedium .
    ex:Zeitschrift rdfs:subClassOf ex:Printmedium .

- rdfs:subClassOf is **transitive**, i.e., it allows for the **propagation** of subclass relationships
  - From the assertions above, the following statement can be deduced:

    ex:Lehrbuch rdfs:subClassOf ex:Printmedium .

Class Hierarchies

- rdfs:subClassOf a… …mplex **class hierarchie**

  - ex:Lehrbuch rdfs:subCla… …
    ex:Buch rdfs:subClassOf …
    ex:Zeitschrift rdfs:subClassOf ex:Pri…

- Relation to **set theory**:

  rdf:type refers to ∈
  rdf:subClassof refers to ⊆

- rdfs:subClassOf is **transitive**, i.e., it allows for the **propagation** of subclass relationships

  - From the assertions above, the following statement can be deduced:

    ex:Lehrbuch rdfs:subClassOf ex:Printmedium .

Predefined Classes

- **rdfs:Class**
  - defines an abstract object and is applied (with `rdf:type`) to create instances
- **rdfs:Resource**
  - Class of all resources (every entity of an RDF model is instance of this class)
- **rdf:Property**
  - Class of all relationships between resources
- **rdfs:Literal / rdf:XMLLiteral**
  - Class for literals / Class of all values pertaining to the predefined data type XMLLiteral
- **rdfs:Datatype**
  - Class of all data types, i.e., just as `rdfs:Class`, it is a class for classes
- **Other classes**
  - rdf:List, rdf:Seq, rdf:Bag, rdf:Alt, rdfs:Container, rdfs:ContainerMembershipProperty, rdf:Statement

Properties

- Properties in RDF(S) are treated as **first-class citizens**
- Characterize the **relationship** between two resources
- Properties are **defined independently** from concrete or specific classes (as opposed to OOP)
- Syntactical rule:
  - properties start with a **lower-case letter**, e.g., `rdf:type`, `ex:authorOf`, `rdfs:subClassOf` etc.

Property Hierarchies

- Properties may be **structured hierarchically** (cf. classes)
- `rdfs:subPropertyOf` is an instance of `rdf:Property` and states that all resources related by one property are also related by another
- `rdfs:subPropertyOf` is **transitive**

```
ex:glücklichVerheiratetMit   rdf:subPropertyOf   rdf:verheiratetMit .
ex:markus   ex:glücklichVerheiratetMit   ex:anja .
```

```
ex:markus  ex:verheiratetMit  ex:anja .
```

Properties in RDF Schema

- **rdfs:subClassOf**
  - transitive property to define inheritance hierarchies for classes
- **rdfs:subPropertyOf**
  - transitive property to define inheritance hierarchies for properties
- **rdfs:domain**
  - defines the domain of a property concerning a class
- **rdfs:range**
  - defines range of a property concerning a class

Restrictions on Properties

- Restrictions allow us to state that a certain property can only be between things of a certain type
  - Example: *"When individual A is married to individual B, then both A and B are instances of class Person."*

  - **rdfs:domain**
    - Any resource that has a given property is an instance of one or more classes

  - **rdfs:range**
    - Values of a property are instances of one or more classes.

rdfs:domain and rdfs:range

- **P rdfs:domain C**

  States that P is an instance of the class rdf:Property, that C is a instance of the class rdfs:Class and that the resources denoted by the subjects of triples whose predicate is P are instances of the class C.

- **P rdfs:range C**

  States that P is an instance of the class rdf:Property, that C is an instance of the class rdfs:Class and that the resources denoted by the objects of triples whose predicate is P are instances of the class C.

- **Example**:
  - *All individuals that are married are persons*

    ex:isMarriedTo rdfs:domain ex:Person .
    ex:isMarriedTo rdfs:range ex:Person .

  - *By stating that…*

    ex:Tom ex:isMarriedTo ex:Jane .

  - *…we can infer the following:*

    ex:Tom rdf:type ex:Person .
    ex:Jane rdf:type ex:Person .

  - *Property restrictions also apply for data types:*

    ex:hasAge rdfs:range xsd:nonNegativeInteger .

- **Example 1**:

  ex:authorOf rdfs:range ex:Textbook .
  ex:authorOf rdfs:range ex:Storybook .

