

Vorlesung Semantic Web



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Vorlesung im Wintersemester 2011/2012

Dr. Heiko Paulheim

Fachgebiet Knowledge Engineering

Was bisher geschah...

- Kann RDF mehr als XML?
- XML ist eine Auszeichnungssprache für Informationen
- In XML kann man beliebige Tags und Attribute definieren
- XML-Tagnamen haben für den Computer keine Bedeutung
- RDF ist eine Auszeichnungssprache für Informationen
- In RDF kann man beliebige Klassen / Relationen definieren
- RDF-Bezeichner haben für den Computer keine Bedeutung

Heute: Ontologien

- Jetzt kommt endlich die Semantik!
- Einfache Ontologien mit RDF Schema bauen
- Elemente von RDF Schema
- Automatisches Schlussfolgern mit RDF Schema

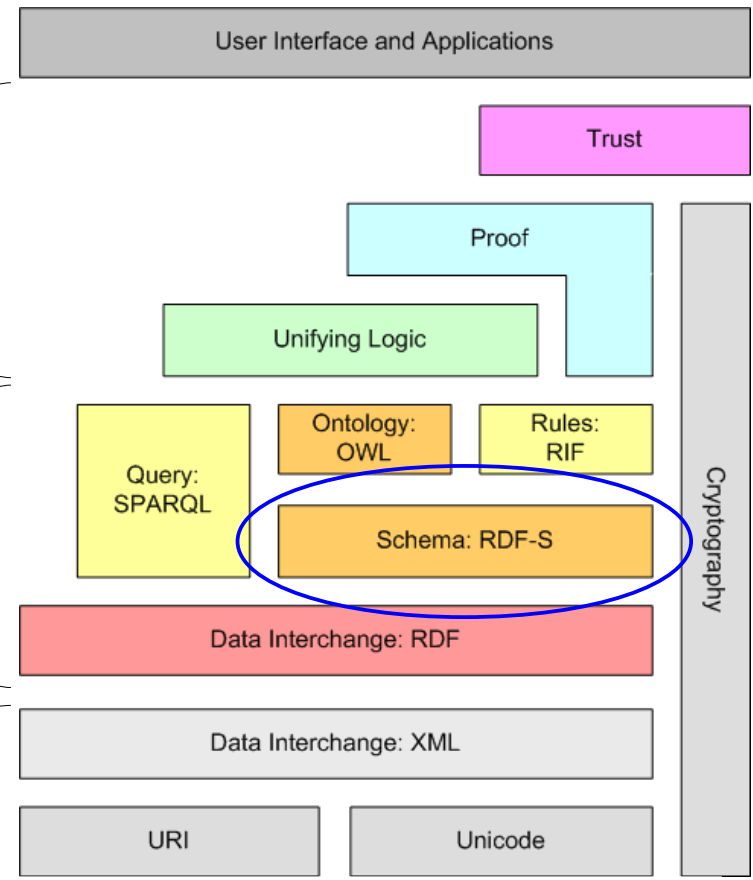
Semantic Web – Aufbau



here be dragons...

Semantic-Web-
Technologie
(Fokus der
Vorlesung)

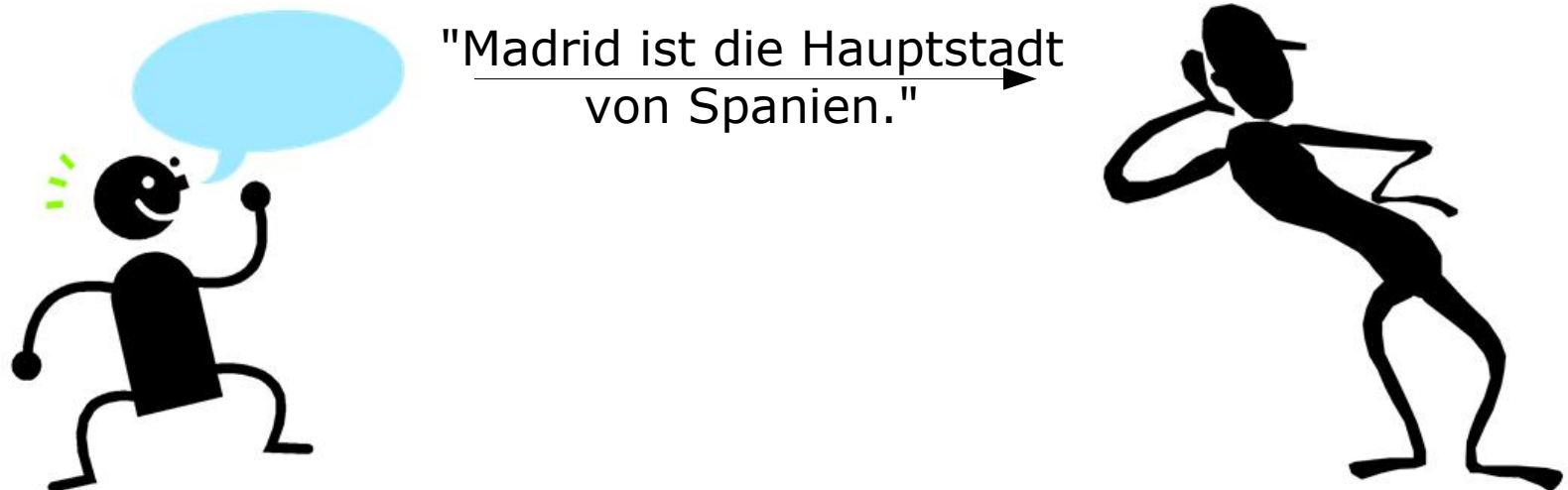
Technische
Grundlagen



Berners-Lee (2009): *Semantic Web and Linked Data*
<http://www.w3.org/2009/Talks/0120-campus-party-tbl/>

Was fehlt bis jetzt?

- Computer verstehen die Informationen im Web nicht
- Aber was heißt eigentlich *verstehen*?



Semantik (revisited)

- Betrachten wir folgenden Satz:

"Madrid ist die Hauptstadt von Spanien."

- Im Semantic Web (RDF):

:Madrid :capitalOf :Spain .

- Wie viele Informationen können wir [Menschen] aus diesem Satz erhalten?
 - (1 Information = 1 Aussagesatz $\langle S, P, O \rangle$)
 - Schätzungen? Meinungen?

Semantik (revisited)

- Betrachten wir folgenden Satz:

"Madrid ist die Hauptstadt von Spanien."

- Aussagen, die wir erhalten können:
 - "Madrid ist die Hauptstadt von Spanien."
 - "Spanien ist ein Land."
 - "Madrid ist eine Stadt."
 - "Madrid liegt in Spanien."
 - "Barcelona ist nicht die Hauptstadt von Spanien."
 - "Madrid ist nicht die Hauptstadt von Frankreich."
 - "Madrid ist kein Land."
 - ...

Wie funktioniert Semantik?



