Informationsintegration

Semantic Web

Ulf Leser

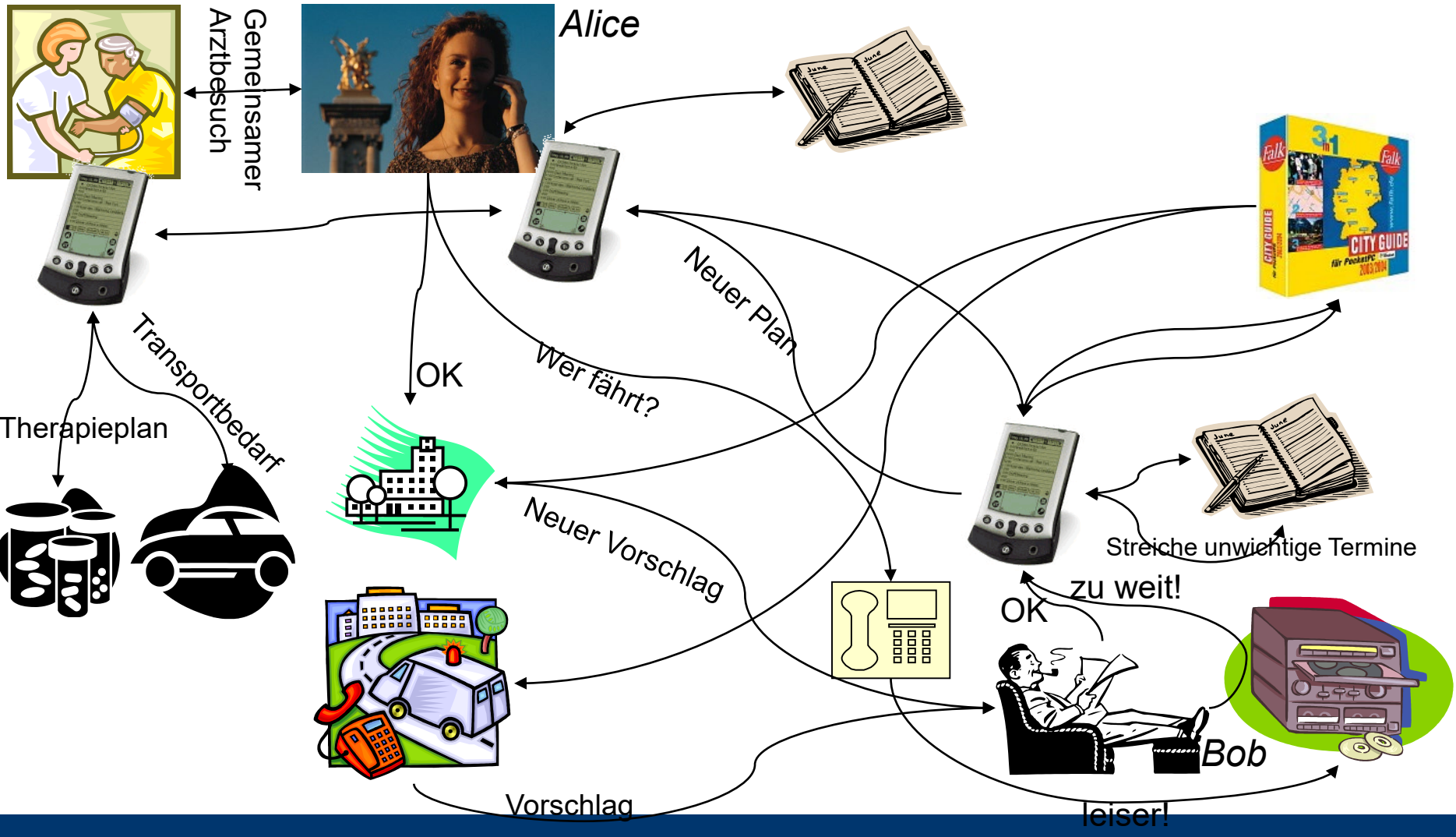
Inhalt dieser Vorlesung

- Grundidee des Semantic Web
- Layer Cake
- RDF und RDFS
- SparQL
- Die OWL Sprachfamilie

Tim Berners-Lee et al.

