## Ontologien mit RDF Schema

Markus Stocker

11. Juni 2018

## Rekapitulation

- Was ist RDFS? Wofür verwendet man die Sprache?
- Was ist eine Klasse?
- Was ist rdfs:subClassOf?
- Was ermöglicht rdfs:range?

#### Übersicht

- Was ist Ontologie?
- Bestandteile und Aufbau
- Ontologien mit RDFS
- Ontologien entwickeln
- Protégé

# Was ist Ontologie?

- Eine Disziplin der Philosphie
- Die Lehre vom Sein
- Befasst sich mit den Grundstrukturen der Wirklichkeit
- Die fundamentalen Klassen und Beziehungen existierender Dinge
- Aristoteles 10 Kategorien: Substanz, Quantität, Relation, ...

## Ontologie in der Informatik

- Teil der Wissensrepräsentation im Gebiet Künstliche Intelligenz
- Beschreibung von "Wissen" über eine Domäne
- Begriffe und Beziehungen eines Gegenstandsbereiches
- Beispiel: Erdähnliche und Gas Planeten mit Satelliten
- Eine maschinenverarbeitbare (formale) Beschreibung
- Beispiel: TerrestrialPlanet rdfs:subClassOf Planet

## Ontologie in der Informatik

- Konsensbildung zwischen mehreren Parteien (z.B. in Projekte)
- Dienen dem Austausch von Wissen zwischen Anwendungen
- Interoperabilität verbessern

#### Bestandteile einer Ontologie

- Klassen (concepts, classes)
  - Mengen der Dinge mit gleichen Eigenschaften
  - Auch als Begriffe bezeichnet
  - ► Beispiel: Die Menge der Planeten
- Relationen (relations, properties)
  - Dienen der Beschreibung von Eigenschaften
  - Beziehungen zwischen Klassen und deren Instanzen
  - ▶ Beispiel: Die Eigenschaft Radius, Beziehung zwischen Planet und Wert
- Instanzen (instances, individuals)
  - ▶ Elemente einer oder mehrerer Klassen
  - Auch als Objekte bezeichnet
  - Beispiel: Der Planet Erde als Instanz der Klasse der Planeten

## Aufbau einer Ontologie

- Schema (terminological box): Vokabular, Klassen und Relationen
- Inhalt (assertional box): Instanzen/Individuum, Aussagen, Daten
- Dabei verwenden Inhalte das Schema
- Beispiel

```
ex:Planet rdf:type rdfs:Class
ex:earth rdf:type ex:Planet
```

- Das erste Tripel ist Schema
- Das zweite Tripel ist Inhalt
- ex:Planet ist eine Klasse
- ex:earth is eine Instanz (Individuum)

#### Aufbau einer Ontologie: Schema

- Das Schema besteht aus Axiome
- Diese gelten als wahr
- Dienen zur Repräsentation von abstraktem Wissen
- Beispiele

```
ex:TerrestrialPlanet rdfs:subClassOf ex:Planet ex:radius rdfs:domain ex:Planet
```

## Aufbau einer Ontologie: Inhalte

- Die Inhalte bestehen aus Aussagen
- Dienen zur Repräsentation von konkretem Wissen
- Beispiele

```
ex:earth ex:radius "6371" ex:earth rdf:type ex:Planet
```

## Ontologien mit RDFS

- RDFS ist eine Sprache mittels der man Ontologien erstellen kann
- RDFS Dokument ist eine Spezifikation von Wissen über eine Domäne
- Wobei die Spezifikation maschinenverarbeitbar ist
- Die wichtigsten Bestandteile der Sprache haben wir bereits gesehen
- Insb. rdfs:Class, rdfs:subClassOf, rdfs:subPropertyOf

## Ontologien Entwickeln

- Ontologien entwickeln (ontology engineering) ist eine eigene Disziplin
- Über einfache Beispiele hinaus, ist die Aufgabe generell komplex
- Schwierigkeit nicht primär im Erlernen von Technologie (z.B. RDFS)
- Schwieriger ist Wissen aus Köpfen in Dokumente zu formalisieren

## Ontologien Entwickeln

- Der zweck der Ontologie spielt eine wichtige Rolle
- Oft bestimmt der Zweck welche Aspekte man formalisiert
- Entwicklung von Ontologien also strukturiert angehen
- Insbesondere auch Anforderungsanalyse durchführen
- Ähnlich wie bei der Entwicklung von Software

## Ontologien Entwickeln

- Zuerst schauen ob es bereits eine entsprechende Ontologie gibt
- Die wiederverwendet werden kann, auch nur Teilweise
- Eventuell gibt es eine übergeordnete Ontologie
- Welche abstrakteres Wissen formalisiert
- Auf einer solchen Aufbauen ist oft hilfreich

## Anforderungsanalyse

- Benötigt die Anwendung eine Ontologie, semantische Repräsentation?
- Oder genügt eine klassische Datenbank (u.U. bereits vorhanden)?
- Ist die Toolunterstützung adäquat für mein Projekt?
- Wie sind die Tools lizensiert und wie ausgereift sind sie?

## Anforderungsanalyse

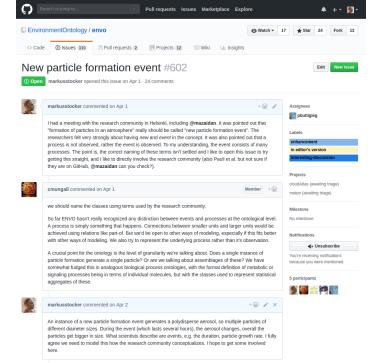
- Welche Domäne wird modelliert?
- Welche Aspekte müssen erfasst werden?
- Wie detalliert soll Wissen beschrieben werden?
- Welche T\u00e4tigkeiten soll die Ontologie unterst\u00fctzten?

