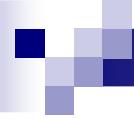


Moderne Web-Technologien

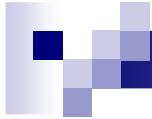
Die Vision und Zukunft des Semantic Web

Alan Akbik, Julius Müller, Stefan Lenz



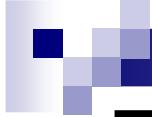
Agenda

- Einführung Semantic Web
 - Web von heute
 - Web von morgen (?)
 - Geschichte des Semantic Web
- Technologien
 - RDF/ RDF-Schema
 - Ontologien (OWL)
- Fazit
- Fragen / Diskussion



Web heute

- HTML
 - keine Trennung von Inhalt und Präsentation
- XML
 - Inhalt und Präsentation getrennt
- ABER: **Bedeutung des Inhalts?**



Problemstellung

- Inhaltliche Bedeutung wird nur abgeschätzt
- Suchmaschinen können nicht unterscheiden zwischen

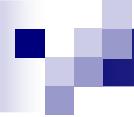


Aus: www.ai.mit.edu

und



Aus: cars.bayaw.com



Information Retrieval

■ Indexierung

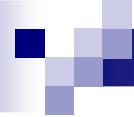
- manuell (der Abstract eines Dokuments)
- automatisch

■ Google PageRank

- Links die auf ein Dokument zeigen

■ Computerlinguistik

- Schlüsselwörter ermitteln
- wenig erfolgreich



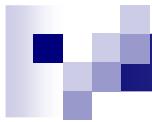
Lesbarkeit

■ Web heute:

- Seiteninhalte werden von Menschen gelesen
- Suchmaschinen können Inhalte nur indirekt ermitteln

■ Web morgen (?):

- Seiteninhalte von Maschinen lesbar
- Suchmaschinen “verstehen” Inhalte
- logisches Folgern



Der Visionär

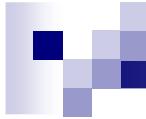
- Sir Timothy Berners-Lee
- Erfinder des WWW



Aus: www.w3.org

"If HTML and the Web made all the online documents look like one huge **book**, RDF, schema, and inference languages will make all the data in the world look like one huge **database**"

Tim Berners-Lee, Weaving the Web, 1999

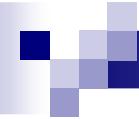


Vision (1)

- Das Telefon klingelt
- alle Geräte im *selben Raum* verringern ihre *Lautstärke*
- Problemstellung: Abendplanung!



Aus: www.unitedmedia.com



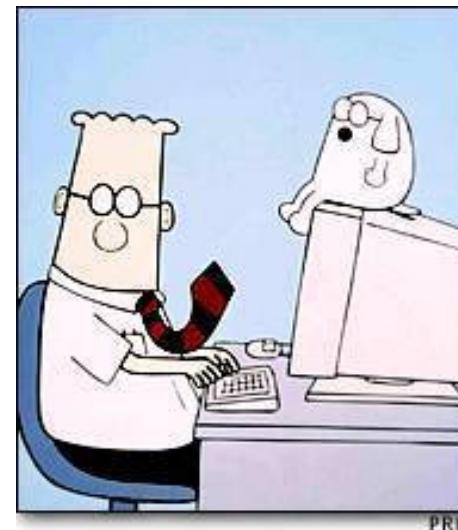
Vision (2)

■ Wünsche:

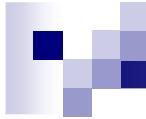
- Kino mit *originalsprachlichen* Filmen
- Kneipe in *Laufreichweite*
- Nachtbusse*
- Etc...

■ Heute:

- Mühelige Recherche



Aus: marc-blog.kataplop.net

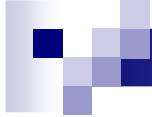


Vision (3)

- Geräte kommunizieren miteinander
- Differenzierte Web-Anfragen
- Agenten
 - autonomes Sammeln von Informationen
- Futuristisch?

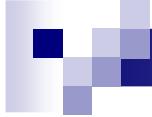


Aus: www.appleblossomart.net



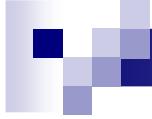
Semantic Web

- Erweiterung des WWW
 - Dokumente für Maschinen „verstehbar“
 - Maschine-Maschine Kommunikation
- Internet als Datenbank („Web of Data“)
 - Dezentral
- Informationen erhalten eine definierte *Bedeutung*



Herausforderungen

- Wissensrepräsentation
 - Formalisierung von inhaltlicher Bedeutung
 - Homonyme („Teekesselchen“)
 - Synonyme
- Inferenzregeln
 - neues Wissen aus bestehendem erschließen



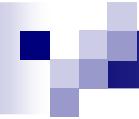
Linguistik / Philosophie

■ Taxonomien

- Hierarchische Klassifikation
- Is-a Relationen
 - „*Frucht* ist ein *Nahrungsmittel*“
 - „*Orange* ist eine *Frucht*“

■ Ontologien

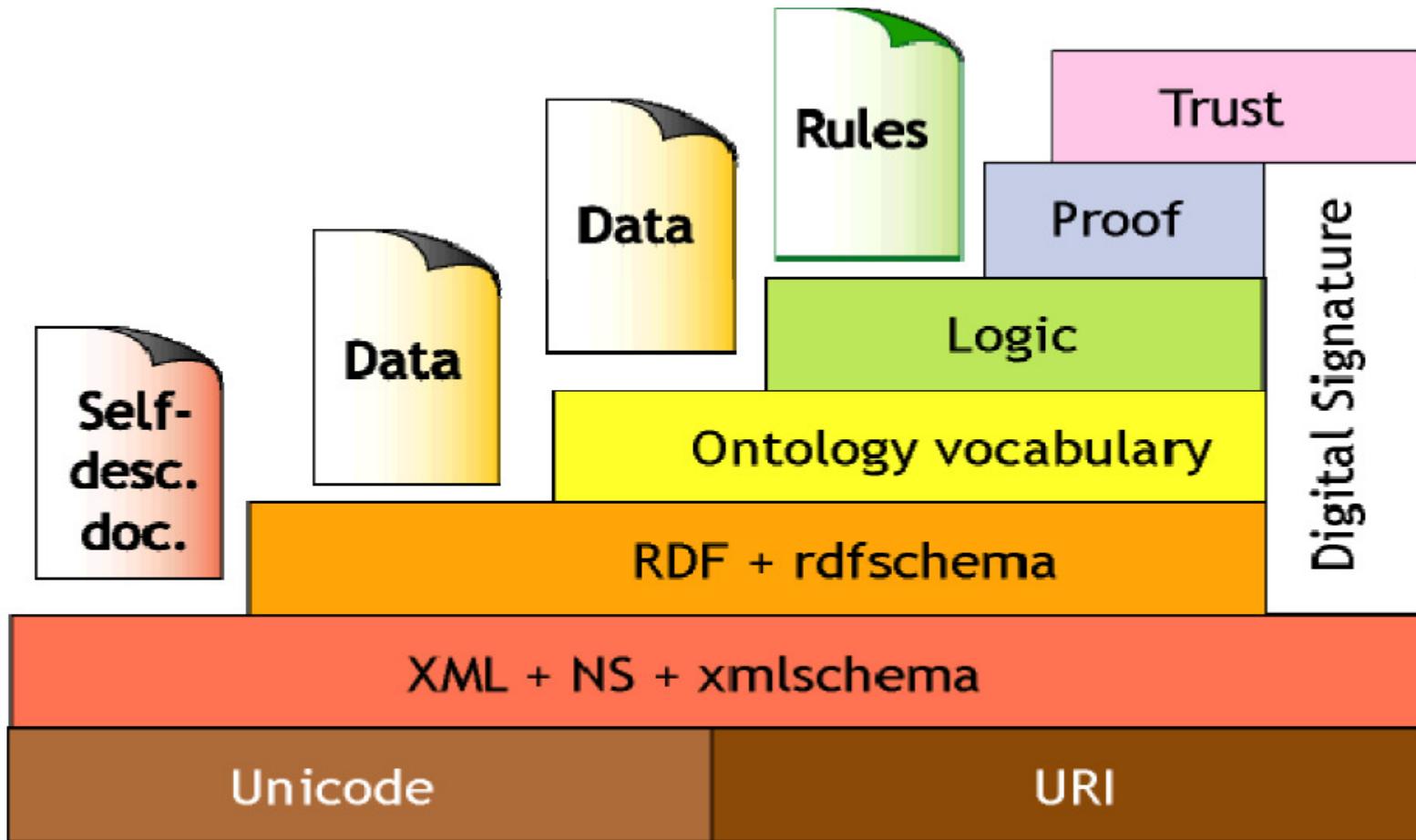
- Seinszusammenhang
- Logische Relationen