- „Das Semantic Web ist eine Erweiterung des gegenwärtigen Web, in der Informationen eine wohldefinierte Bedeutung erhalten, so dass Computer und Menschen besser zusammenarbeiten können“ [BHL01]
  - Erweiterung, kein Neudesign; Abwärtskompatibilität ist essentiell
  - Zusammenarbeit zwischen Menschen und Computern und Computern und Computern verbessern
    - Integration von Daten und Anwendungen
    - Intelligentere, persönliche, kontextbezogene, … Dienste
  - Mittel: Explizite Definition der Bedeutung von Informationen
- Sollte als Vision verstanden werden
  - An deren Erfüllung man arbeitet (z.B. W3C)

Szenario [BHL01]



Ist-Zustand: Web 1.0/2.0

- Web Seiten werden in HTML verfasst
- Struktur nur zur Unterstützung des Layouts
  - Gut für Lesbarkeit
  - Nicht automatisch interpretierbar (aber: NLP)
- Informationen leben in zwei Welten
  - Für Menschen als Konsumenten
    - Gedichte, Filme, Text,...
  - Für Computer als Konsumenten
    - Daten, Programme,...
- Das Web betont den Menschen
- Das Semantic Web soll dies ausgleichen

Beispiel-Anwendungen

- Anfragen statt Suche, Antworten statt Webseiten
  - Question Answering versus Information Retrieval
  - Liste alle Telefonnummern aller Mitarbeit*innen der HU Informatik
  - Wann wurde Rembrandt geboren?
- Web Commerce
  - Shopping-Agenten suchen bestes und billigstes Angebot
    - Die billigste Waschmaschine mit Energieklasse A+ in Düsseldorf
  - Online Shops präsentieren Waren zielgerichtet
    - Wer einen Grill kauft, braucht Kohlen
  - Broker vermitteln zwischen Anbieter*innen und Käufer*innen
    - Angebote für 10.000 Dübel, 6mm, Hohlkammer
- ...