TECHNISCHE
UNIVERSITÄT
DARMSTADT

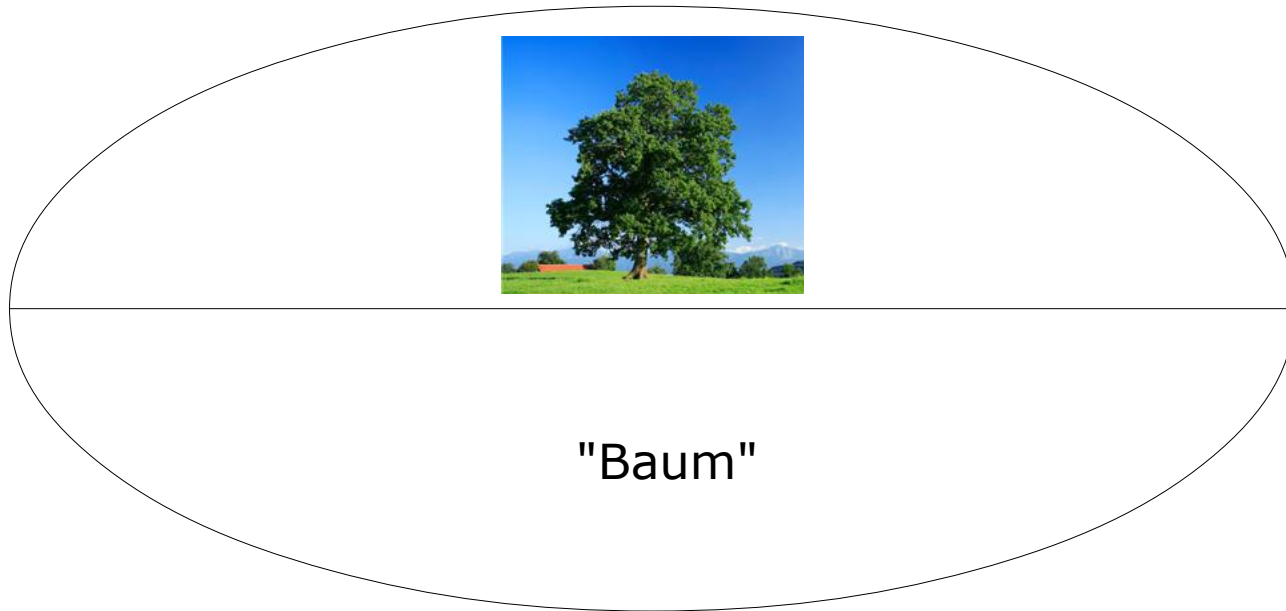


Ausflug in die Linguistik: Das Zeichenmodell von Saussure



TECHNISCHE
UNIVERSITÄT
DARMSTADT

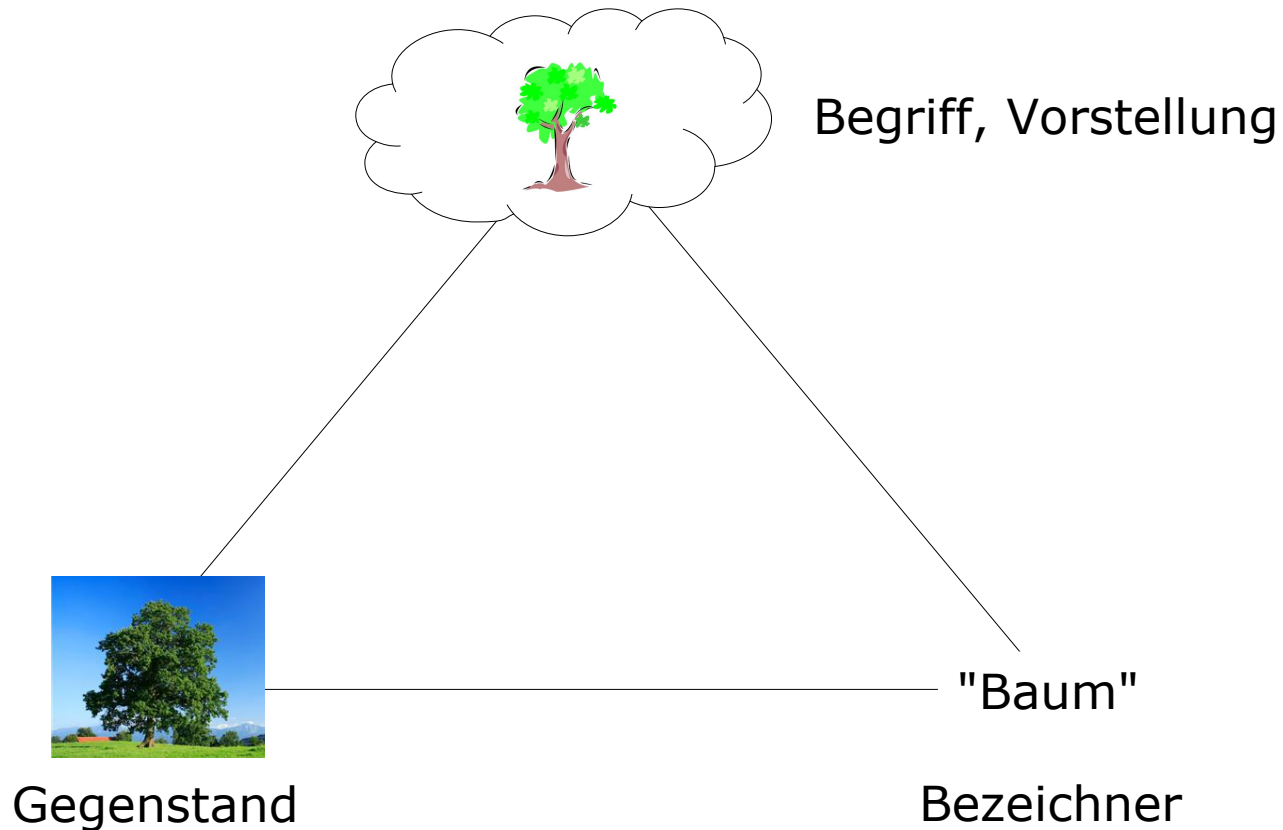
- Ferdinand de Saussure (1857-1913):
 - Zeichen (signifiant) und Bezeichnetes (signifié) untrennbar verbunden



Ausflug in die Linguistik: Das semiotische Dreieck



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Charles Odgen (1923): *The Meaning of Meaning*.

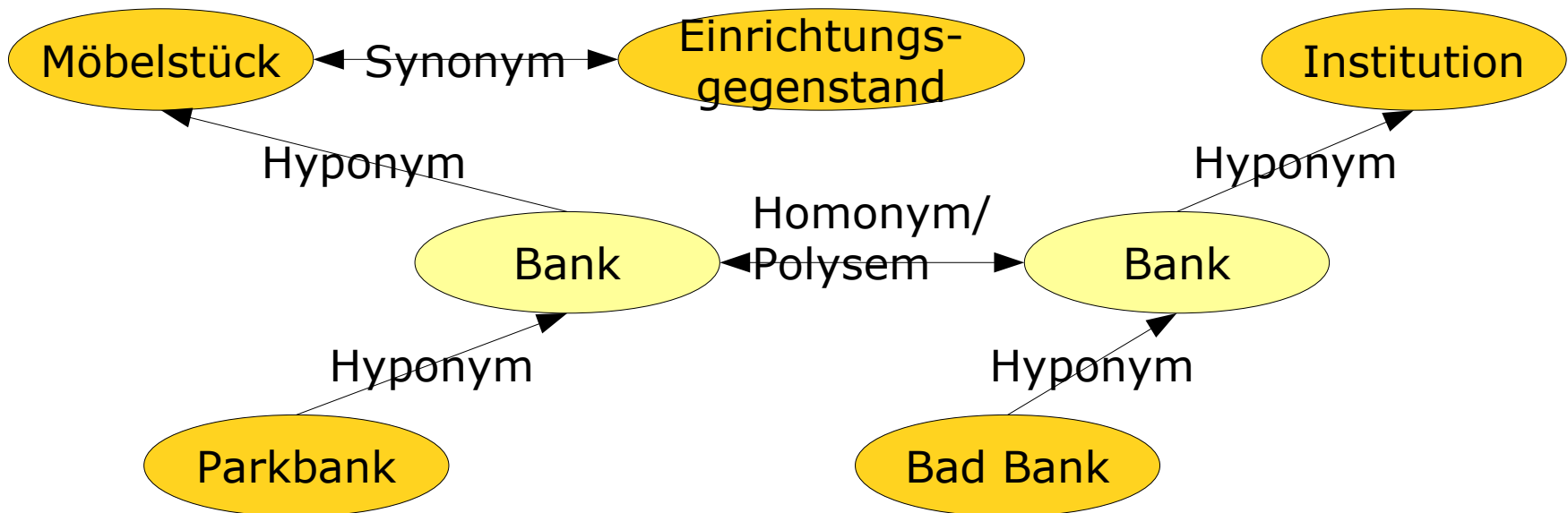
Wie funktioniert Semantik?



