



Semantic Web Technologies I

Lehrveranstaltung im WS11/12

Dr. Elena Simperl

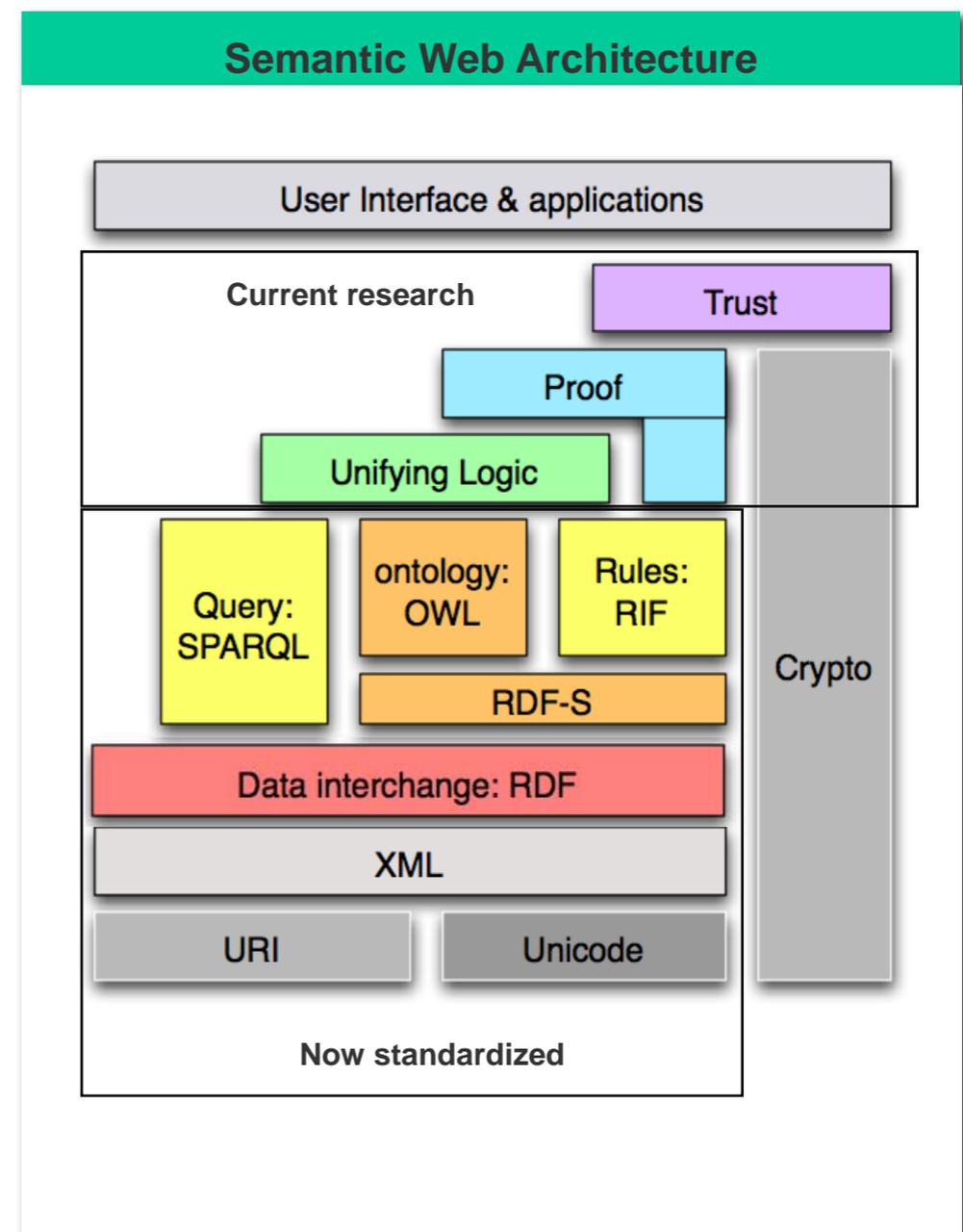
PD Dr. Sebastian Rudolph

M. Sc. Anees ul Mehdi

Ontology Engineering

Dr. Elena Simperl

XML und URIs
Einführung in RDF
RDF Schema
Logik - Grundlagen
Semantik von RDF(S)
SPARQL - Syntax und Intuition
Semantik von SPARQL
OWL - Syntax und Intuition I
OWL – Syntax und Intuition II
OWL - Semantik und Reasoning
Ontology Engineering
Linked Data
Konjunktive Anfragen / Einführung Regelsprachen
Anwendungen



Warum Ontology Engineering?



- Ontologiesprachen und Reasoner liefern die technische Infrastruktur für die Verwendung Semantischer Technologien
- Aber: domänenspezifische Wissensspezifikationen (=Ontologien) müssen irgendwoher kommen
- Fragen:
 - Wie erzeugt man (gute, nützliche) Ontologien?
 - Wie beurteilt man die Güte von Ontologien?