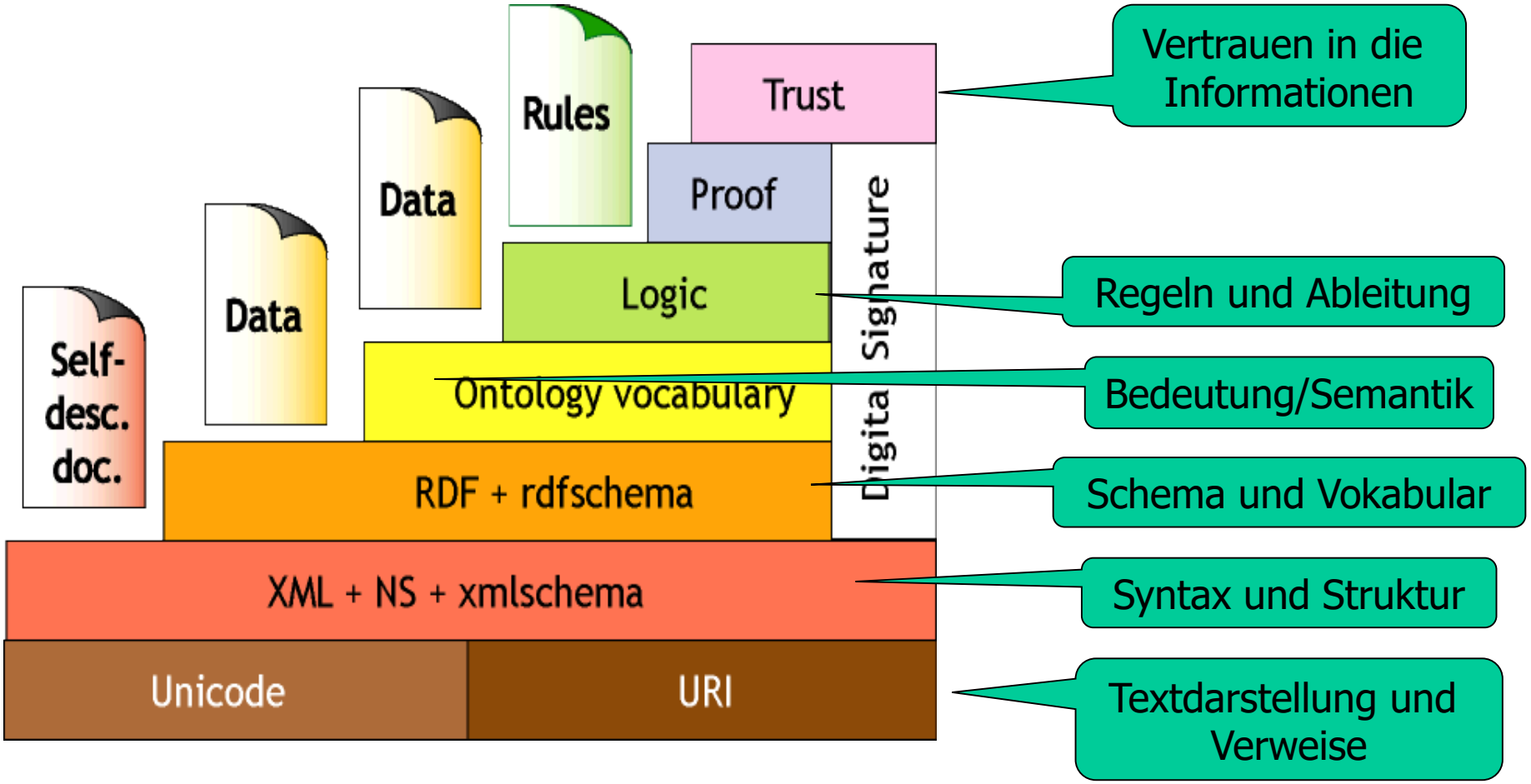
Grundprinzipien „Semantic Web“

- Semantik wird durch Annotation und Ontologien spezifiziert
- Uniform Resource Identifier (URI)
  - Sage nicht „Farbe“, sage "http://www.pantomime.com/std6#farbe"
  - Definitionen können im Laufe der Zeit ergänzt werden
  - Aber: Definitionen können jederzeit verändert oder ersetzt werden
- Keine Erzwingung von Konsistenz oder Kontinuität
  - Ist „http://www.pantomime.com/std6#farbe“ noch dasselbe?
  - Ist „http://www.pantomime.com/std6#farbe“ dasselbe wie „http://www.colors.com/colors“?
  - „Jeder kann Beliebiges über Beliebiges sagen"
- Sehr lose Koppelung
  - Hochgradig dezentrales und flexibles Design
  - Preis: Keine Sicherstellung von Konsistenz

Inhalt dieser Vorlesung

- Grundidee des Semantic Web
- Layer Cake
- RDF und RDFS
- SparQL
- Die OWL Sprachfamilie

Semantic Web „Layer Cake“



Quelle: [Hen02]

1+2. Unicode, URI und XML

- Unicode / URI
  - Semantikfrei
  - Unicode: Standard zur binären Repräsentation von Zeichen
  - URI: Identifikation von virtuellen oder physischen Ressourcen
  - URI ist ein globaler Schlüssel
- XML / Namespaces / XML Schema
  - Standard zur Darstellung strukturierter Daten
  - Mit geeigneter Kodierung auch für nicht-hierarchische Daten
    - Serialisierung von Daten in XML
    - Z.B. für RDF, OWL, Relationen, …
  - XSchema: Definition von Schemata
    - Erlaubt Konsistenzprüfung von XML Dokumenten
  - Austauschschema, nicht Speicherschema