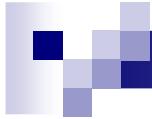
Geschichte

- 1994:
 - Berners-Lee beschreibt seine Vision auf WWW-Konferenz
- 1999:
 - RDF wird W3C-Recommendation
 - *Weaving the Web*
- 2001:
 - *The Semantic Web*, Scientific American
- 2004:
 - RDF-Schema wird W3C-Recommendation
 - OWL wird W3C-Recommendation

Semantic Web Stack



Aus: Tim Berners-Lee: <http://www.w3.org/2000/Talks/1206-xml2k-tbl/slides10-0.html>



Technologien (1)

- XML

- erlaubt das Erstellen von strukturierten Dokumenten

- XML-Schema

- legt syntaktische Regeln für XML-Dokumente fest

- RDF

- bietet Semantik für ein einfaches Datenmodell (Ressourcen und ihre Beziehungen)



Technologien (2)

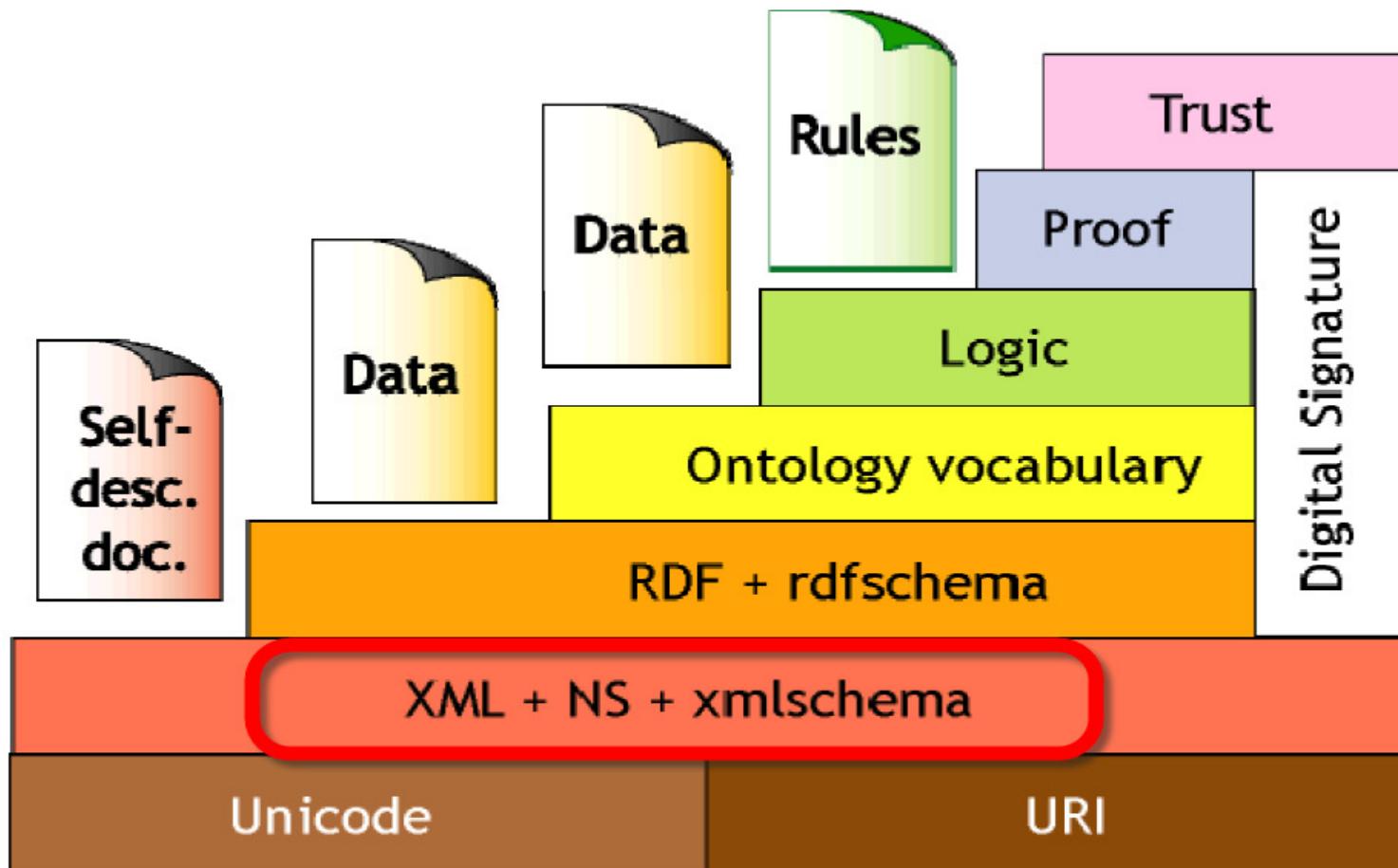
■ RDF-Schema

- beschreibt Klassen und Eigenschaften aus RDF
- Semantik für Generalisierungshierarchien (IS-A)

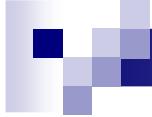
■ OWL

- weitergehende Beschreibungsmöglichkeiten, wie u.a.
 - Disjunktheit
 - Kardinalität
 - Gleichheit/ Symmetrie
 - Typisierung von Eigenschaften
 - Enumerationen

Semantic Web Stack



Aus: Tim Berners-Lee: <http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>

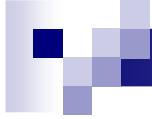


XML

■ Was bietet XML ?

- Trennung zwischen Struktur und Inhalt
- Rahmen für Informationen
- explizite Inhaltsdarstellung
- maschinenlesbar

■ Warum ist XML nicht ausreichend ?