TECHNISCHE
UNIVERSITÄT
DARMSTADT

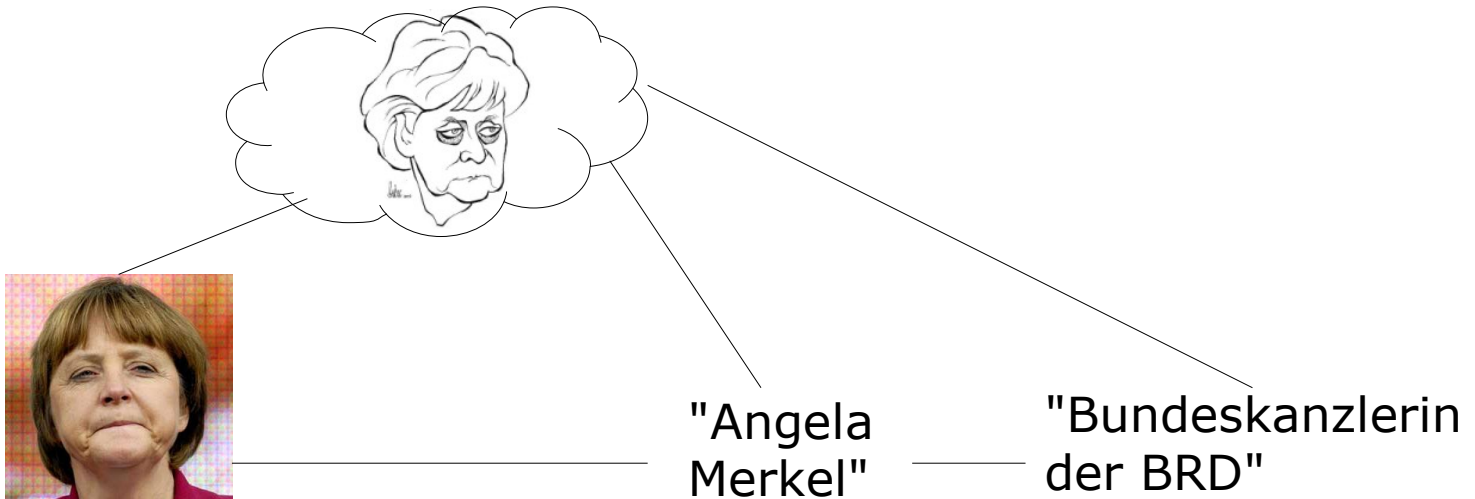
- Lexikalische Semantik
 - Die Bedeutung wird bestimmt durch Beziehungen zu anderen Begriffen
- Extensionale Semantik:
 - Die Bedeutung wird bestimmt durch alle Instanzen
- Intensionale Semantik, z.B. Merkmalssemantik
 - Die Bedeutung wird bestimmt durch Eigenschaften, die eine Instanz haben muss
- Prototypensemantik
 - Die Bedeutung wird bestimmt durch die Nähe zu einer prototypischen Instanz
- ...

- Definiert Begriffe über Beziehungen zu anderen Begriffen



Extensionale Semantik

- Aufzählung von Instanzen
 - Beispiel: EU-Mitgliedsstaaten sind Belgien, Bulgarien, Dänemark, Deutschland, ..., Zypern.
- *Angela Merkel == Bundeskanzlerin der BRD*
 - beide Begriffe haben dieselbe Extension



Intensionale Semantik

- Beschreibt Eigenschaften von Dingen
- Seme: bedeutungsunterscheidende Elemente

Begriff	Hat Flügel	Kann schwimmen	Hat Fell	Hat Federn
Ente	+	+	-	+
Vogel	+	O	-	+
Biene	+	-	-	-
Delphin	-	+	-	-
...				

Intensionale vs. extensionale Semantik

- Intensional verschiedene Dinge können extensional gleich sein
- Klassisches Beispiel: Morgenstern und Abendstern

Begriff	Himmelskörper	hell	sichtbar am Morgenhimmel	sichtbar am Abendhimmel
Morgenstern	+	+	+	-
Abendstern	+	+	-	+
...				

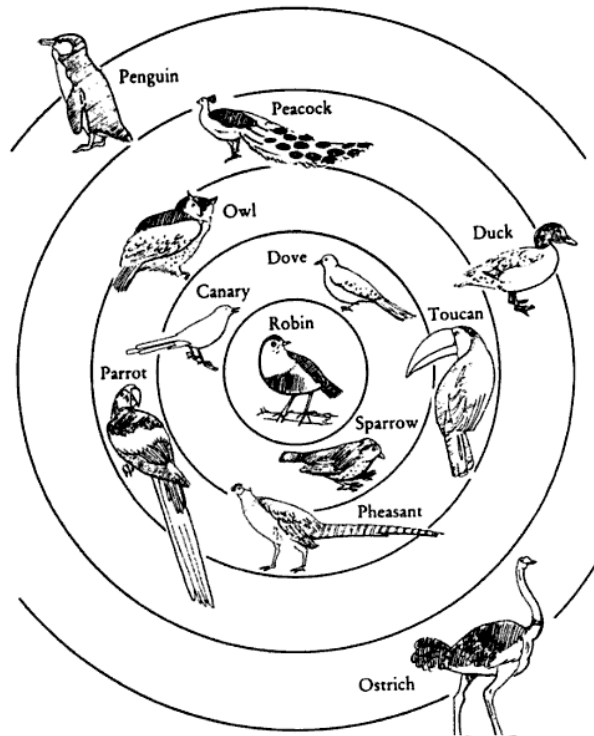
- aber: beide Begriffe haben dieselbe Extension (die Venus)

Intensionale vs. Extensionale Semantik

- Die Extension kann sich über die Zeit ändern, ohne dass die Intension sich ändert
 - z.B.: die Extension von "Student"
 - ändert das die Semantik?
- Auch die Intension kann sich über die Zeit ändern
- Die Extension eines Begriffes kann auch leer sein
 - Einhorn
 - Marsmensch
 - Yeti (?)
 - ...

Prototypensemantik

- Intentionale und extensionale Semantik basieren auf bool'scher Logik
- Prototypensemantik:
Fuzzy-Variante



Jean Aitchison: Words in the Mind (1987)

Wie funktioniert Semantik?



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Semantik definiert die Bedeutung von Begriffen
- Im Semantic Web machen wir das auch
 - Methoden aus lexikalischer, intensionaler und extensionaler Semantik

© Original Artist
Reproduction rights obtainable from
www.CartoonStock.com



<http://walkinthewords.blogspot.com/2008/05/linguistic-cartoon-favorites-semantics.html>

Wie funktioniert Semantik?



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Semantik im Semantic Web



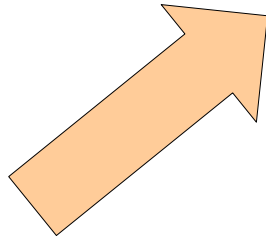
TECHNISCHE
UNIVERSITÄT
DARMSTADT

$\text{City}(x) \Leftarrow \exists y: \text{capitalOf}(x, y)$

$\text{Country}(y) \Leftarrow \exists x: \text{capitalOf}(x, y)$

$\text{locatedIn}(x, y) \Leftarrow \text{capitalOf}(x, y)$

...



$\text{:Madrid :capitalOf :Spain .}$



$\text{City}(x) \leftarrow \exists y: \text{capitalOf}(x,y)$

$\text{Country}(y) \leftarrow \exists x: \text{capitalOf}(x,y)$

$\text{locatedIn}(x,y) \leftarrow \text{capitalOf}(x,y)$

...

- "An ontology is an explicit specification of a conceptualization."¹
- Ontologien codieren das Wissen einer Domäne
- Sie bilden ein gemeinsames Vokabular
 - und beschreiben die Semantik der darin enthaltenen Begriffe

¹ Gruber (1993): *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*.
In: International Journal Human-Computer Studies Vol. 43, Issues 5-6, pp. 907-928.

Was ist eigentlich eine Ontologie?



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Ontologie (ohne Artikel) ist die "Lehre vom Seienden"
 - griechisch: *ὄντος* (das, was ist), *λόγος* (die Lehre)
 - Subdisziplin der Philosophie
- In der Informatik (mit Artikel)
 - eine formalisierte Beschreibung einer Domäne
 - ein gemeinsam genutztes Vokabular
 - eine logische Theorie

Ontologie – weitere Definitionen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Guarino und Giaretta (1995):
"a logical theory which gives an explicit, partial account of a conceptualization"

- Uschold und Gruninger (1996):
"shared understanding of some domain of interest"
"an explicit account or representation of some part of a conceptualisation"

- Guarino (1998):
"a set of logical axioms designed to account for the intended meaning of a vocabulary"