  States that everything in the rdfs:range of ex:author is <u>both</u> a ex:Textbook and a ex:Storybook

- **Example 2**:

  ex:isMarriedTo rdfs:domain ex:Person .
  ex:isMarriedTo rdfs:range ex:Person .
  ex:instituteAIFB rdf:type ex:Institution .

  ex:pascal ex:isMarriedTo ex:instituteAIFB .

  A logical consequence of this is:

  ex:instituteAIFB rdf:type ex:Person .

Further Properties

- **rdfs:seeAlso**
  - defines a relation of a resource to another, which explains it
- **rdfs:isDefinedBy**
  - subproperty of rdf:seeAlso, defines the relation of a resource to its definition
- **rdfs:comment**
  - comment, usually in the form of a text
- **rdfs:label**
  - Human-readable name of a resource

An example Ontology

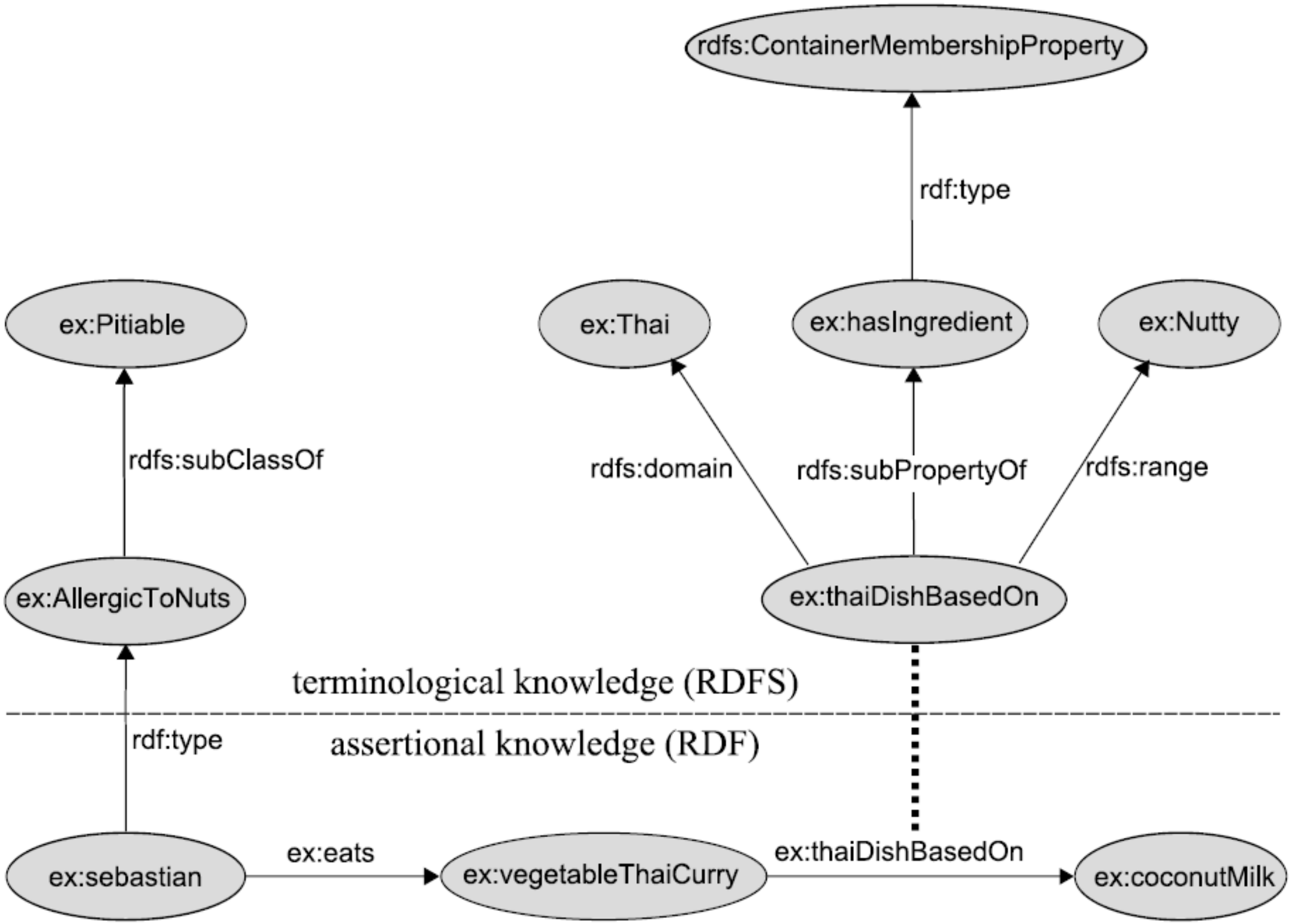
Assertional Knowledge

```
ex:vegetableThaiCurry   ex:thaiDishBasedOn     ex:coconutMilk .
ex:sebastian            rdf:type               ex:AllergicToNuts .
ex:sebastian            ex:eats                ex:vegetableThaiCurry .

ex:AllergicToNuts       rdfs:subClassOf        ex:Pitiable .
ex:thaiDishBasedOn      rdfs:domain            ex:Thai .
ex:thaiDishBasedOn      rdfs:range             ex:Nutty .
ex:thaiDishBasedOn      rdfs:subPropertyOf     ex:hasIngredient .
ex:hasIngredient        rdf:type    rdfs:ContainerMembershipProperty.
```