XML

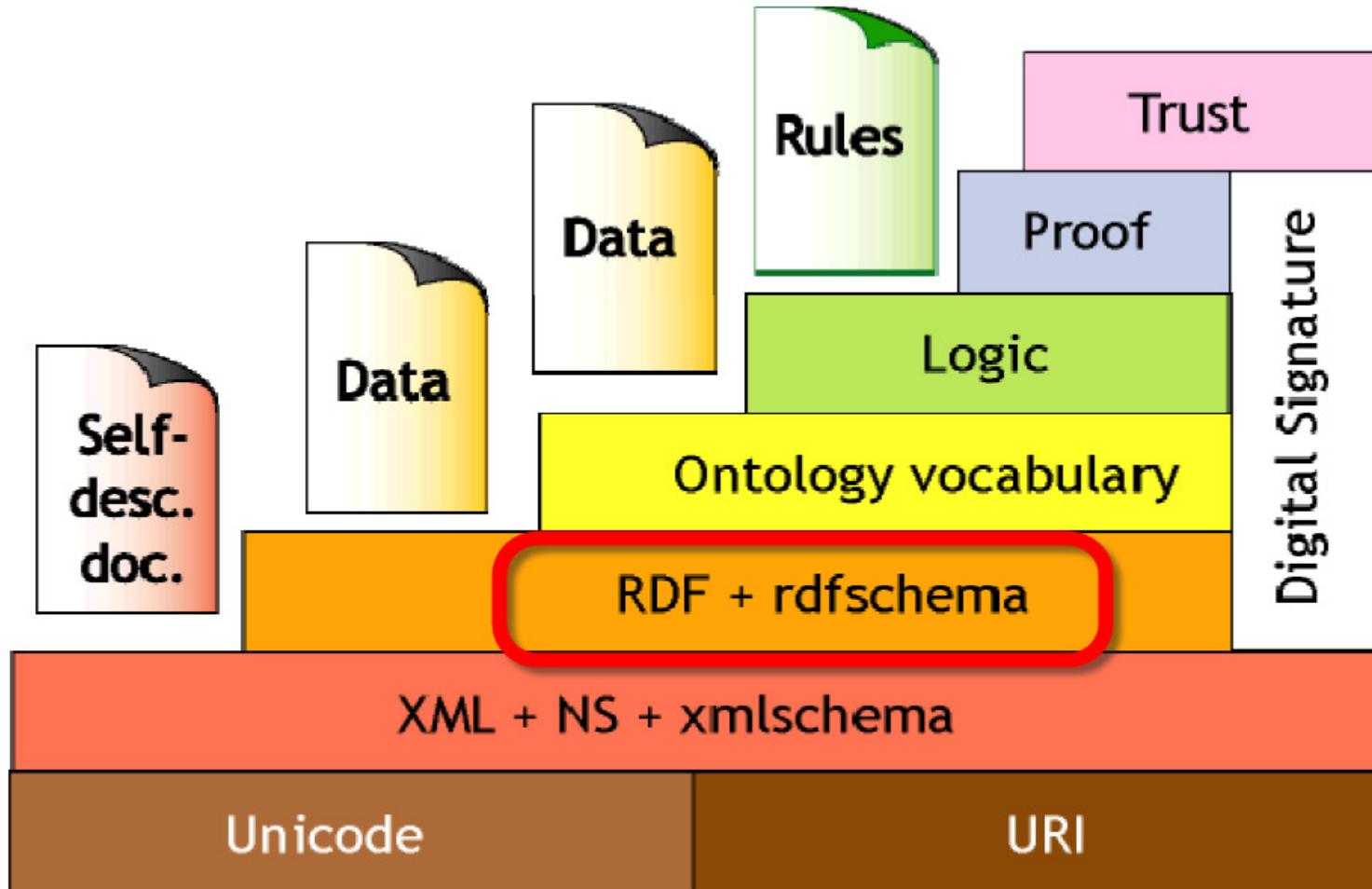
■ Was bietet XML ?

- Trennung zwischen Struktur und Inhalt
- Rahmen für Informationen
- explizite Inhaltsdarstellung
- maschinenlesbar

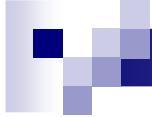
■ Warum ist XML nicht ausreichend ?

- Bezug zwischen Objekten

Semantic Web Stack

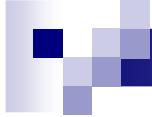


Aus: Tim Berners-Lee: <http://www.w3.org/2000/Talks/1206-xml2k-tbl/slides10-0.html>



RDF

- Resource Description Framework
- Menge von Aussagen
- eine Aussage ist ein Tripel bestehend aus Subjekt, Prädikat und Objekt



RDF

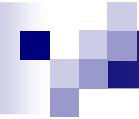
- Resource Description Framework
- Menge von Aussagen
- eine Aussage ist ein Tripel bestehend aus Subjekt, Prädikat und Objekt

- umgangssprachlich formulierte Information
 - ein Hersteller erstellt ein Produkt



RDF (2)

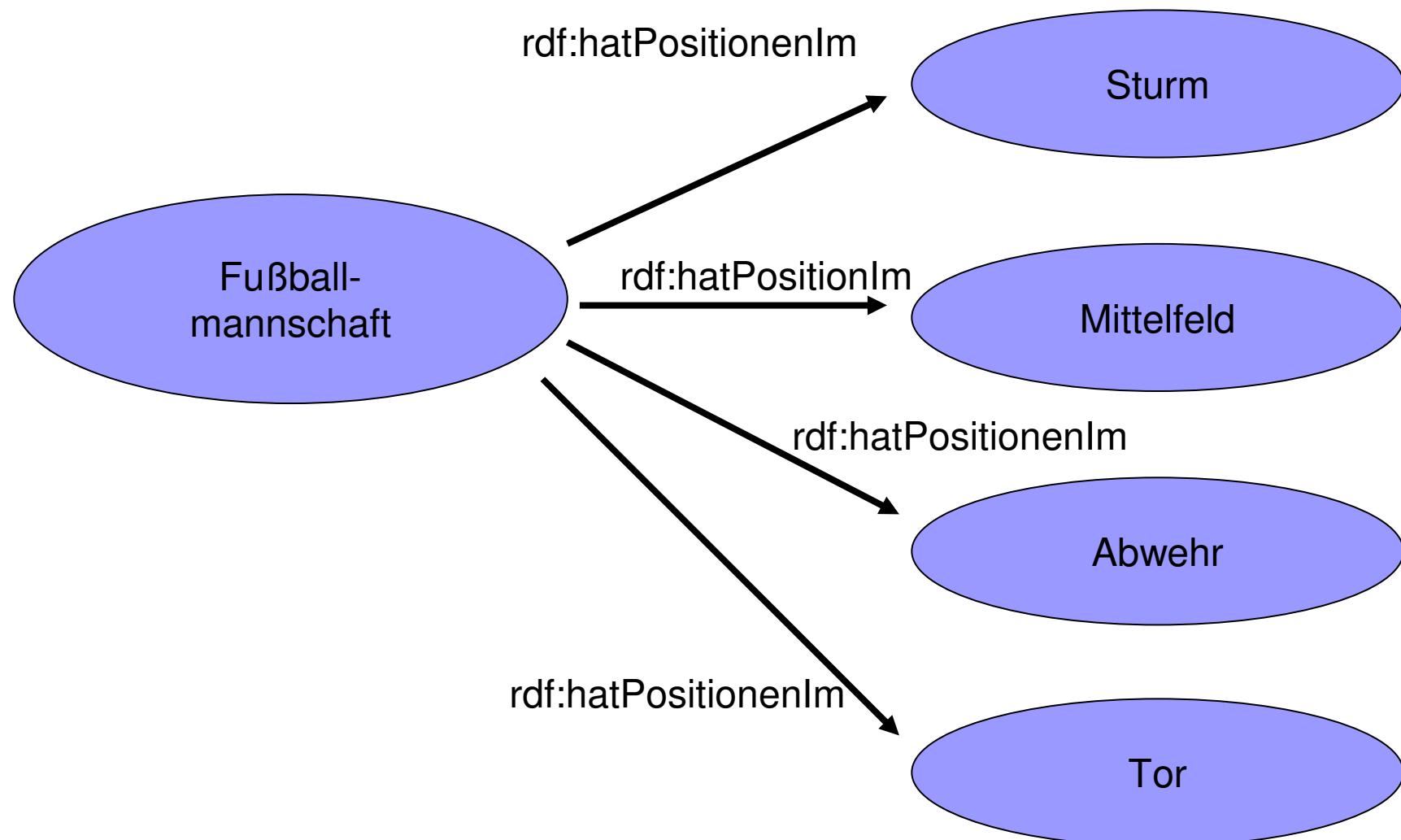
```
<?xml version="1.0"?>
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns"
      xmlns:s="http://meinSchema.de/schema/">
  <Description about="http://www.Hersteller.de">
    <s:Erstellt
      rdf:resource="http://www.Hersteller.de/Produkt"/>
    <s:Seit>1909</s:Seit>
    ...
  </Description>
  <!-- weitere Description-Elemente -->
</RDF>
```