Ontologie – essentielle Eigenschaften



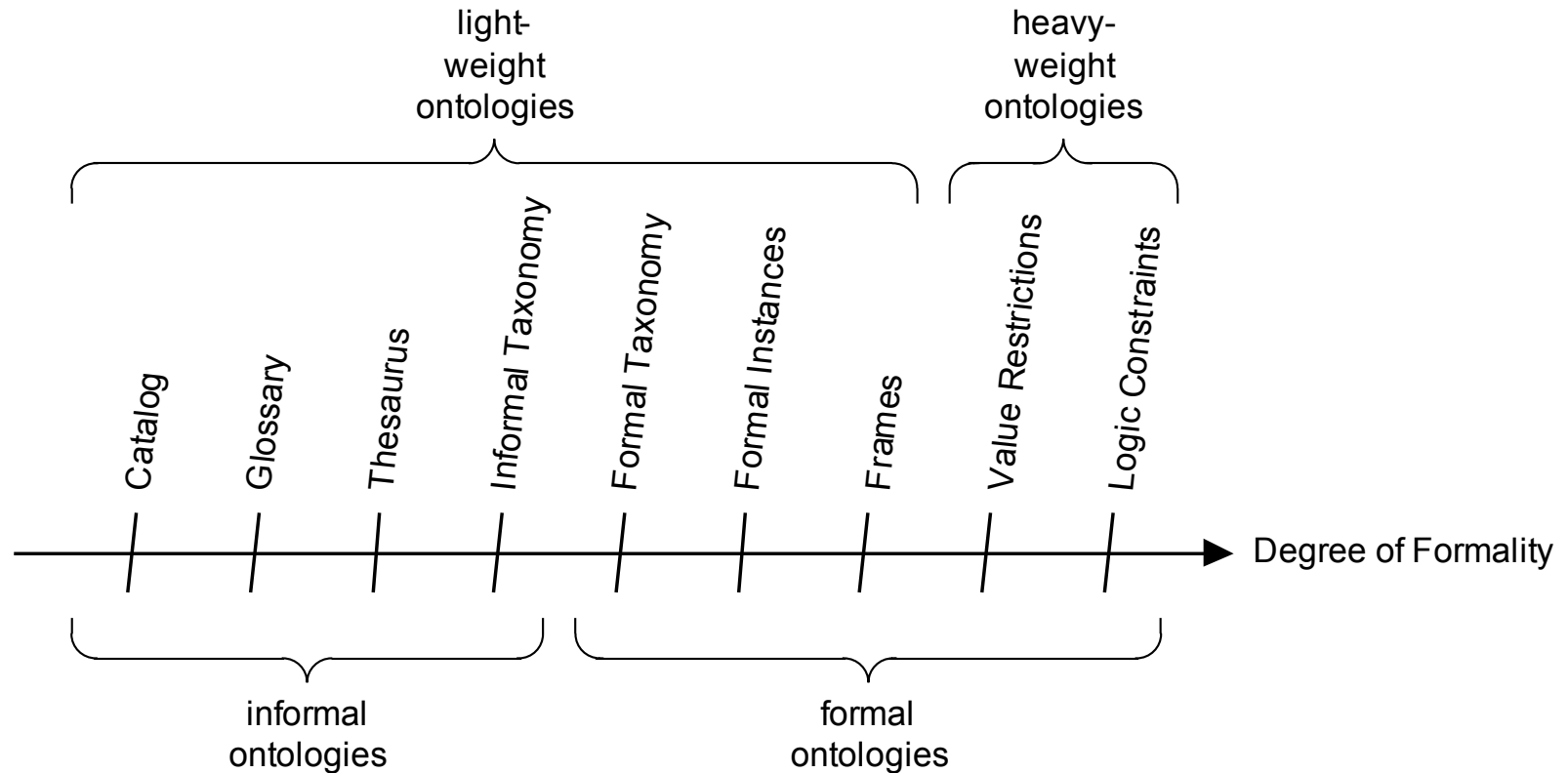
TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Explizit
 - Keine "versteckten" Bedeutungen
- Formal
 - z.B. Logik-Sprachen, Regeln, ...
- Geteilt
 - Martin Hepp: "Autists don't build Ontologies"
 - Eine Ontologie für einen allein ist nicht unbedingt sinnvoll
- Partiell
 - Eine komplette "Welt-Ontologie" wird es (wahrscheinlich) nie geben

Klassifikation von Ontologien



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Lassila & McGuinness (2001): *The Role of Frame-Based Representation on the Semantic Web*.
In: Linköping Electronic Articles in Computer and Information Science 6(5).

Klassifikation von Ontologien: informelle Ontologien



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Katalog
 - Begriffssammlung, geordnetes Vokabular
- Glossar
 - Katalog + Erklärungen zu Begriffen (informell, textuell)
- Thesaurus
 - Glossar + Verweise auf andere Begriffe
 - Synonyme, Antonyme, Hyponyme, Hyperonyme
- Informelle Taxonomie
 - Subklassenbeziehungen
 - können "unecht" sein: $C \supset D \not\Rightarrow D(x) \rightarrow C(x)$
 - Beispiel Webshop-Hierarchie: "Fachbücher" \supset "Informatik"

Klassifikation von Ontologien: Informelle Ontologien

- Informelle Ontologien helfen im Semantic Web kaum weiter
 - textuelle Beschreibungen
 - unechte Beziehungen
- Informelle Taxonomien können zu falschen Schlussfolgerungen führen
- Beispiel:
 - Fachbücher \supset Informatik \supset Algorithmen
 - Quicksort \in Algorithmen \rightarrow Quicksort \in Fachbücher??

Klassifikation von Ontologien: formale Ontologien



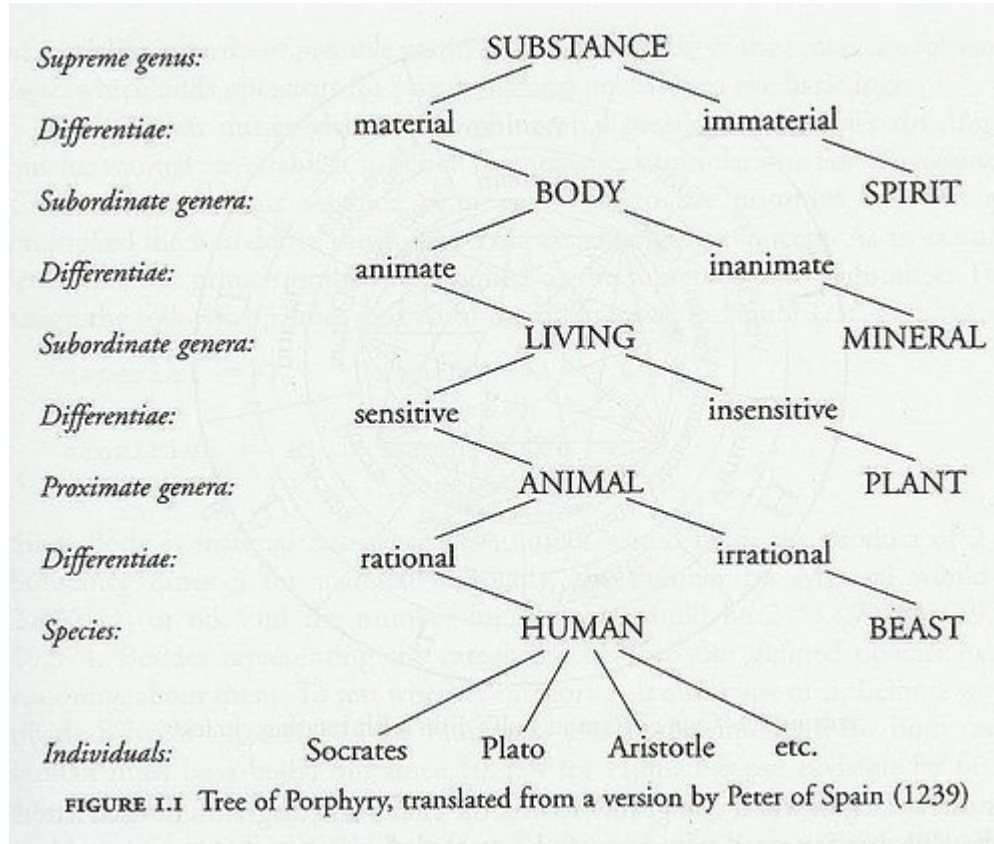
TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Formale Taxonomien
 - echte Subklassenbeziehung: $C \supset D \Rightarrow D(x) \rightarrow C(x)$
- Formale Instanzen
 - Formale Taxonomien plus Instanzen
 - z.B. Spanien \in Länder
- Frames
 - Formale Instanzen + Beziehungen zwischen Dingen
 - z.B. hatHauptstadt(Spanien, Madrid)
- Wertrestriktionen
 - Frames + Einschränkungen von Definitions- und Wertebereich
 - z.B. "Die Hauptstadt eines Landes kann nur eine Stadt sein."
- Logische Bedingungen
 - Alles, was darüber hinausgeht (z.B. komplexe Regeln)

Die älteste Ontologie



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Porphyrios, griechischer Philosoph, ca. 233-301

Einfache Ontologien codieren: RDF Schema (RDFS)