#### Ontologie Erstellen

- Übersetzung von Wissen in eine maschinenlesbare Form
- Es gibt mehrere Arten von Quellen für Wissen
- Z.B. menschliche Köpfe, Bücher, Web, Datenbanken
- Einige Quellen eignen sich besser als andere für Formalisierung

#### Wissensquelle: Mensch

- Experten halten viel Wissen in Köpfen
- Die meisten können Wissen aber nicht formalisieren
- Weil sie sich mit RDFS nicht auskennen
- Zusammenarbeit mit "Wissensingenieure" nötig
- Diese führen Interviews durch und erfassen Expertenwissen
- Experten können nicht immer dinge genau definieren
- In der Kommunikation können Misverständnisse entstehen



# Wissensquelle: Bücher (Unstrukturiert)

- Text ist einfach für Menschen zu verarbeiten
- Formales Wissen aus Text (automatisiert) zu extrahieren ist schwierig
- Natürliche Sprache ist oft mehrdeutig, kontextabhängig, usw.
- Der Prozess leidet meist unter Informationsverlust
- Obwohl hier viel geforscht wurde, sind Resultate eher bescheiden

# Wissensquelle: Web (Semistrukturiert)

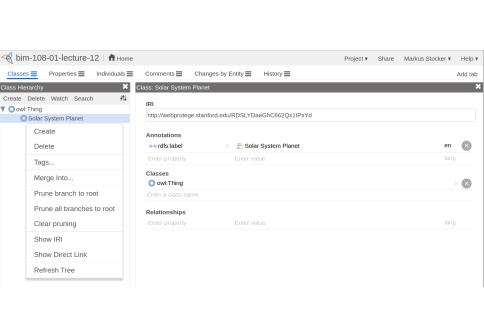
- Gewisse Inhalte kommen Strukturiert daher
- Beispiel: Wikipedia Artikel enthalten auch strukturierte Information
- Solche Inhalte kann man einfacher formalisieren
- Struktur gibt es auch zwischen Inhalten
- Beispiel: Explizite Verlinkung zwischen Seiten
- Weist auf eine Verbundenheit hin, wobei die Art oft unklar bleibt
- Metadaten sind oft Strukturiert
- Beispiel: Exif in Bilder
- Bildinhalt unbekannt aber Zugang auf Dateiname/-grösse/-typ, Zeit

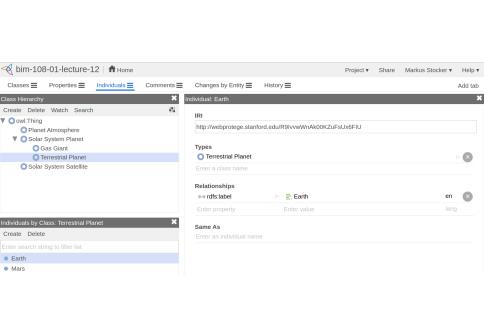
# Wissensquelle: Datenbanken (Strukturiert)

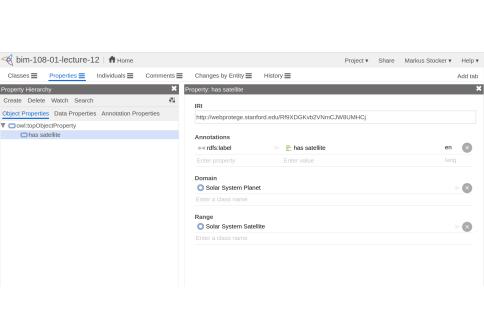
- Datenbankinhalte können meist nach RDF übersetzt werden
- Damit erhält man auf einfache Weise semantische Aussagen
- Schemainformation kann man zusätzlich verwenden
- Um terminologisches Wissen aufzubauen
- Andere Ontologien sind auch Quellen f
  ür bereits formalisiertes Wissen

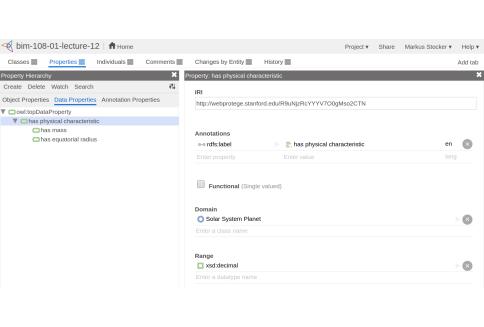
# Ontologien Bearbeiten: Protégé

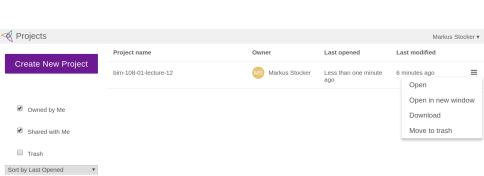
- Protégé ist ein Programm mit dem man Ontologien editieren kann
- Frei erhältlich unter https://protege.stanford.edu
- Als Lokalinstallation oder im Web verfügbar
- Web Version unter https://webprotege.stanford.edu/
- Wir benutzen Protégé um eine kleine Ontologie zu entwickeln











## Zusammenfassung

- Mit Ontologien kann man Wissen formalisieren
- Mit RDFS können einfache Ontologien erstellt werden
- Die Sprache hat eine relativ kleine Expressivität
- Man kann nur "einfaches Wissen" ausdrücken
- Es gibt noch weitere ausdruckstärkere Sprachen (OWL)
- Die Entwicklung von Ontologien ist generell komplex
- Die Aufgabe muss somit strukturiert angegangen werden