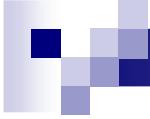


Datenmodell

- Aussagen
 - Tripel (S,P,O)
- Beziehungen
 - verknüpfen Objekte paarweise
- Objekte
 - URI
 - Aussagen sind wieder Objekte
- gerichteter Graph
 - Knoten: Objekt
 - Kanten: Beziehungen

Modellierung von 1:N-Beziehungen

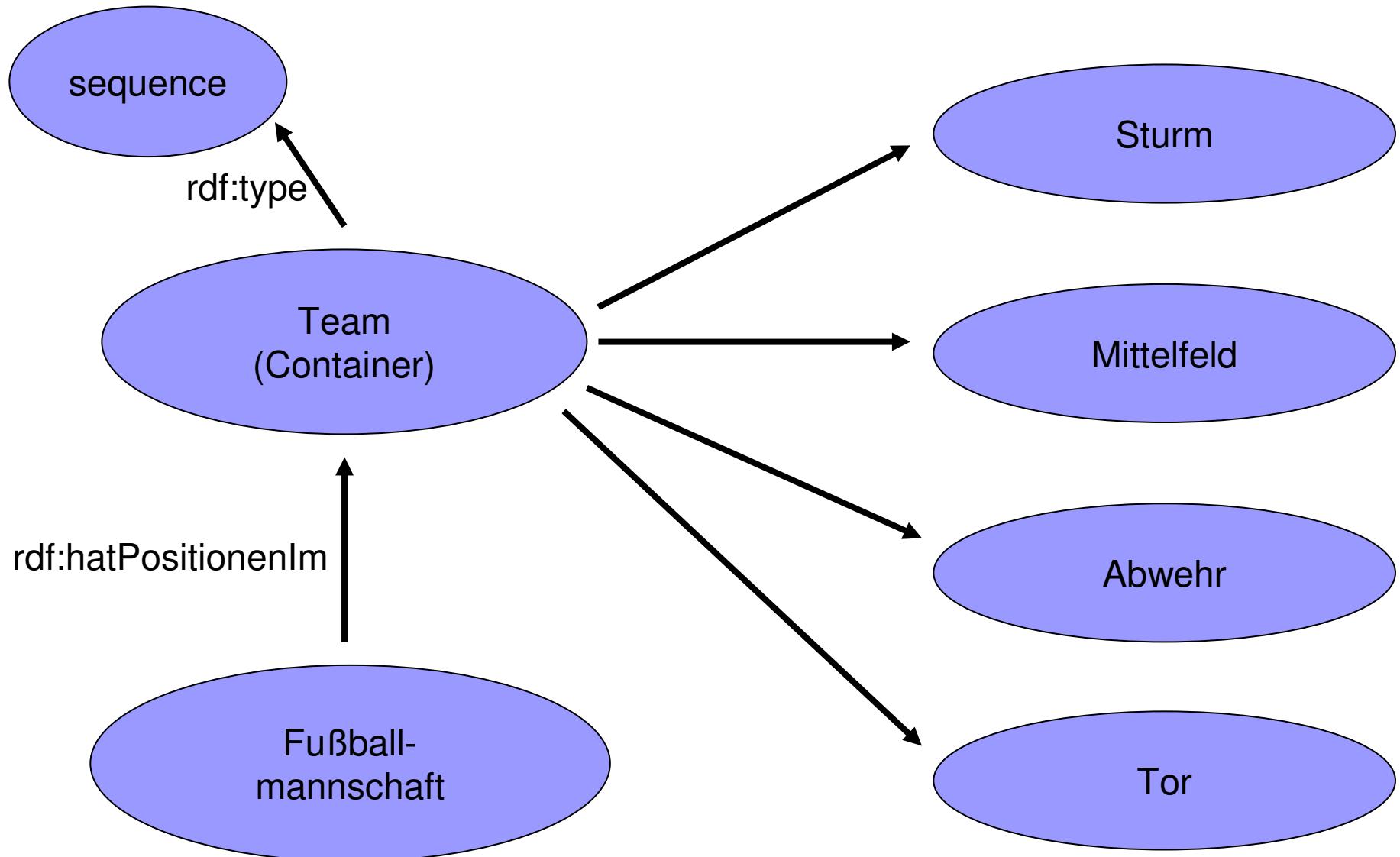


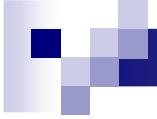


Repertoire

- Containertypen (offen)
 - bag (ungeordnet)
 - sequence (geordnet)
 - alternative (eindeutiger Wert)
- Collection (geschlossen)
 - rekursiv
 - first
 - rest
 - nil
- Blank Nodes
 - unbekannte Ressourcen

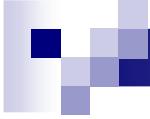
Modellierung von 1:N-Beziehungen





Modellierung von 1:N-Beziehungen

```
<RDF>
<Description
  about="http://www.mannschaft.de">
  <s:hatPositionenIm>
    <Seq ID="hatPositionenIm">
      <li resource="http://www.mannschaft.de/Sturm"/>
      <li resource="http://www.mannschaft.de/Mittelfeld"/>
      <li resource="http://www.mannschaft.de/Abwehr"/>
      <li resource="http://www.mannschaft.de/Tor"/>
    </Seq>
  </s:Fussball >
</Description>
</RDF>
```

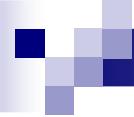


RDF (3)

- Was bietet RDF ?

- Menge an Relationen
 - minimale Semantik
 - Aussagen über Aussagen
 - definiert einfaches Datenmodell
 - formalisiertes Wissen verteilt abgelegt

- Warum ist RDF nicht ausreichend?



RDF (3)

■ Was bietet RDF ?

- Menge an Relationen
- minimale Semantik
- Aussagen über Aussagen
- definiert einfaches Datenmodell
- formalisiertes Wissen verteilt abgelegt

■ Warum ist RDF nicht ausreichend?

- Hierarchie
- komplexere Sachverhalte
- Vergleiche zwischen Objekten

RDF-Schema (1)

Klassen & Unterklassen

Generalisierung

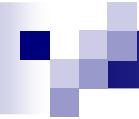
Existenzquantor

RDF-Schema (Ausschnitt)

```
<rdfs:Class rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag">
  <rdfs:isDefinedBy rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
  <rdfs:label>Bag</rdfs:label>
  <rdfs:comment>The class of unordered containers.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Container"/>
</rdfs:Class>

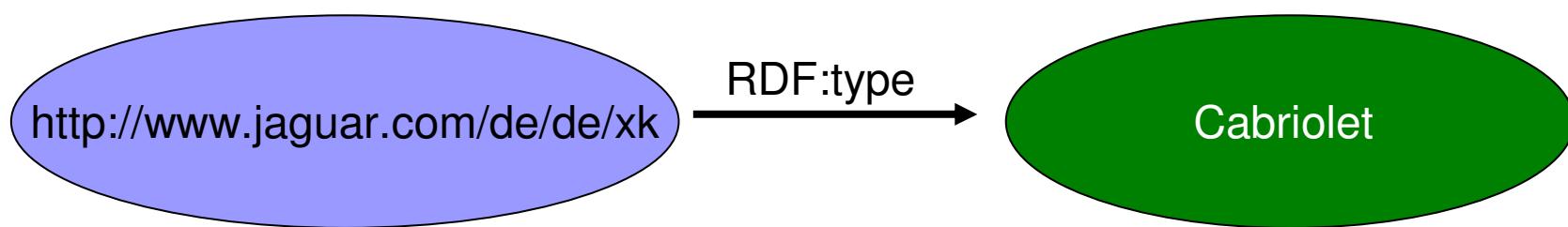
<rdfs:Class rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq">
  <rdfs:isDefinedBy rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
  <rdfs:label>Seq</rdfs:label>
  <rdfs:comment>The class of ordered containers.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Container"/>
</rdfs:Class>

<rdfs:Class rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Alt">
  <rdfs:isDefinedBy rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
  <rdfs:label>Alt</rdfs:label>
  <rdfs:comment>The class of containers of alternatives.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Container"/>
</rdfs:Class>
```