Warum Ontology Engineering?

AIFB 

- Viele Aspekte dieser Fragestellungen verwandt mit Problemen des Software Engineering
- In beiden Fällen werden (häufig kollaborativ) komplexe Artefakte geschaffen, die möglichst
 - korrekt
 - funktional
 - verständlich
 - wiederverwendbar
 - etc.sein sollen.

Bedarfsanalyse



- ...immer eine gute Idee, bevor man anfängt eine Ontologie zu bauen:
 - Wird überhaupt eine semantische Repräsentation benötigt? (Alternativ: z.B. Datenbanklösung)
 - Falls ja, sollte es eine formallogische sein? (Alternativ: z.B. textuelle Repräsentation, v.a. wenn es um den Mensch-Mensch-Informationsaustausch geht)
 - ◆ kontra: Einführungskosten, bisherige Praxis
 - ◆ pro: flexiblere Verwendung/Austausch
 - ◆ pro: Reasoning erlaubt Umgang mit implizitem Wissen

Bedarfsanalyse



- Wie ist die Toolunterstützung?
 - Macht man sich durch die Wahl des Ansatzes von einem speziellen Tool abhängig?
 - Wie ist die Lizenzsituation bei der verfügbaren Software?
 - Wie stabil/ausgereift ist die Software?
 - Welchen Support bietet der Tool-Hersteller an?
 - Sind die zur Verfügung stehenden Tools ausreichend interoperabel?

Bedarfsanalyse

- Funktionale Aspekte
 - Welche Domäne soll modelliert werden? Welche Aspekte dieser Domäne sollen abgebildet werden?
 - Was ist die benötigte Detailstufe für die zu erstellende Spezifikation?
 - Welche Aufgaben sollen mit Hilfe der Ontologie bewerkstelligt werden: „Browsen“ einer Wissensmenge, Suche nach Information, Anfragebearbeitung, automatisches Schlussfolgern?
 - Welche Schlussfolgerungen sind gewünscht?

Ontologieerzeugung



- Es gibt viele mögliche Quellen von Wissen, die ontologisch verwertet werden können:
 - Menschen
 - Texte
 - Webseiten
 - Datenbanken
- Diese Quellen unterscheiden sich dahingehend, (i) wie explizit und (ii) wie strukturiert das Wissen vorliegt

Menschen



- Domänenexperten = Mensch die über das nötige Wissen verfügen
- ...das sind i.A. keine Logiker und selten einer Ontologiesprache mächtig
- es werden sogenannte *knowledge engineers* als „Mittelsmänner“ benötigt
- diese müssen nicht nur über formallogisches Wissen sondern auch über kommunikative Fähigkeiten verfügen