Terminological Knowledge

The Ontology represented as Graph



Overview of the Lecture

- Grundlegende Begriffe und Konzepte
  - Integriertes Informationssystem
  - Heterogenität
- Ontologien
  - Definitionen
  - Eigenschaften
  - Formale Sprachen für Ontologien
- **Ontologie-Sprachen**
  - RDF
  - RDF/S
  - **OWL**
- Beispiel – Semantische Integration
  - Semantisches Mapping von Wissensbasen

OWL – The Web Ontology Language

OWL – The Web Ontology Language

- Web Ontology Language
  - W3C Recommendation for the Semantic Web, 2004
  - OWL 2 (revised W3C Recommendation), 2009
- Semantic Web Wissensrepräsentationssprache basierend auf Beschreibungslogiken (description logics, kurz: DLs)
  - OWL DL entspricht der Beschreibungslogik SROIQ(D)
  - gemacht für das Semantic Web: Verwendung von URIs
  - es gibt Syntaxen, die auf XML und RDF basieren
- OWL – Grundlegende Prinzipien
  - Open World Assumption
  - rein deklarative Semantik
  - keine Unique Name Assumption
  - Integration mit RDFS

- Namespace: http://www.w3.org/2002/07/owl#

Elemente der Semantic Web Ontology Language – OWL (I)

- **Equivalent classes**: C equivalentClass D.
  - Intuitive Semantik: C und D bezeichnen die gleiche Menge von Dingen.
  - Z.B.: Woman equivalentClass Female
- **Equivalent properties**: R equivalentProperty S.
  - Intuitive Semantik: R und S sind die gleiche Relation.
  - Z.B.: hasParent equivalentProperty childOf
- **Equivalent individuals**: A sameAs B.
  - Intuitive Semantik: Zwei Dinge sind gleich.
  - Z.B.: Eve sameAs Eva
- **Different individuals**: A differentFrom B.
  - Intuitive Semantik: Zwei Dinge sind *nicht* gleich.
  - Z.B.: Eve differentFrom Adam
  - Hinweis: Das muss explizit angegeben werden wegen der **Non-Unique Name Assumption**

Elemente der Semantic Web Ontology Language – OWL (II)

- **Union**: C equivalentClass (unionOf D E).
  - Intuitive Semantik: C ist die Vereinigung von D und E.
  - Z.B.: Person equivalentClass (unionOf Man Woman)

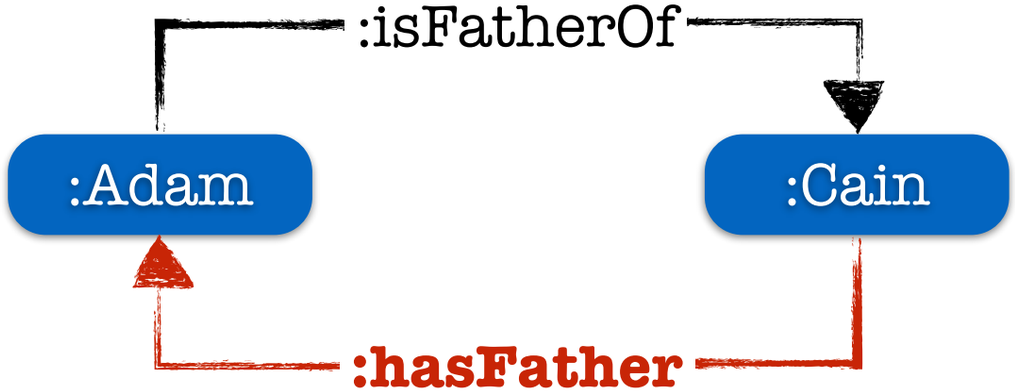
- **Intersection**: C equivalentClass (intersectionOf D E).
  - Intuitive Semantik: Wenn etwas sowohl D und E ist, dann ist es C umgekehrt !).
  - Z.B.: nonFlyingBird equivalentClass (intersectionOf Bird nonFlyingAnimal)