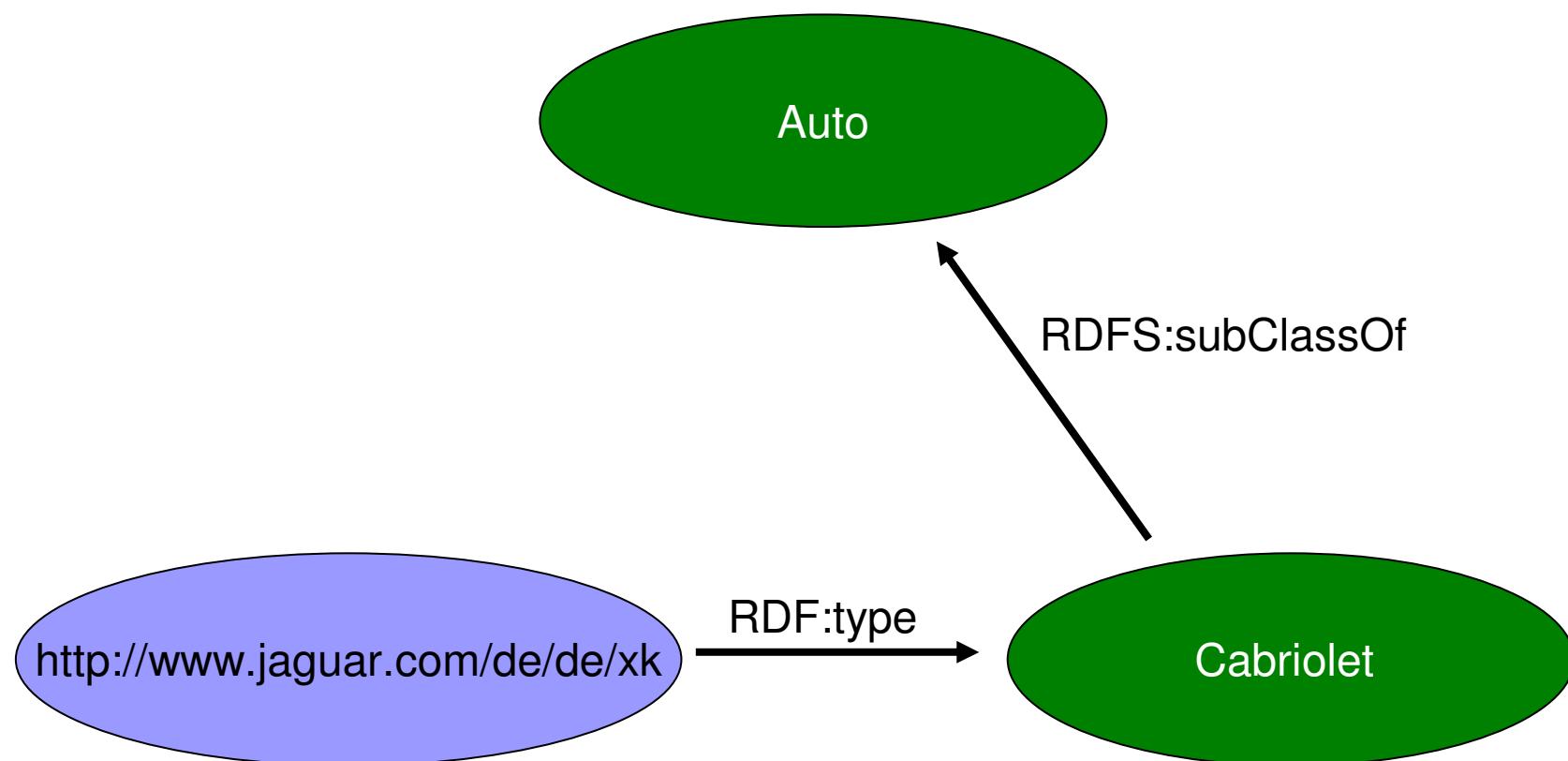
RDF-Schema (2)

Inferenz



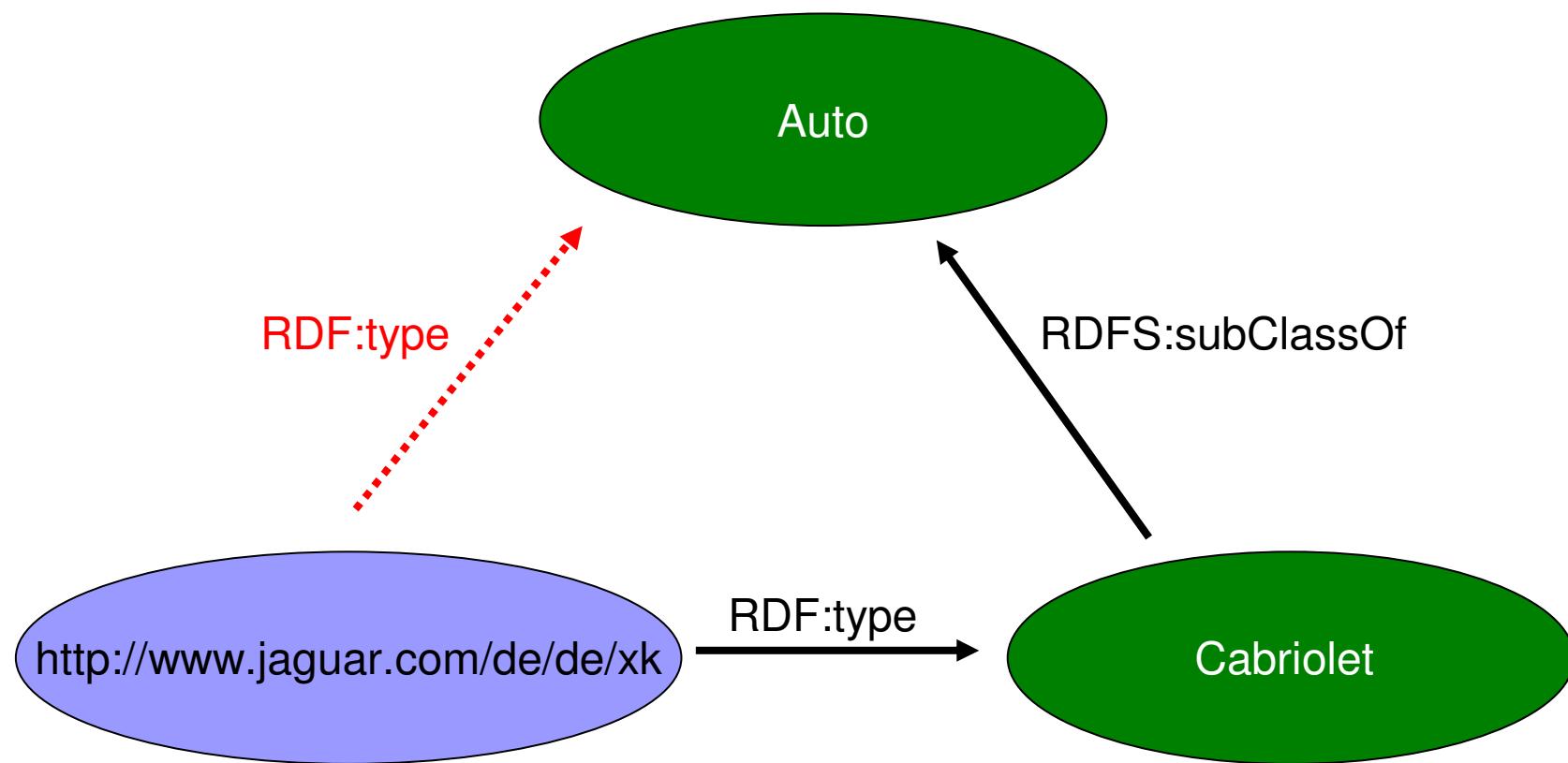
RDF-Schema (2)