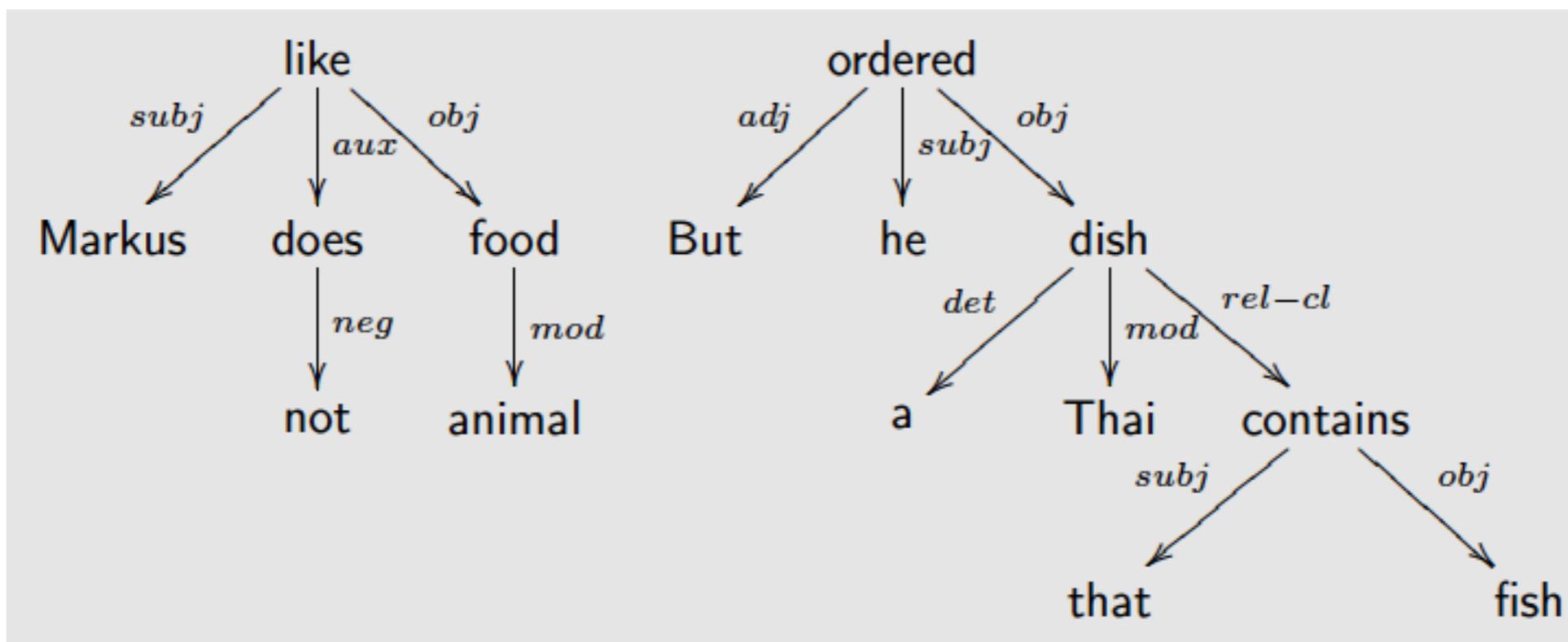
Texte



- am direktesten zugreifbar: digitale Textressourcen, trotzdem Text \neq Logik
- Möglichkeit: Anwendung von automatischen Verfahren zur Gewinnung von Information aus Text (*Ontology Learning*)
- Verschiedene Ansätze:
 - pattern-basierte Suche nach vordefinierten Relationen (*Information Extraction*)
 - vollständige syntaktische Zerlegung und Umwandlung in logische Ausdrücke (*deep semantic analysis*)

Deep Semantic Analysis - Beispiel

Markus does not like animal food. But he ordered a Thai dish that contains fish.



$$\neg \exists \text{likes.}(\text{Animal} \sqcap \text{Food})(\text{markus})$$

$$\exists \text{ordered}(\text{Dish} \sqcap \exists \text{contains.Fish})(\text{markus})$$

Semistrukturierte Ressourcen



- Im Gegensatz zu Texten haben z.B. Webseiten schon mehr explizite Struktur (Markup, Verlinkungen etc.)
- Diese können direkt z.B. in RDF abgebildet werden.
- Andere Beispiele:
 - Wikis
 - Dateisysteme

Strukturierte Ressourcen



- Datenbanken sind vollständig strukturiert
- Zur „Anbindung“ einer Datenbank an eine Ontologie ist ein Mapping z.B. vom Schemateil der Datenbank auf Klassen / Propertys der Ontologie nötig.
- Dann kann der „Datenteil“ der DB als Abox interpretiert werden.

Ontology Evaluation



- Was macht eine gute Ontologie aus?
- Es gibt verschiedene Kriterien, z.B.:
 - logische Kriterien
 - strukturelle/formelle Kriterien
 - „Korrektheit“

Logische Kriterien



- Konsistenz der Wissensbasis / Klassen

Horse \sqsubseteq \neg Flies
 FlyingHorse \equiv Horse \sqcap Flies

- logische Vollständigkeit

Bird \sqsubseteq \neg Mammal Bird(ostrich)
 Bird \sqsubseteq Oviparous Mammal \sqcap Viviparous(lion)
 Oviparous \sqsubseteq \neg Viviparous

Mammal \sqsubseteq Viviparous

Mammal \sqcap Oviparous(platypus)

Strukturelle Kriterien

- Untersuchung der Klassenhierarchie:

Architecture \sqsubseteq Faculty
University \sqsubseteq Building

Faculty \sqsubseteq University
Building \sqsubseteq Architecture

- Korrektheit zu prüfen ist eher problematisch
(*grounding problem*)

Typische Modellierungs“fehler“



- Disjointness vergessen

Man \sqsubseteq Human
Man(pascal)

Human \sqsubseteq Man \sqcup Woman

Woman \sqsubseteq Human
Woman(anne)

- Rolleneigenschaften vergessen
- Domain / Range zu spezifisch
- Fehlinterpretation der Semantik des Allquantors
- Verwechslung von „Teil von“ mit „Subklasse von“

Happy $\equiv \forall \text{hasChild.Happy}$

Finger \sqsubseteq Hand
Toe \sqsubseteq Foot

Hand \sqsubseteq Arm
Foot \sqsubseteq Leg

Arm \sqsubseteq Body
Leg \sqsubseteq Body

Arm \sqcap Leg $\sqsubseteq \perp$

Typische Modellierungs“fehler“



- Richtung der Property unklar

```
ex:author    rdfs:range    ex:Publication .  
ex:author    rdfs:domain  ex:Person .  
ex:macbeth   ex:author    ex:shakespeare .
```

- Subklassen vs. Äquivalenz
- zu „wörtliche“ Übersetzung...

Modularisierung / Patterns



- erleichtert Wiederverwendbarkeit
- ermöglicht ggf. schnelleres Reasoning
- Verwendung von „Best Practices“

Ontology Refinement



(semi)automatische „Verbesserung“ von Ontologien

- Ontology Repair
- Ontology Update / Evolution
- logische Vervollständigung von Ontologien