- **Negation**: C equivalentClass (complementOf D).
  - Intuitive Semantik: Wenn etwas nicht D ist, dann ist es C.
  - Hinweis: Man equivalentClass (complementOf Woman) ist falsch, da nicht alles, was nicht Frau ist, ein Mann ist.
    Man subClassOf (complementOf Woman) ist richtig.

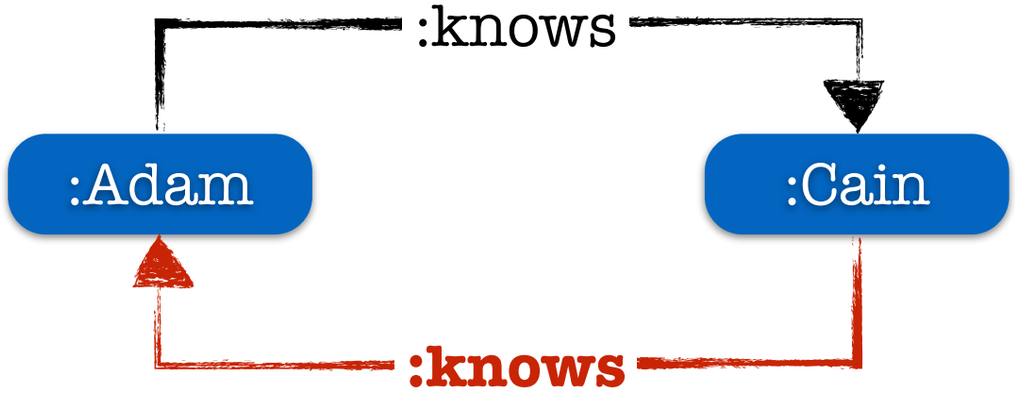
- **Disjointness**: C disjointWith D.
  - Intuitive Semantik: Es kann nichts geben, das gleichzeitig C und D ist.
  - Z.B.: Person disjointWith Island.

Elemente der Semantic Web Ontology Language – OWL (III)

- **Inverse properties**: R inverseOf S.
  - Intuitive Semantik: Die gegenteilige Relation zu R heißt S.
  - Z.B.: hasFather inverseOf isFatherOf
    - Adam isFatherOf Cain
    - —> Cain hasFather Adam

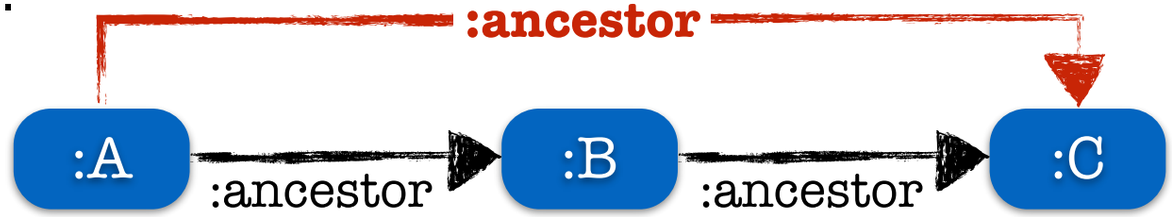


- **Symmetric properties**: R type SymmetricProperty.
  - Semantik: Wenn A R B gilt, dann gilt auch B R A.
  - Z.B.: knows type SymmetricProperty
    - Adam knows Cain
    - —> Cain knows Adam