3. Resource Description Framework

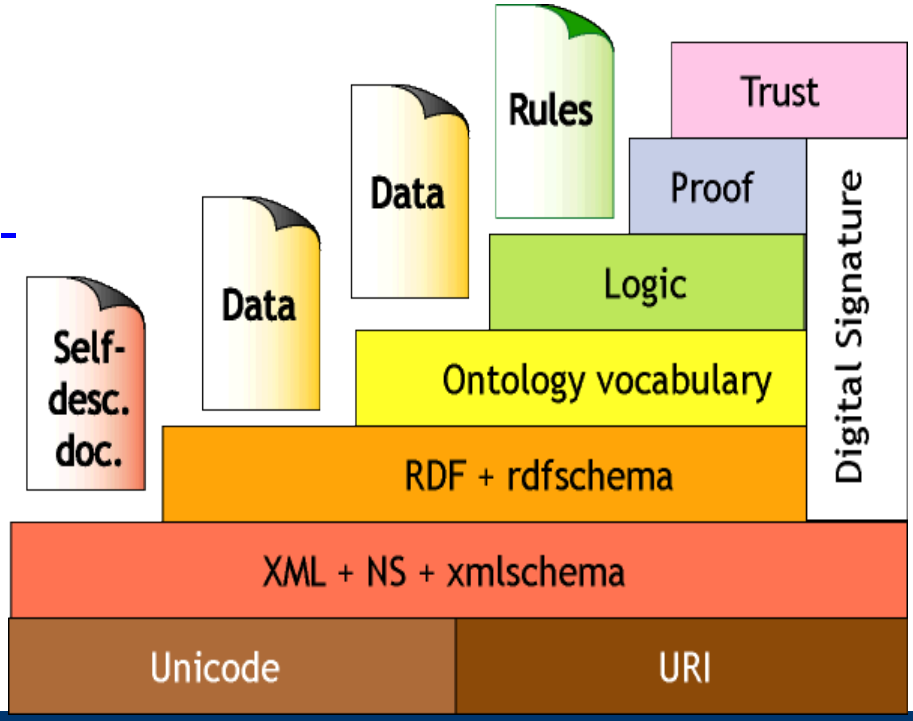
- Graphbasiertes Datenmodell
- Informationen werden als Tripel modelliert
  - (Subjekt Prädikat Objekt)
  - Beschreibung der Werte von Eigenschaften von Ressourcen
  - Erlaubt Aussagen über alles
    - Insbesondere auch über andere Aussagen -> später
- RDF Datenbasis = Menge von Tripeln
- RDFS („RDF Schema“)
  - Festlegung eines Vokabulars für RDF Datenbasen
  - Typisierung von RDF Daten
    - Datentypen, Spezialisierung, getypte Beziehungen, …
- Gleich mehr dazu

4. Ontology Vocabulary

- Konzeptualisierung von Domänen
- Ressourcen erhalten Bedeutung durch Einbettung in eine formale Ontologie
- Interoperabilität von RDF Datenbasen durch Verwendung derselben Ontologie
  - Beziehungen von Konzepten über Datenbasis-Grenzen hinweg
  - Bei verschiedenen Ontologien: Ontologie-Alignment
- Alles freiwillig, lose Kopplung
  - RDF Dokumente müssen nicht zu RDFS-Definitionen konform sein
  - Konzepte müssen nicht durch Ontologien untermauert werden
  - Alles kann sich ständig ändern

5-7. Logic, Proof, Trust

- Keine klare Aufteilung
  - Logic: Inferenz einer Wissensrepräsentationssprache wie OWL
  - Proof: ?
- Trust
  - Maßnahmen zur Beurteilung des Vertrauens in Daten und Schlüsse
  - Schutz vor Spam-Seiten, Spam-RDF-Datenbanken, Spam-Ontologien
  - Sehr schwierige Umsetzung
- Meines Wissens nicht umgesetzt



Inhalt dieser Vorlesung

- Grundidee des Semantic Web
- Layer Cake
- RDF und RDFS
- SparQL
- Die OWL Sprachfamilie