- Standardisiert vom W3C (2004)
- Wichtigstes Element: Klassen



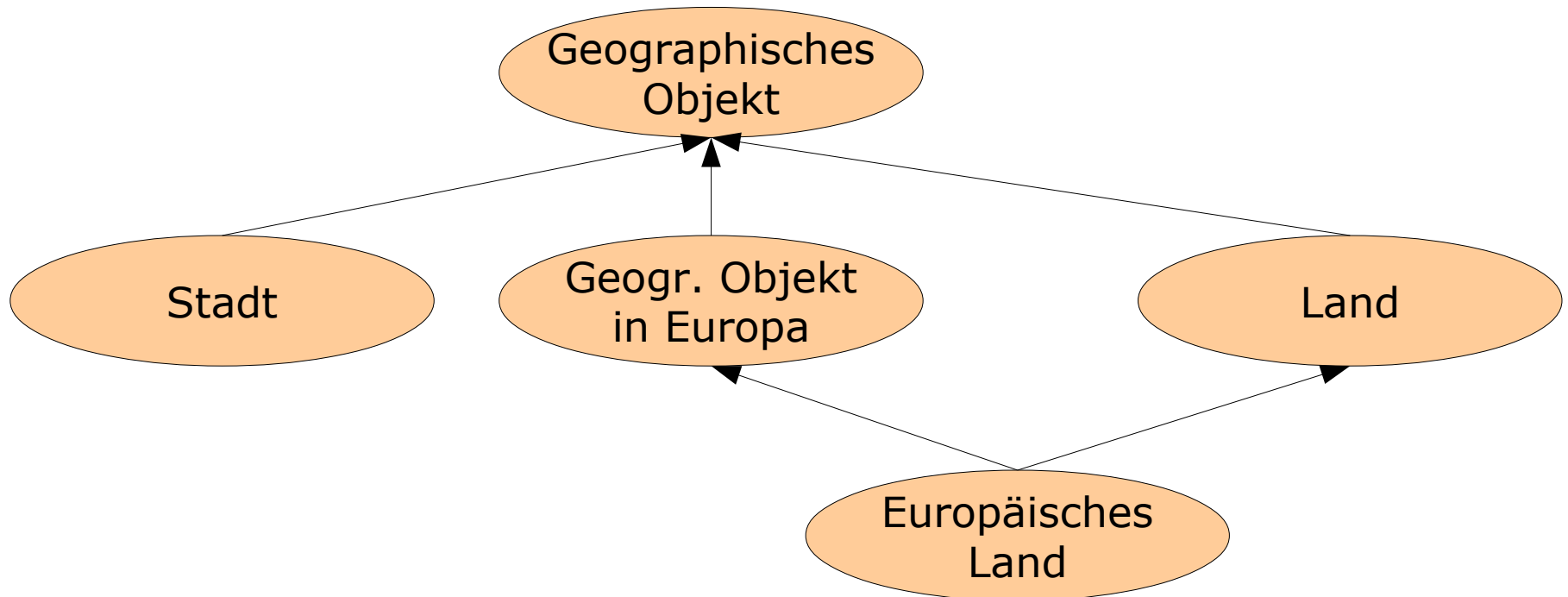
```
:Country a rdfs:Class .
```

- Klassen bilden Hierarchien

```
:EuropeanCountry rdfs:subClassOf :Country .
```

Klassenhierarchien in RDF Schema

- Mehrfachvererbung ist erlaubt

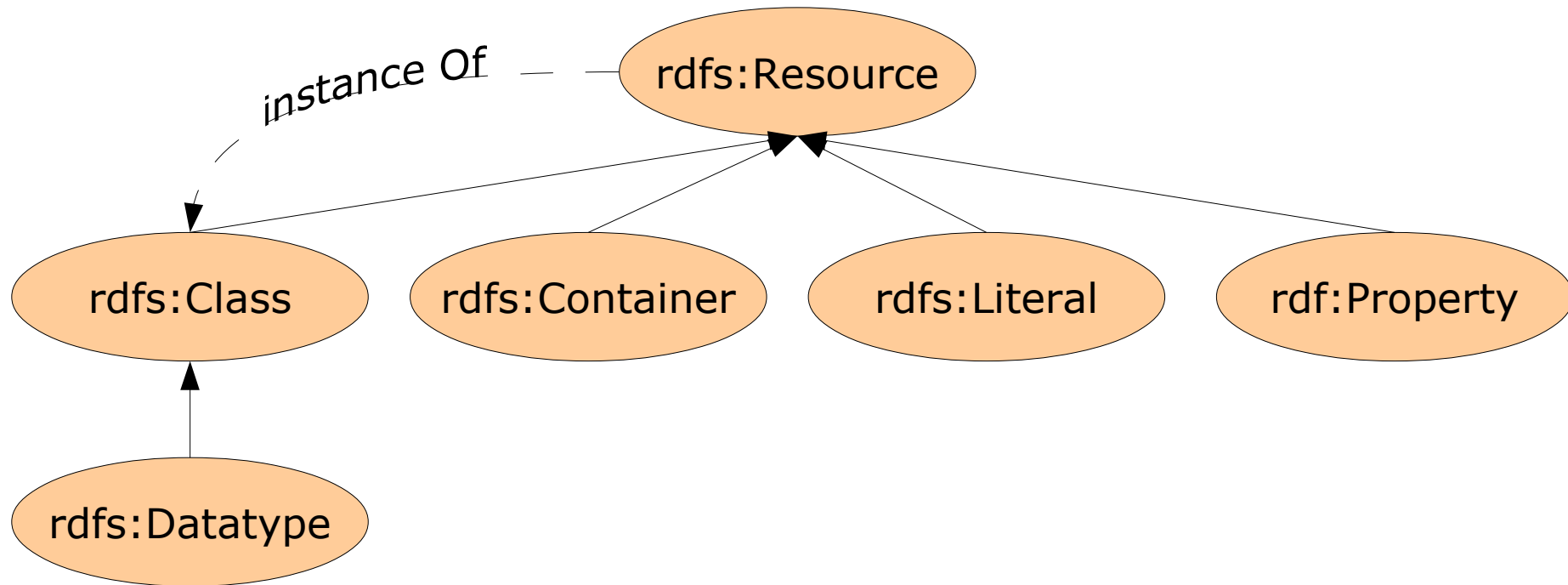


Konvention für diese VL: unbeschriftete Pfeile = `rdfs:subClassOf`

Einige vordefinierte Konzepte in RDFS



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Properties in RDF Schema

- Properties sind das zweitwichtigste Element
- entsprechen zweiwertigen Prädikaten

```
:Madrid :capitalOf :Spain .  
:capitalOf a rdf:Property .
```

- Auch Properties bilden Hierarchien

```
:capitalOf rdfs:subPropertyOf :locatedIn .
```

Definitions- und Wertebereiche von Properties



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Properties existieren prinzipiell losgelöst von Klassen
 - das ist anders als z.B. in OOP
- Festlegen von Definitions- und Wertebereich:

```
:capitalOf rdfs:domain :City .  
:capitalOf rdfs:range :Country .
```

- Definitions- und Wertebereich werden an Sub-Properties vererbt
- Können dort weiter eingeschränkt werden (dazu später mehr)

Vordefinierte Properties

- Einige haben wir schon kennen gelernt:

```
rdf:type  
rdf:first  
rdf:rest  
rdf:_1, rdf:_2, ...  
    ⊆ rdfs:containerMembershipProperty
```

```
rdfs:subClassOf  
rdfs:subPropertyOf  
rdfs:domain  
rdfs:range
```

Weitere vordefinierte Properties

- Labels:

- `:Germany rdfs:label "Deutschland"@de .`

- `:Germany rdfs:label "Germany"@en .`

- Kommentare:

- `:Germany rdfs:comment "Germany as a political entity."@en .`

- Verweise auf weitere Ressourcen:

- `:Germany rdfs:seeAlso <http://www.deutschland.de/> .`

- Verweis auf definierendes Schema:

- `:Country rdfs:isDefinedBy <http://foo.bar/countries.rdfs> .`

URIs vs. Labels

- Ein URI ist letztlich nur ein eindeutiger Bezeichner
 - eben ein Identifier
 - muss nicht als solcher verstehbar sein

`http://www.countries.org/4327893`

- Labels sind für das menschliche Verständnis gedacht
- und potentiell mehrsprachig

```
countries:4327893 rdfs:label "Deutschland"@de .
countries:4327893 rdfs:label "Germany"@en .
countries:4327893 rdfs:label "Tyskland"@sv .
...
```

URIs vs. Labels



- Auch auf Schemaebene können Labels vergeben werden

```
:Country a rdfs:Class .
```

```
:Country rdfs:label "Land"@de .
```

```
:Country rdfs:label "Country"@en .
```

```
:locatedIn a rdf:Property .
```

```
:locatedIn rdfs:label "liegt in"@de .
```

```
:locatedIn rdfs:label "is located in"@en .
```

```
:locatedIn rdfs:comment "bezogen auf geographische Lage" .
```