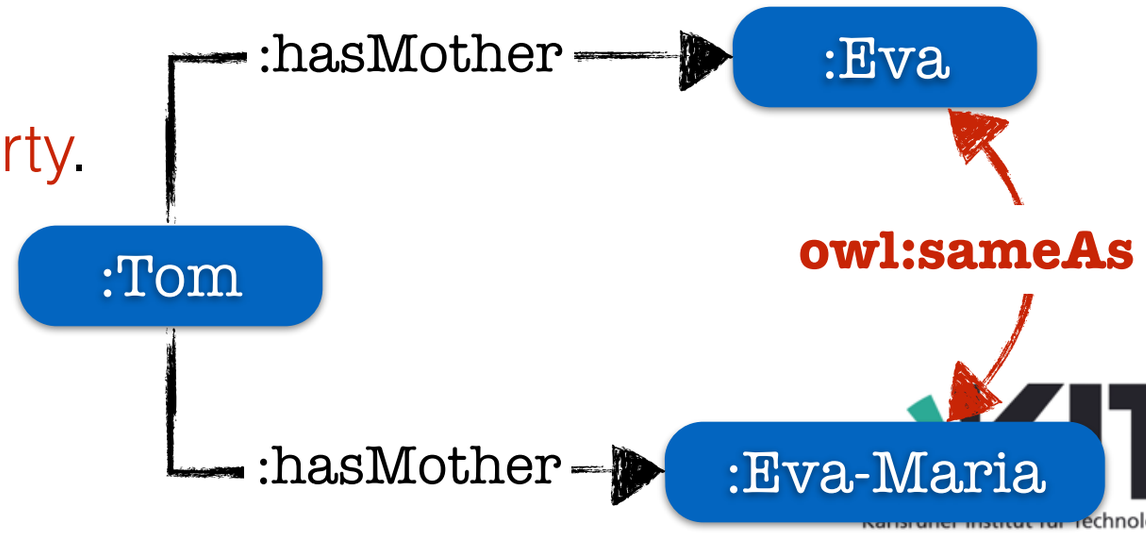


Elemente der Semantic Web Ontology Language – OWL (IV)

- **Transitive properties**: R type TransitiveProperty.
  - Semantik: Wenn A R B und B R C, dann auch A R C.
  - Z.B.: ancestor type TransitiveProperty.



- **Functional properties**: R type FunctionalProperty.
  - Intuitive Semantik: Die Relation R ist eine Funktion, d.h. von jedem Individuum aus kann über diese Relation nur max. ein Individuum erreicht werden.
  - Z.B.: hasMother type FunctionalProperty.



Elemente der Semantic Web Ontology Language – OWL (V)

- **Object properties**: R type ObjectProperty.
  - Semantik: Relationen zwischen Instanzen von Klassen. (So wie alle bisher vorgestellten Properties)
  - Z.B.: hasFather type ObjectProperty.

- **DatatypeProperty properties**: R type DatatypeProperty.
  - Intuitive Semantik: Relationen zwischen Instanzen von Klassen und RDF-Literalen oder XML-Schema-Datentypen.
  - Z.B.: hasAge type DatatypeProperty.
    - hasAge range integer.

Overview of the Lecture

- Grundlegende Begriffe und Konzepte
  - Integriertes Informationssystem
  - Heterogenität
- Ontologien
  - Definitionen
  - Eigenschaften
  - Formale Sprachen für Ontologien
- Ontologie-Sprachen
  - RDF
  - RDF/S
  - OWL
- **Beispiel – Semantische Integration**
  - **Semantisches Mapping von Wissensbasen**

Ontology-based Information Integration – An Example

Assertionale Wissensbasis 1 (A-Box)

- **In natural language:**

  Rudi Studer is a researcher working on the NeOn project, which deals with Ontology Engineering.

- **In a logic language:**

  PersonX **type** Researcher.

  PersonX **first_name** “Rudi”.

  PersonX **last_name** “Studer”.

  PersonX **works_in** Project1.

  Project1 **type** Project.

  Project1 **name** “NeOn”.

  Project1 **isAbout** Ontology_Engineering.

Assertionale Wissensbasis 2 (A-Box)

In natural language:

Enrico Motta is a professor managing the NeOn project, funded by the European Comm.

In a logic language:

PersonA **type** Professor.

PersonA **given_name** “Enrico”.

PersonA **familiy_name** “Motta”.

ProjectA **type** ResearchProject.

ProjectA **title** “NeOn”.

ProjectA **managedBy** PersonA.

ProjectA **fundedBy** European_Commission.

Zusammenführung von Informationen aus unterschiedlichen Quellen durch Ontology Mapping

PersonX **type** Researcher.

PersonX **first_name** “Rudi”.

PersonX **last_name** “Studer”.

PersonX **works_in** Project1.

Project1 **type** Project.

Project1 **name** “NeOn”.

Project1 **isAbout**
Ontology_Engineering.

PersonA **type** Professor.

PersonA **given_name** “Enrico”.

PersonA **familiy_name** “Motta”.