RDF Grundlagen

- Grundlegendes Element sind Aussagen bestehend aus
  - Ressource (Subjekt)
  - Eigenschaft (Prädikat)
  - Wert / Ressource (Objekt)
- Beispiel: „Hitchcock ist der Regisseur von Marnie“
  - RDF-Tripel: **(Hitchcock, www.duden.de/regisseur, Marnie)**
  - Serialisiert in XML

```
<?xml version="1.0">
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
   <rdf:Description rdf:about="Alfred Hitchcock">
       <ist_regisseur_von> Marnie </ist_regisseur_von>
   </rdf:Description>
</rdf:RDF>
```

  - Prädikatenlogik: **istRegisseur(Hitchcock, Marnie)**

Mehr-arige Prädikate

- RDF Tripel können nur binäre Prädikate ausdrücken
- Für ternäre, quartäre, … Prädikate müssen „künstliche" Ressourcen erschaffen werden
  - Entweder manuell oder durch Blank Nodes (später)
  - Formal ist das das Skolemisierung (siehe Frozen Facts)
- „Hitchcock hat 1964 den Film Marnie gedreht"
  - Neue künstliche Ressource **`MarnieFilm`**

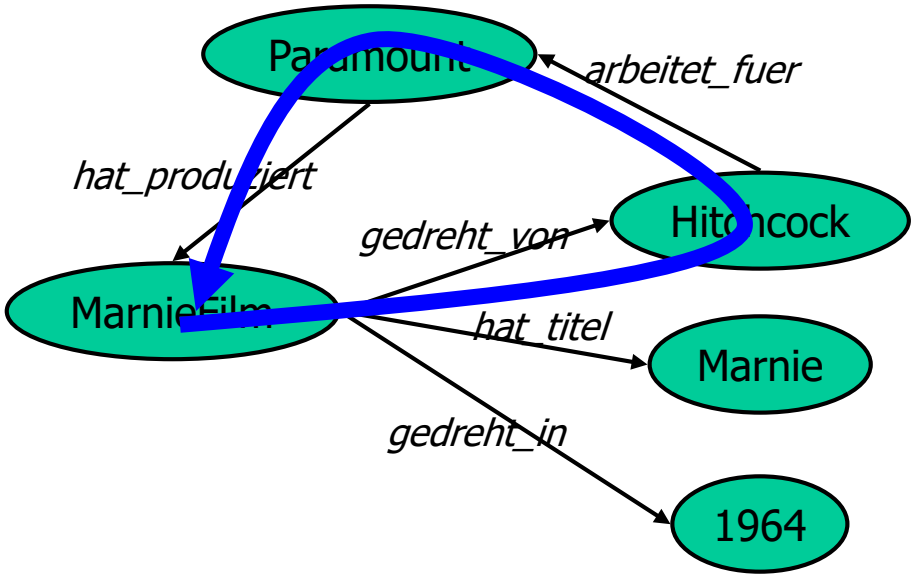
```
(MarnieFilm, gedreht_von, Hitchcock)
(MarnieFilm, hat_titel, Marnie)
(MarnieFilm, gedreht_in, 1964)
```