RDF Schema und RDF

- Jedes RDF Schema ist selbst ein gültiges RDF-Dokument
- Damit gelten alle Eigenschaften von RDF auch für RDF Schema!
- Non-unique Naming Assumption

```
schema1:Country a rdfs:Class .  
schema2:Land a rdfs:Class .
```

- Open World Assumption

```
:Country rdfs:subClassOf :GeographicObject .  
:City rdfs:subClassOf :GeographicObject .
```

Unsere erste Ontologie

▪ Städte, Länder und ihre Hauptstädte

```
:Country a rdfs:Class .  
:City a rdfs:Class .  
:locatedIn a rdf:Property .  
:capitalOf rdfs:subPropertyOf :locatedIn .  
:capitalOf rdfs:domain :City .  
:capitalOf rdfs:range :Country .
```

Beschreibung
der Terminologie
(T-Box)

Beschreibung der
Assertionen oder
Behauptungen
(A-Box)

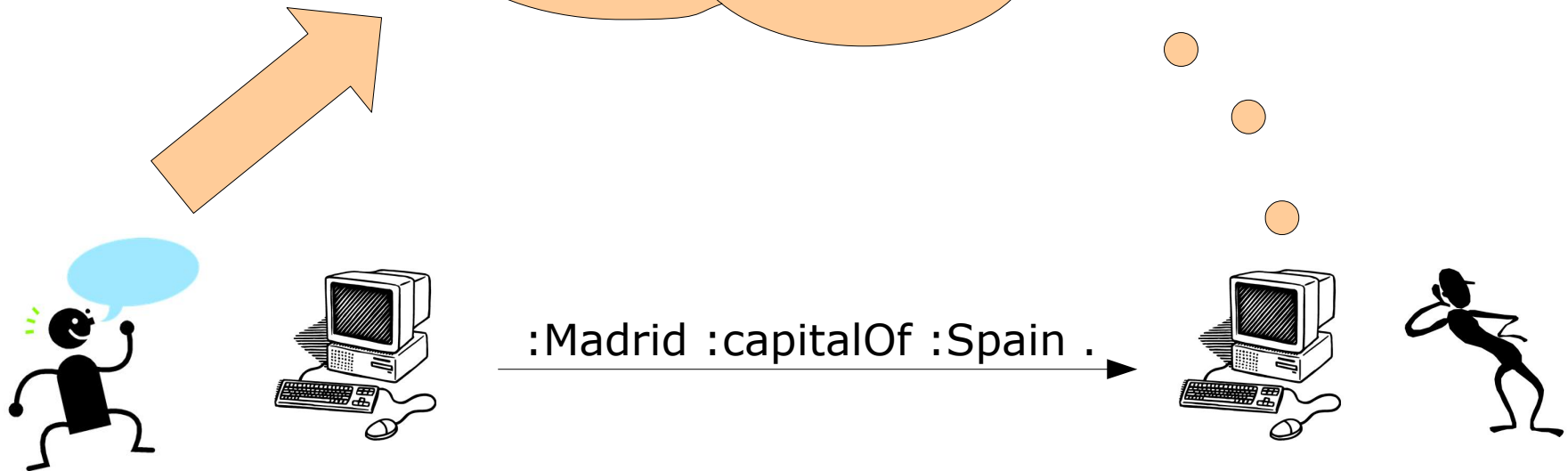
```
:Madrid :capitalOf :Spain .
```


Was haben wir jetzt gewonnen?



TECHNISCHE
UNIVERSITÄT
DARMSTADT

```
:Country a rdfs:Class .  
:City a rdfs:Class .  
:locatedIn a rdfs:Property .  
:capitalOf rdfs:subPropertyOf :locatedIn .  
:capitalOf rdfs:domain :City .  
:capitalOf rdfs:range :Country .
```



Was haben wir jetzt gewonnen?



TECHNISCHE
UNIVERSITÄT
DARMSTADT

```
:Madrid :capitalOf :Spain .  
+ :capitalOf rdfs:domain :City  
→ :Madrid a :City .
```

```
:Madrid :capitalOf :Spain .  
+ :capitalOf rdfs:range:Country  
→ :Spain a :Country .
```

```
:Madrid :capitalOf :Spain .  
+ :capitalOf rdfs:subPropertyOf :locatedIn .  
→ :Madrid :locatedIn :Spain .
```

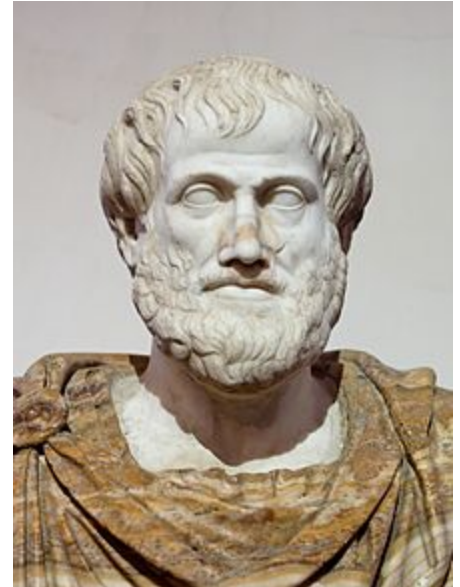
Reasoning auf RDF

- Mit Hilfe von RDF Schema kann man *deduktiv* schließen
- Das heißt,
 - aus Regeln und Fakten
 - neue Fakten ableiten
- Tools dazu heißen Reasoner

- Das Gegenteil hierzu ist *induktives* Schließen
 - aus Fakten Regeln ableiten
 - wird u.a. in der Vorlesung "Maschinelles Lernen" behandelt

Kleine Geschichte des Reasoning

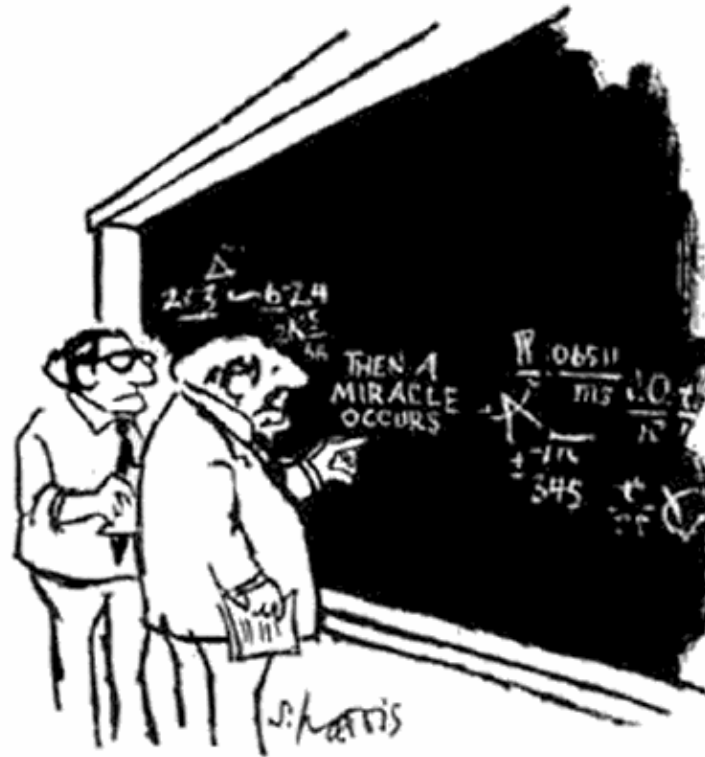
- Aristoteles (384 – 322 v.Chr.)
- Syllogismen
 - Ableitung von Fakten mit Hilfe von Regeln
- Beispiel:
 - Alle Menschen sind sterblich.
 - Sokrates ist ein Mensch.
 - Sokrates ist sterblich.



Wie funktioniert Reasoning mit RDF Schema?



TECHNISCHE
UNIVERSITÄT
DARMSTADT



"I THINK YOU SHOULD BE MORE EXPLICIT
HERE IN STEP TWO."