Inferenz



RDF-Schema (2)

Inferenz

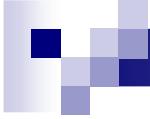


RDF-Schema (3)

- Eigenschaften
 - rdfs:label
 - rdfs:comment
 - rdfs:type
- Spezialisierung
 - rdfs:subClassOf
 - rdfs:subPropertyOf
- Wertebereich
 - rdfs:range
- Herkunftsbereiche
 - rdfs:domain
- Verweise
 - rdfs:seeAlso
 - rdfs:isDefinedBy

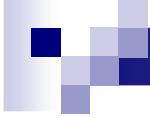
Vererbung

```
<rdf:RDF xmlns:rdf= "http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">  
    <rdf:Description rdf:ID="mensch">  
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-  
            schema#Class"/>  
        <rdfs:subClassOf rdf:resource="#lebewesen"/>  
    </rdf:Description>  
    <rdf:Description rdf:ID="mann">  
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-  
            schema#Class"/>  
        <rdfs:subClassOf rdf:resource="#mensch"/>  
    </rdf:Description>  
    <rdf:Description rdf:ID="frau">  
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-  
            schema#Class"/>  
        <rdfs:subClassOf rdf:resource="#mensch"/>  
    </rdf:Description>  
</rdf:RDF>
```



Domain & Range

```
<rdf:Property rdf:ID="URL">  
  <rdfs:domain rdf:resource="„Internetseite"/>  
  <rdfs:range  
    rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>  
</rdf:Property>
```

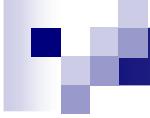


RDF-Schema

■ Was bietet RDF-S ?

- Eigenschaften
- Spezialisierung
- Wertebereich
- Herkunftsbereiche
- Verweise

■ Warum ist RDF-S nicht ausreichend ?



RDF-Schema

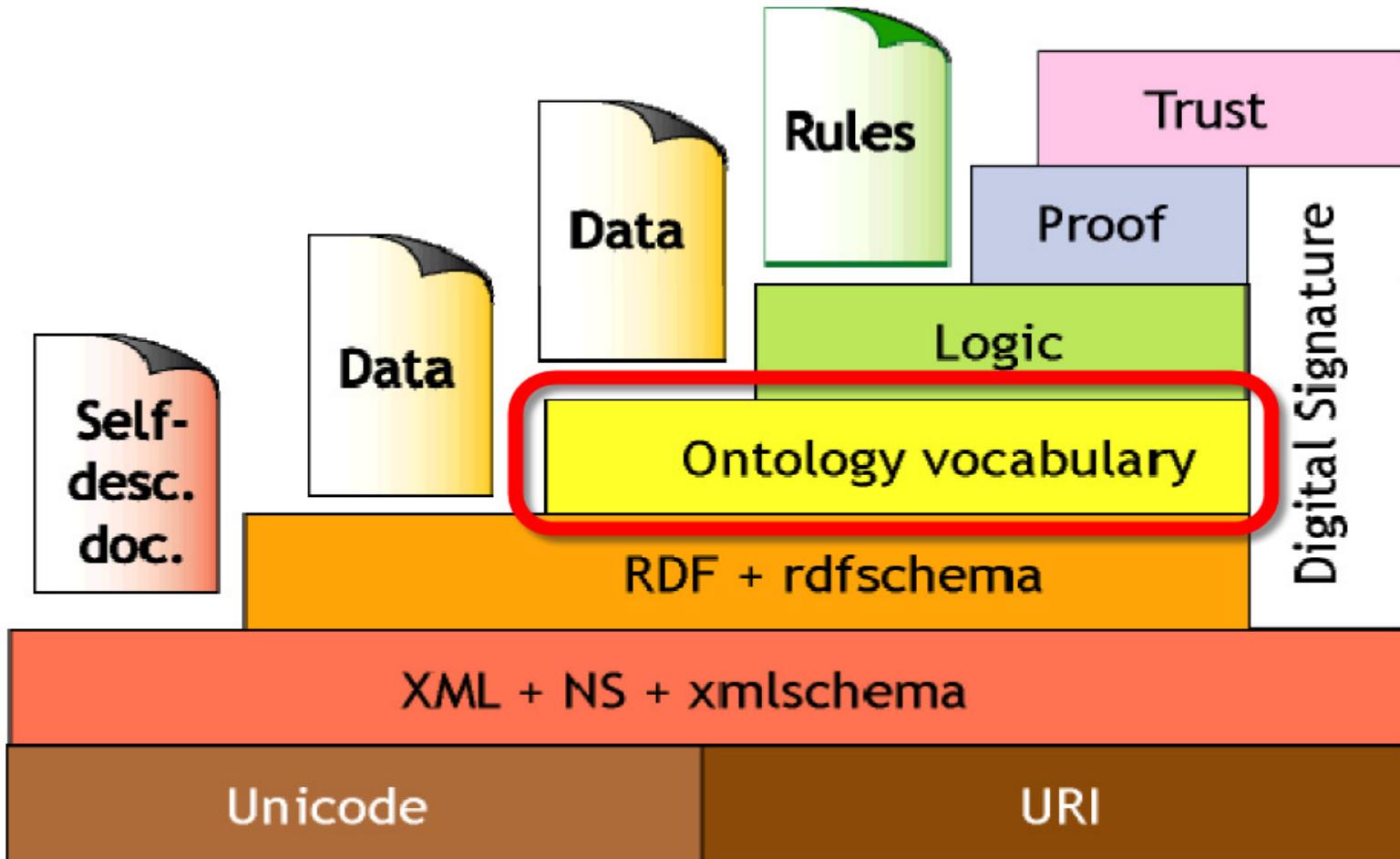
■ Was bietet RDF-S ?

- Eigenschaften
- Spezialisierung
- Wertebereich
- Herkunftsbereiche
- Verweise

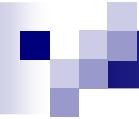
■ Warum ist RDF-S nicht ausreichend ?