ProjectA **type** ResearchProject.

ProjectA **title** “NeOn”.

ProjectA **managedBy** PersonA.

ProjectA **fundedBy**
European_Commission.

Mapping Ontology (T-Box):

first_name **equivalentProperty**
given_name .

last_name **equivalentProperty**
family_name .

name **equivalentProperty** title .

Professor **subClassOf** Researcher .

works_in **inverseProperty** member .

Project1 **sameAs** ProjectA .

Erweiterung der Wissensbasis durch Inference

PersonX **type** Researcher.

PersonX **first_name** “Rudi”.

PersonX **last_name** “Studer”.

PersonX **works_in** Project1.

Project1 **type** Project.

Project1 **name** “NeOn”.

Project1 **isAbout**
Ontology_Engineering.

PersonA **type** Professor.

PersonA **given_name** “Enrico”.

PersonA **familiy_name** “Motta”.

ProjectA **type** ResearchProject.

ProjectA **title** “NeOn”.

ProjectA **managedBy** PersonA.

ProjectA **fundedBy**
European_Commission.

Inferred Statements:

PersonA type Researcher .

da Researcher **subClassOf** Professor (*).

PersonA differentFrom PersonX .

wichtig, da Non-unique Naming Assumption

ProjectA type Project .

da ResearchProject **subClassOf** Project (*).

Project1 managedBy PersonA .

Project1 fundedBy European_Commission .

da Project1 **sameAs** ProjectA .

Referenzen und weiterführende Literatur

- **Ontologien**
  - A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, L. Schneider: Sweetening Ontologies with DOLCE. EKAW 2002: 166-181, Springer, Berlin Heidelberg 2002.
  - A. Gómez-Pérez, M. Fernández-López, O. Corcho: Ontological Engineering: with Examples from the Areas of Knowledge Management, E-Commerce and the Semantic Web. (Advanced Information and Knowledge Processing). Springer-Verlag New York, 2007.
  - P. Hitzler, M. Krötzsch, S. Rudolph, Y. Sure: Semantic Web – Grundlagen. Springer, Berlin Heidelberg, eXamen.press, 2008
  - I. A. Richards, C. K. Ogden: The Meaning of Meaning. 1989 (Erstausgabe 1923). dt. Die Bedeutung der Bedeutung. Eine Untersuchung über d. Einfluss d. Sprache auf d. Denken u. über d. Wiss. d. Symbolismus. Suhrkamp, Frankfurt (am Main) 1974.
  - Steffen Staab, Rudi Studer (Hrsg.): Handbook on Ontologies (2. Auflage). International Handbooks on Information Systems Springer, Berlin Heidelberg 2008.
  - Miller, George A. WordNet: a lexical database for English. Communications of the ACM 38.11 (1995): 39-41.
  - McGuinness, Deborah L. Ontologies come of age., in Fensel et al. (eds.) Spinning the Semantic Web: Bringing the World Wide Web to its full potential, The MIT Press(2003).
  - Gruber, Thomas R. "A translation approach to portable ontology specifications." Knowledge acquisition 5.2 (1993): 199-220.
  - Noy, Natalya F., and Michel Klein. "Ontology evolution: Not the same as schema evolution." Knowledge and information systems 6.4 (2004): 428-440.
  - Noy, Natalya Fridman, and Carole D. Hafner. "The state of the art in ontology design: A survey and comparative review." AI magazine 18.3 (1997): 53.

Referenzen und weiterführende Literatur

- **RDF, RDF/S und OWL**
  - http://www.w3.org/RDF/
    zentrale Website für RDF sowie entsprechender Anwendungen
  - http://www.w3.org/TR/rdf-schema/
    RDF Vocabulary Description Language 1.0: RDF Schema Seite des W3C.
  - http://www.w3.org/2004/OWL/
    zentrale W3C Webseite für OWL.
  - http://www.w3.org/TR/owl-features/
    Überblick über OWL.
  - http://www.w3.org/TR/owl-ref/
    vollständige Beschreibung der OWL-Sprachkomponenten.
  - http://www.w3.org/TR/owl-guide/
    zeigt, wie OWL zur Wissensmodellierung verwendet werden kann.
  - http://www.w3.org/TR/owl-semantics/
    beschreibt die Semantik von OWL.
  - Deutsche Übersetzungen mancher W3C Dokumente findet man unter http://www.w3.org/2005/11/Translations/Lists/ListLang-de.html