<http://www.flickr.com/photos/skepticalist/4372728626/>

Definition: Extension

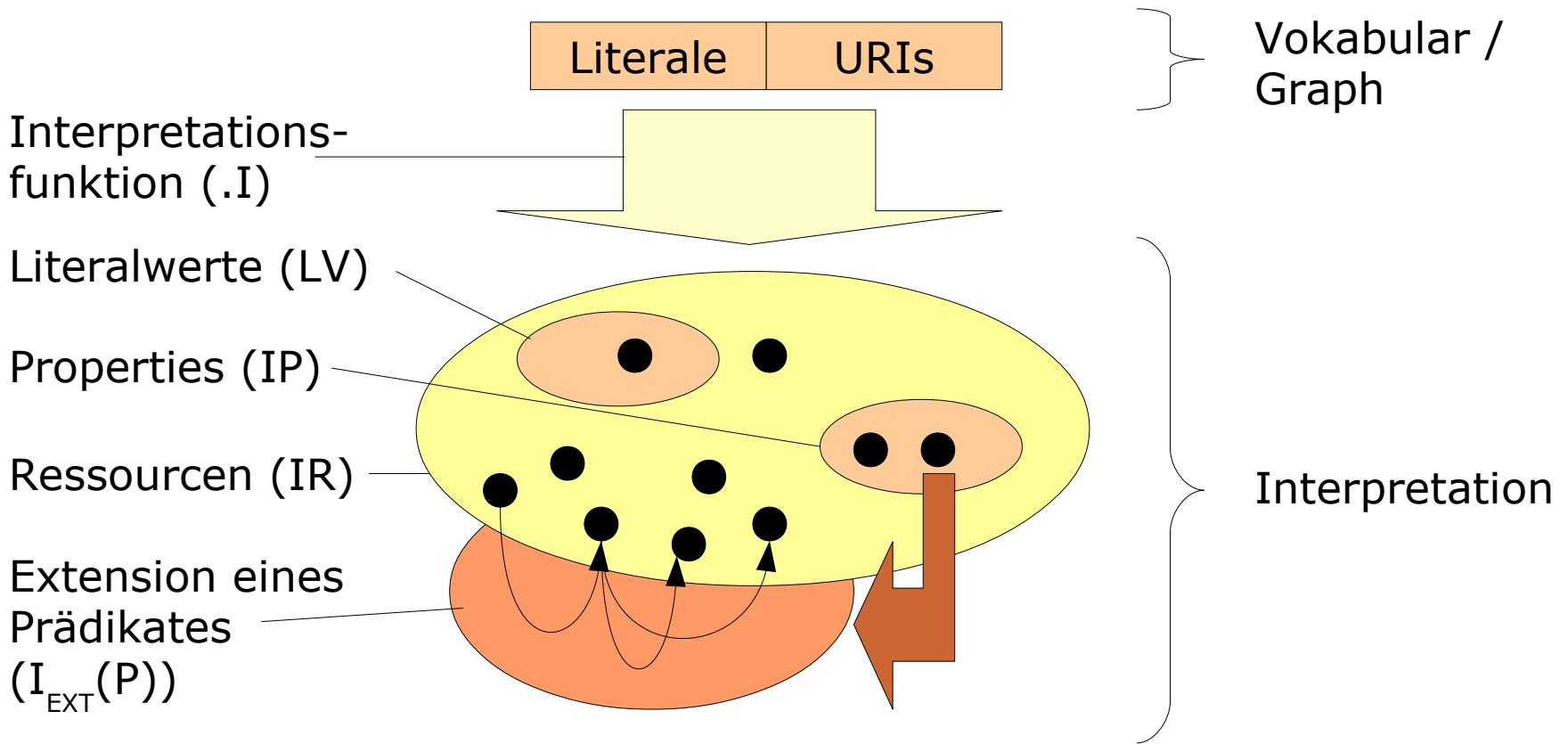
- Wiederholung Semantik:
 - Menge aller Dinge, die von einem Begriff identifiziert werden
 - z.B. Menge aller Häuser ist die Extension von "Haus"
 - kann auch leer sein (z.B. Extension von "Yeti")
- Extension eines RDF-Graphen:
 - Menge aller Aussagen, die in einem RDF-Graphen enthalten sind
 - Abbildung von IP auf $IR \times IR$
 - IP: Menge der Prädikate
 - IR: Menge der Ressourcen

Extension vs. Interpretation



- Die Interpretation bestimmt die Extension eines Graphen
- Bildet den Graph auf eine Extensionsfunktion ab
- Einfachste Interpretation:
 - $\langle s, p, o \rangle \in G \rightarrow \langle s, o \rangle \in I_{\text{EXT}}(p)$
- Diese Interpretation enthält alle Aussagen, die *explizit* im Graph enthalten sind
- Wir sind aber besonders an den *impliziten* Aussagen interessiert!

Extension: schematisch



Angelehnt an: Hitzler et al. (2008): *Semantic Web Grundlagen*.

Extension mit Ableitungsregeln



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- RDFS-Ableitungsregeln bilden eine Interpretationsfunktion
- Erzeugen eine Extension
 - auf Basis bestehender Ressourcen, Literale und Properties
 - zusätzliche Paare der Form $\langle s, o \rangle \in I_{\text{EXT}}(p)$
 - z.B.
 - $\langle \text{Madrid, Stadt} \rangle \in I_{\text{EXT}}(\text{rdf:type})$
 - $\langle \text{Madrid, Spanien} \rangle \in I_{\text{EXT}}(\text{liegt_in})$
- Merke:
 - es entstehen niemals *neue* Ressourcen, Literale, Properties!



Reasoning mit Hilfe von Ableitungsregeln



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Ableitungsregeln bilden eine Interpretationsfunktion
- Entailment: Menge aller Konsequenzen aus einem Graphen
- Einfachster Reasoning-Algorithmus (Forward Chaining):

Gegeben: RDF-Graph G

Menge von Ableitungsregeln R

Entailment $E = G$

Wiederhole

$M := \{ \}$

 Für alle Regeln in R

 Für jede Aussage A in G

 Wende R auf A an

 Wenn Ergebnis nicht in M enthalten

 Füge Ergebnis zu M hinzu

 Füge alle Elemente von M zu E hinzu

bis $M = \{ \}$

Ableitungsregeln für RDF und RDF Schema (1)

ID	Voraussetzung	Konsequenz
rdf1	<code>s p o .</code>	<code>p rdf:type rdf:Property .</code>
rdfs1	<code>s p l .</code> <code>l</code> ist ein Literal	<code>l rdf:type rdfs:Literal .</code>
rdfs2	<code>s p o .</code> <code>p rdfs:domain c .</code>	<code>s rdf:type c .</code>
rdfs3	<code>s p o .</code> <code>p rdfs:range c .</code>	<code>o rdf:type c .</code>
rdfs4a	<code>s p o .</code>	<code>s rdf:type rdfs:Resource .</code>
rdfs4b	<code>s p o .</code> <code>o</code> ist ein URI oder Blank Node	<code>o rdf:type rdfs:Resource .</code>
rdfs5	<code>p1 rdfs:subPropertyOf p2 .</code> <code>p2 rdfs:subPropertyOf P3 .</code>	<code>p1 rdfs:subPropertyOf p3 .</code>
rdfs6	<code>p rdf:type rdf:Property .</code>	<code>p rdfs:subPropertyOf p .</code>

W3C (2004): *RDF Semantics*. <http://www.w3.org/TR/rdf-mt/>

Ableitungsregeln für RDF und RDF Schema (2)



TECHNISCHE
UNIVERSITÄT
DARMSTADT

ID	Voraussetzung	Konsequenz
rdfs7	<code>p1 rdfs:subPropertyOf p2 .</code> <code>s p1 o .</code>	<code>s p2 o .</code>
rdfs8	<code>c rdf:type rdfs:Class .</code>	<code>c rdfs:subClassOf rdfs:Resource .</code>
rdfs9	<code>s rdf:type c1 .</code> <code>c1 rdfs:subClassOf c2 .</code>	<code>s rdf:type c2 .</code>
rdfs10	<code>c rdf:type rdfs:Class .</code>	<code>c rdfs:subClassOf c .</code>
rdfs11	<code>c1 rdfs:subClassOf c2 .</code> <code>c2 rdfs:subClassOf c3 .</code>	<code>c1 rdfs:subClassOf c3 .</code>
rdfs12	<code>p rdf:type rdfs:container-MembershipProperty .</code>	<code>p rdfs:subPropertyOf rdfs:member .</code>
rdfs13	<code>d rdf:type rdfs:Datatype .</code>	<code>d rdfs:subClassOf rdfs:Literal .</code>

W3C (2004): *RDF Semantics*. <http://www.w3.org/TR/rdf-mt/>

- Betrachten wir wieder unser Ausgangsbeispiel
:Madrid :capitalOf :Spain .
(ohne die zusätzliche Ontologie)

	:Madrid :capitalOf :Spain .	(a0)
(a0 + rdf1)	:capitalOf rdf:type rdf:Property .	(a1)
(a1 + rdf1)	rdf:type rdf:type rdf:Property .	(a2)
(a0 + rdfs4a)	:Madrid rdf:type rdfs:Resource .	(a3)
(a0 + rdfs4b)	:Spain rdf:type rdfs:Resource .	(a4)
(a1 + rdfs4a)	rdf:type rdf:type rdfs:Resource .	(a5)

...