- Disjunktheit, Kardinalität, Gleichheit/ Symmetrie
- Typisierung von Eigenschaften

Semantic Web Stack

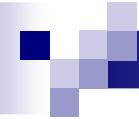


Aus: Tim Berners-Lee: <http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>



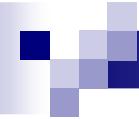
Ontologie

- Begriff ursprünglich aus der Philosophie
 - beschäftigt sich mit dem Wesen des Seins
 - erklärt die Beschaffenheit der Welt
- In der Informatik versteht man unter Ontologie die Beschreibung eines Wissensgebiets
 - Sie beschreibt Dinge (Klassen) aus der Domäne
 - die Beziehungen die zwischen Dingen existieren können
 - die Eigenschaften (Attribute) die solche Dinge haben können



Ziele von Ontologien

- Ziel ist die Definition einer gemeinsamen Sprache als Grundlage von Wissensaustausch
- Automatisches Schließen
 - Anwendung von Regeln auf Wissen
 - Inferieren neuen Wissens
- Repräsentation von Wissen
 - Explizite Notation von Wissen
 - Wiederverwendung

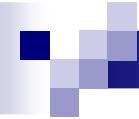


Klassifikation nach Gültigkeitsbereich

- Top Level Ontologien
 - Allgemeine domänenüberschreitende Konzepte
 - Person, Ort, Organisation
- Domänenontologien
 - Auf bestimmtes Wissensgebiet bezogen
 - Seminar, Dozent

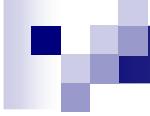
Klassifikation nach innerer Struktur

- Kontrolliertes Vokabular
 - Liste definierter Schlagwörter (Glossar)
 - zur Klassifizierung oder Verschlagwortung
 - ohne inhaltliche Bezüge
- Thesaurus
 - kontrolliertes Vokabular, das Synonyme, Bezüge sowie Ober- und Unterbegriffe verwaltet
 - Ziel ist das Auffinden von Begriffen
 - „Ontology: A thesaurus gone mad“
- Taxonomie
 - Streng hierarchische Klassifizierung (aus der Biologie)



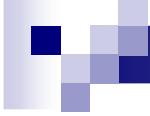
OWL

- Web Ontology Language
- beschreibt die Bedeutung von Begriffen und die Beziehungen zwischen diesen Begriffen
- Ausdrucksstärker als RDF/ RDF-S
- Weiterentwicklung von DAML+OIL
- W3C Recommendation seit 2004
- für M2M Kommunikation



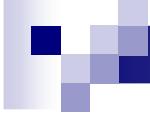
OWL Sprachebenen

- OWL besteht aus 3 Untersprachen mit steigender Ausdrucksfähigkeit:
 - OWL Lite
 - OWL DL
 - OWL Full
- OWL Lite
 - einfache Klassifizierungshierarchie und Einschränkungen (z.B.: Kardinalität ist 1)
 - geringere Komplexität als OWL DL
 - soll Werkzeugherstellung erleichtern
 - leichte Migration anderer (bestehender) Taxonomien (z.B. Thesasuren)



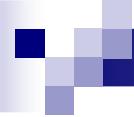
OWL Sprachebenen (2)

- OWL DL (Description Logic)
 - Maximale Ausdrucksfähigkeit unter Beibehaltung von Berechen- und Entscheidbarkeit
 - Voller Sprachumfang mit gewissen Einschränkungen
 - z.B. ist zwar Mehrfachvererbung möglich aber eine Klasse kann nicht Instanz einer anderen sein
- OWL Full
 - Maximale Ausdrucksfähigkeit und syntaktische Freiheit
 - z.B. kann eine Menge von Individuen gleichzeitig selber ein Individuum sein
 - keine garantiierte Berechenbarkeit



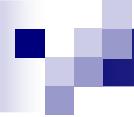
OWL Sprachkonstrukte

- Zunächst Beschreibung der Konstrukte aus OWL Lite
 - Klassenmodell
 - Gleichheit/ Ungleichheit
 - Eigenschaftsmerkmale
 - Eigenschaftsbeschränkungen
 - Sonstige Sprachkonstrukte
- Anschließend Beschreibung der in OWL DL/ Full eingeführten Erweiterungen



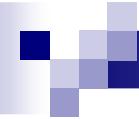
Klassenmodell

- Class (Thing, Nothing)
 - gruppiert Individuen die Eigenschaften teilen
 - z.B.: Debbie und Frank sind Mitglieder der Klasse Person
 - Thing
 - Superklasse aller OWL-Klassen
 - Nothing
 - hat keine Instanzen
 - Unterklasse von allen OWL-Klassen
- Individual
 - Individuen sind Instanzen von Klassen
 - Z.B. Debbie ist ein Individuum der Klasse Person



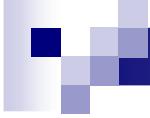
Gleichheit

- **equivalentClass**
 - zeichnet 2 Klassen als äquivalent aus
 - z.B. Auto und Automobil
- **equivalentProperty**
 - zeichnet 2 Eigenschaften als äquivalent aus
 - z.B.: hatChef und hatVorgesetzten
- **sameAs**
 - zeichnet 2 Individuen als gleich aus
 - z.B.: Frank und FrankMüller



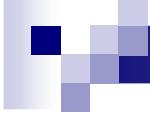
Ungleichheit

- `differentFrom`
 - zeichnet zwei Individuen als unterschiedlich aus
 - z.B.: Frank ist verschieden von Debbie
 - wichtig, da in OWL nicht davon ausgegangen wird, dass Individuen mit unterschiedlichen Namen sich unterscheiden
- `AllDifferent`
 - zeichnet eine Menge von Individuen als unterschiedlich aus



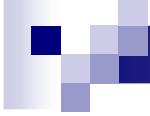
Eigenschaftsmerkmale

- inverseOf
 - z.B.: hatElternteil und hatKind
- TransitiveProperty
 - z.B.: hatVorfahr, liegtIn
- SymmetricProperty
 - z.B.: hatBruder, NachbarVon
- FunctionalProperty (eindeutiger Wert)
 - z.B.: hatGeburtstag
 - äquivalent zu Kardinalität 0 oder 1
- InverseFunctionalProperty
 - z.B.: hatAusweisNummer und IstAusweisNrVon



Eigenschaftsbeschränkungen

- Beschränkungen gelten immer nur lokal
- allValuesFrom
 - z.B.: hatTochter wird beschränkt durch Werte aus Klasse Frau
 - Aber bei der Klasse Katze könnte die selbe Eigenschaft auf Werte aus der Klasse Katze beschränkt werden
- someValuesFrom
 - Wenigstens ein Wert für eine durch someValuesFrom eingeschränkte Eigenschaft muss aus einer bestimmten Klasse kommen
 - z.B.: ein gültiges SemanticWebPaper muss wenigstens ein Schlüsselwort vom Typ SemanticWebThema haben
- minCardinality, maxCardinality, cardinality
 - begrenzt auf die Werte 0 oder 1

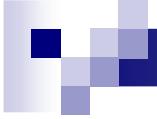


Sonstige Sprachkonstrukte

- Schnitt von Klassen
 - intersectionOf
 - z.B.: Definition der Klasse Rotein als Schnitt der Klassen Wein und der „Dinge mit roter Färbung“
- Datentypen
 - im wesentlichen die XML-Schema Datentypen
- Header-Informationen
 - Namensräume, Einschluss anderer Ontologien
- Annotationen
 - Kommentare etc
- Versionierung

Erweiterungen in OWL DL/ Full

- **oneOf**
 - Aufzählung aller Individuen einer Klasse
 - z.B.: Beschreibung der Klasse Wochentag durch Aufzählung ihrer Individuen (Montag, Dienstag, ...)
- **hasValue**
 - erzwingt, dass der Wert einer Eigenschaft ein bestimmtes Individuum ist
 - z.B.: Mitglieder der Klasse Franzosen definiert als die Personen die Frankreich als Wert der hatNationalität-Eigenschaft haben
- **disjointWith**
 - Klassen als disjunkt kennzeichnen
 - z.B.: Mann und Frau



Erweiterungen in OWL DL/ Full

- unionOf, complementOf, intersectionOf
 - Boolesche Kombinationen von Klassen und Einschränkungen
 - Klasse KinderUndJugendliche als Vereinigung der Klassen Kinder und Jugendliche
 - Mit Hilfe von complementOf kann festgelegt werden, dass Kinder *keine* Erwachsenen sind - sondern eine Teilmenge (!) des Komplements
- minCardinality, maxCardinality, cardinality
 - Restriktion auf maximalen Wert 1 fällt weg