▪ Beispiel mit Ontologie:

```
:Employee a rdfs:Class .  
:Employee rdfs:subClassOf :Human .  
:Room a rdfs:Class .  
:hasOffice a rdf:Property .  
:worksIn rdfs:subPropertyOf :hasOffice .  
:hasOffice rdfs:domain :Employee .  
:hasOffice rdfs:range :Room .  
  
:Tim :worksIn :D0815 .
```

Anwendung von Ableitungsregeln



TECHNISCHE
UNIVERSITÄT
DARMSTADT

▪ Beispiel:

```
:Tim :worksIn :D0815 .  
:worksIn rdfs:subPropertyOf :hasOffice .
```

ID	Voraussetzung	Konsequenz
rdfs7	<pre>p1 rdfs:subPropertyOf p2 . s p1 o .</pre>	<pre>s p2 o.</pre>

→ :Tim :hasOffice :D0815 .

Anwendung von Ableitungsregeln



TECHNISCHE
UNIVERSITÄT
DARMSTADT

▪ Beispiel:

```
:Tim :hasOffice :D0815 .  
:hasOffice rdfs:domain :Employee .
```

ID	Voraussetzung	Konsequenz
rdfs2	<pre>s p o . p rdfs:domain c .</pre>	<pre>s rdf:type c .</pre>

→ :Tim rdf:type :Employee .

Anwendung von Ableitungsregeln



TECHNISCHE
UNIVERSITÄT
DARMSTADT

▪ Beispiel:

```
:Tim rdf:type :Employee.  
:Employee rdfs:subClassOf :Human .
```

ID	Voraussetzung	Konsequenz
rdfs9	<pre>s rdf:type c1 . c1 rdfs:subClassOf c2 .</pre>	<pre>s rdf:type c2 .</pre>

→ :Tim rdf:type :Human .

Was passiert bei mehreren (konkurrierenden) Aussagen?



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Beispiel soziales Netzwerk:

```
:knows rdfs:range :People .  
:knows rdfs:range :MemberOfSocialNetwork .
```

- Was soll die Semantik davon sein?

- Jeder, den jemand kennt,
ist ein Mensch *und* ein Mitglied des sozialen Netzwerks
- Jeder, den jemand kennt,
ist ein Mensch *oder* ein Mitglied des sozialen Netzwerks

Die Ableitungsregeln werden es uns verraten...



TECHNISCHE
UNIVERSITÄT
DARMSTADT

```
      :knows rdfs:range :Human .                (a0)
      :knows rdfs:range :MemberOfSocialNetwork . (a1)
      :Peter :knows :Stephen .                  (a2)
(rdfs3+a0+a2) :Stephen rdf:type :Human .        (a3)
(rdfs3+a1+a2) :Stephen rdf:type :MemberOfSocialNetwork . (a4)
      ...
```

- Diese Schlusskette funktioniert für beliebige Objekte
 - diese sind stets in beiden Klassen enthalten
 - es gilt also die Semantik mit der Schnittmenge!

Was haben wir jetzt gewonnen?



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Betrachten wir folgenden Satz:
- "Madrid ist die Hauptstadt von Spanien."
- Aussagen, die wir erhalten können:
 - "Madrid ist die Hauptstadt von Spanien." ✓
 - "Spanien ist ein Land." ✓
 - "Madrid ist eine Stadt." ✓
 - "Madrid liegt in Spanien." ✓
 - "Barcelona ist nicht die Hauptstadt von Spanien." ✗
 - "Madrid ist nicht die Hauptstadt von Frankreich." ✗
 - "Madrid ist kein Land." ✗
 - ...

Was wir (bis jetzt) noch nicht können

- "Jedes Land hat nur genau eine Hauptstadt"
 - Kardinalitäten von Properties
- "Jede Stadt kann nur Hauptstadt von genau einem Land sein"
 - Funktionale Properties
- "Eine Stadt ist nicht gleichzeitig ein Land."
 - Disjunkte Klassen
- ...
- hier brauchen wir mächtigere Sprachmittel als RDF Schema!

Was wir (bis jetzt) noch nicht können

- "Jedes Land hat nur genau eine Hauptstadt"
 - d.h., "Ein Land kann *nicht* zwei oder mehr Hauptstädte haben."
- "Jede Stadt kann nur Hauptstadt von genau einem Land sein"
 - d.h., "Eine Stadt kann *nicht* Hauptstadt von zwei *verschiedenen* Ländern sein."
- "Eine Stadt ist *nicht* gleichzeitig ein Land."

Was wir (bis jetzt) noch nicht können

- Merke: in RDF und RDFS gibt es *keine* Verneinung
- Damit kann man auch keine Widersprüche erzeugen
 - das macht das Reasoning schön einfach
 - schränkt aber auch die Mächtigkeit ein
 - Beispiel:
 - Säugetiere legen keine Eier
 - ein Pinguin legt Eier
 - ein Pinguin ist kein Säugetier
- Wir werden noch Formalismen kennenlernen,
die auch Verneinung unterstützen

Was wir (bis jetzt) noch nicht können



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Die Freiheit von Verneinung passt gut zum AAA-Prinzip
 - Anybody can say anything about anything
- und zur Open World Assumption
- Neu hinkommende Aussagen fügen sich immer in das vorhandene Wissen ein
- Dieses Prinzip heißt "Monotonie"

Was wir (bis jetzt) noch nicht können



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Kurt Gödel (1906-1978)
- Logische Systeme sind entweder
 - nicht besonders ausdrucksstark oder
 - nicht widerspruchsfrei
- RDF Schema gehört zur ersten Klasse



Was wir (bis jetzt) noch nicht können

- Jim Hendler (*1957)
- "A little semantics goes a long way."



Moment mal...



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- "Damit kann man auch keine Widersprüche erzeugen"
- Und was ist mit
 - :Peter a :Baby .
 - :Peter a :Adult .
- Ist das denn etwa kein Widerspruch?!
- Für uns Menschen schon
- aber ein Rechner weiß das nicht
 - Non-unique name assumption!

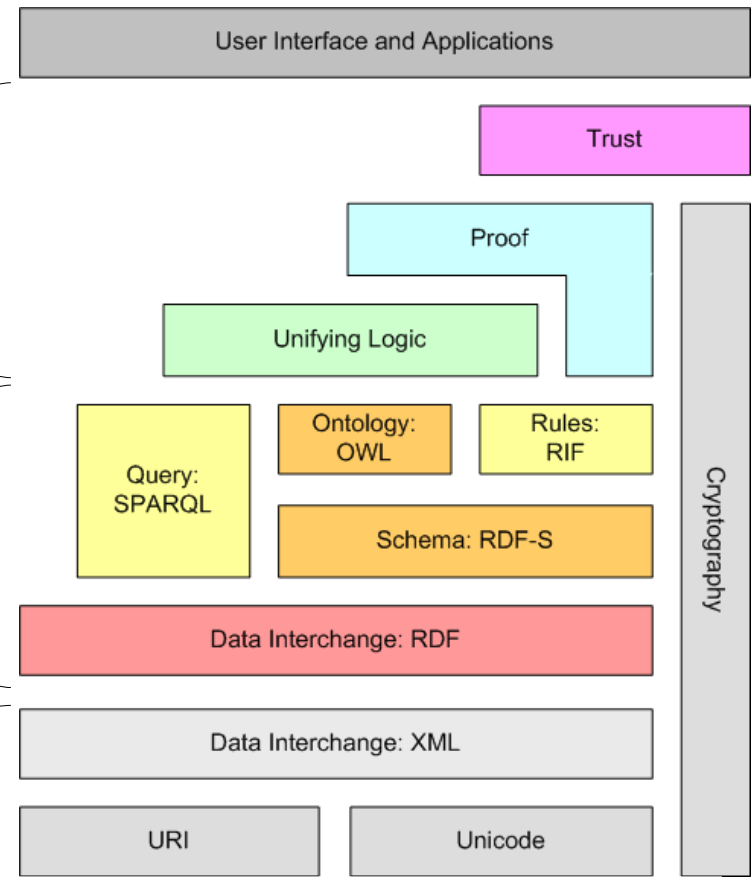
Semantic Web – Aufbau



here be dragons...

Semantic-Web-
Technologie
(Fokus der
Vorlesung)

Technische
Grundlagen



Berners-Lee (2009): *Semantic Web and Linked Data*
<http://www.w3.org/2009/Talks/0120-campus-party-tbl/>

Vorlesung Semantic Web



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Vorlesung im Wintersemester 2011/2012

Dr. Heiko Paulheim

Fachgebiet Knowledge Engineering