Beispiel (1)

```
<owl:Class rdf:ID="Wine">
    <rdfs:subClassOf rdf:resource="&food;PotableLiquid" />
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasColor" />
            <owl:cardinality
                rdf:datatype="&xsd;nonNegativeInteger">
                1
            </owl:cardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
    ...
</owl:Class>
```

Beispiel (2)

```
<owl:Class rdf:ID="WineColor">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#White" />
    <owl:Thing rdf:about="#Rose" />
    <owl:Thing rdf:about="#Red" />
  </owl:oneOf>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasColor">
  <rdf:type rdf:resource="&owl;FunctionalProperty" />
  ...
  <rdfs:domain rdf:resource="#Wine" />
  <rdfs:range rdf:resource="#WineColor" />
</owl:ObjectProperty>
```

Beispiel (3)

```
<WineColor rdf:ID="Red" />  
<WineColor rdf:ID="Rose" />  
<WineColor rdf:ID="White" />  
  
<owl:AllDifferent>  
  <owl:distinctMembers rdf:parseType="Collection">  
    <vin:WineColor rdf:about="#Red" />  
    <vin:WineColor rdf:about="#White" />  
    <vin:WineColor rdf:about="#Rose" />  
  </owl:distinctMembers>  
</owl:AllDifferent>
```

Beispiel (4)

```
<owl:Class rdf:ID="CabernetSauvignon">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Wine" />
    <owl:Restriction>
      <owl:onProperty rdf:resource="#madeFromGrape" />
      <owl:hasValue
        rdf:resource="#CabernetSauvignonGrape"/>
    </owl:Restriction>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#madeFromGrape" />
      <owl:maxCardinality>1</owl:maxCardinality>
    </owl:Restriction>
  </owl:intersectionOf>
```

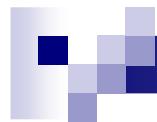
Beispiel (5)

```
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#hasColor" />
    <owl:hasValue rdf:resource="#Red" />
  </owl:Restriction>
</rdfs:subClassOf>
...
<-- Weitere Einschränkungen bezüglich
  hasSugar (#Dry), hasFlavor (#Moderate | #Strong)
  hasBody (#Medium | #Full)
-->
</owl:Class>
```

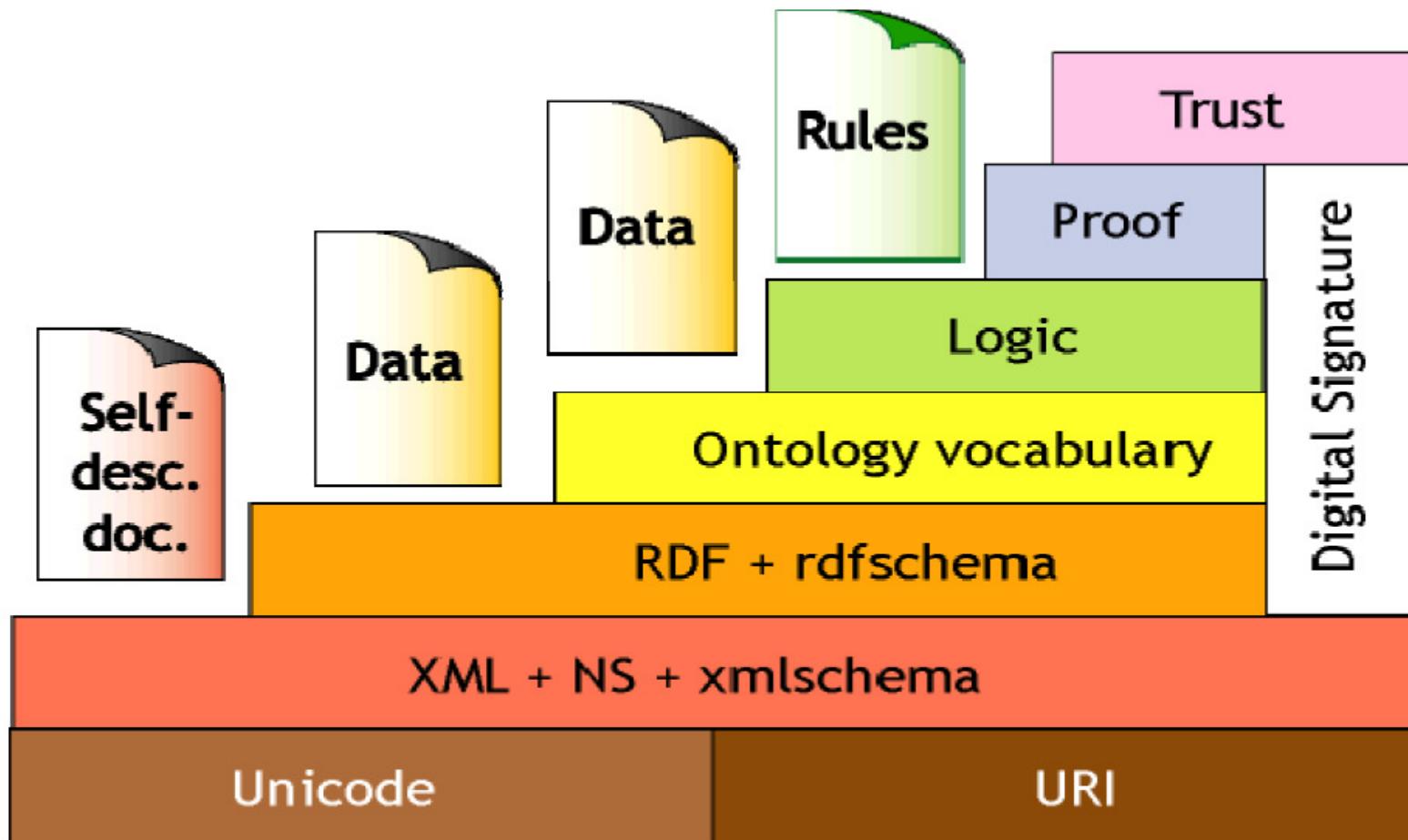
Beispiel (6)

```
<CabernetSauvignon rdf:ID="MariettaCabernetSauvignon">
  <locatedIn rdf:resource="#SonomaRegion" />
  <hasMaker rdf:resource="#Marietta" />
  <hasSugar rdf:resource="#Dry" />
  <hasFlavor rdf:resource="#Moderate" />
  <hasBody rdf:resource="#Medium" />
</CabernetSauvignon>
```

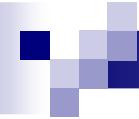
<http://www.w3.org/TR/2003/CR-owl-guide-20030818/wine>



Semantic Web Stack



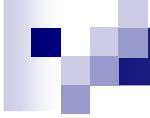
Aus: Tim Berners-Lee: <http://www.w3.org/2000/Talks/1206-xml2k-tbl/slides10-0.html>



Ausblick

- Regeln
- Beweise (Proof)
- Web of Trust

- Regel- und Anfragesprachen
- Editoren und Validatoren
- Reasoners und Frameworks
- Anwendungen

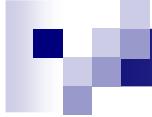


Fazit

- bietet neue Möglichkeiten
 - verteilte Expertensysteme
 - autonome selbstlernende Agenten
- Offene Fragen
 - Akzeptanz?
 - Sicherheit?
 - Datenschutz?
 - Effizienz?
- Weitreichende Standardisierung
- Weitere Entwicklung vermutlich innerhalb von einzelnen Domänen vorangetrieben



DANKE!



Quellen

- Robert Tolksdorf: [13-SemanticWeb.pdf](#)
- OWL-Overview:
<http://www.w3.org/TR/owl-features/>
- Scientific American: The Semantic Web
 - <http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&pageNumber=2&catID=2>
- [http://www.w3.org/2001/sw/](#)
- [www.wikipedia.de](#)