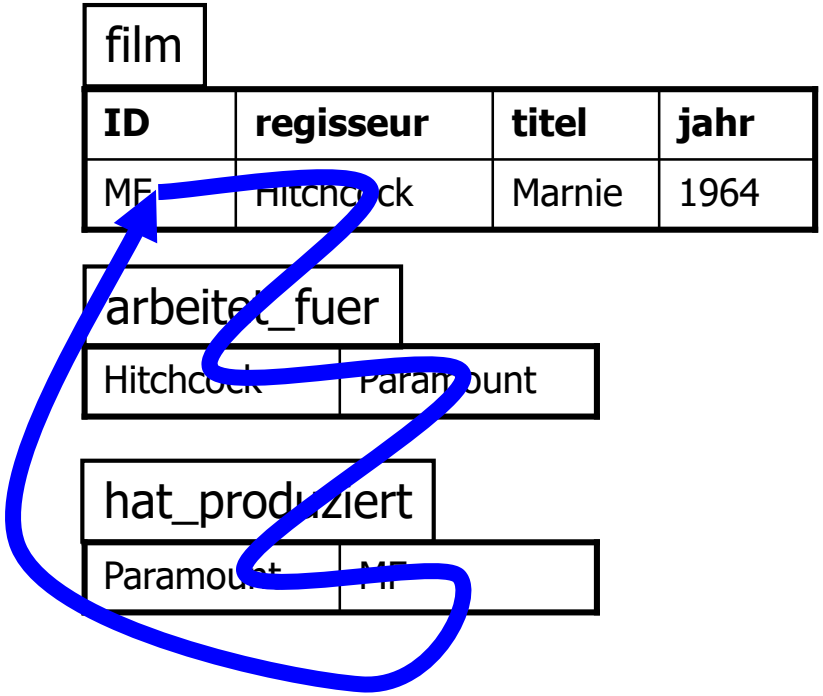
RDF als Graph

- Graphen als natürliche Repräsentation von RDF
- Naiver Ansatz
  - Subjekte und Objekte werden Knoten
  - Prädikate sind Kanten
- Eigenschaften des Graphen
  - (Alle) Knoten haben eindeutige Label (URI oder Wert)
  - Kanten sind gerichtet und haben keine eindeutigen Label
  - Knoten können durch mehr als eine Kante verbunden sein (Multigraph)

RDF als Graph

```
(MarnieFilm gedreht_von Hitchcock)
(MarnieFilm hat_titel Marnie)
(MarnieFilm gedreht_in 1964)
(Paramount hat_produziert MarnieFilm)
(Hitchcock arbeitet_für Paramount)
```

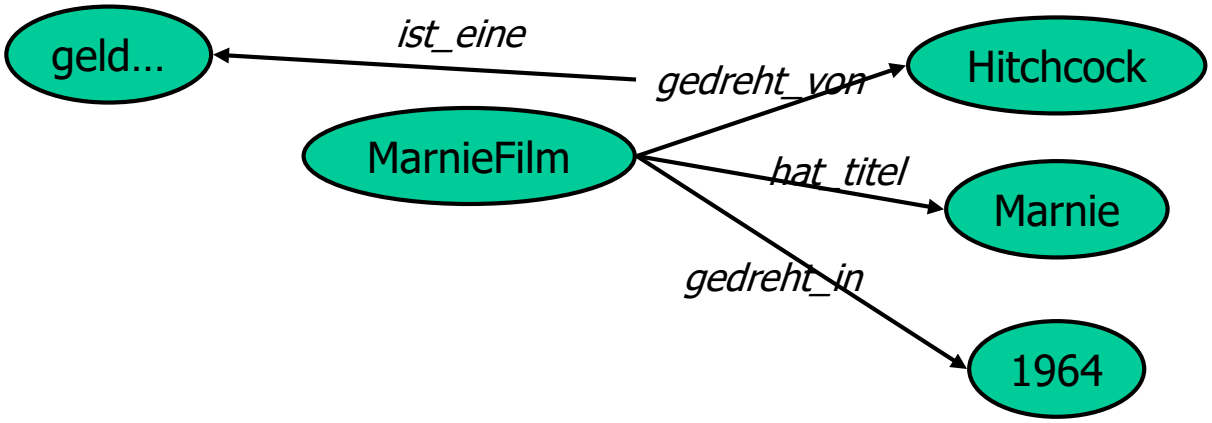




Problem

- Auch Prädikate sind Ressourcen
- Über die kann man Aussagen machen
- Erfordert Kanten von/auf Kantenlabel

```
(MarnieFilm gedreht_von Hitchcock)
(MarnieFilm hat_titel Marnie)
(MarnieFilm gedreht_in 1964)
(gedreht_von ist_eine gelderwerbstätigkeit)
```

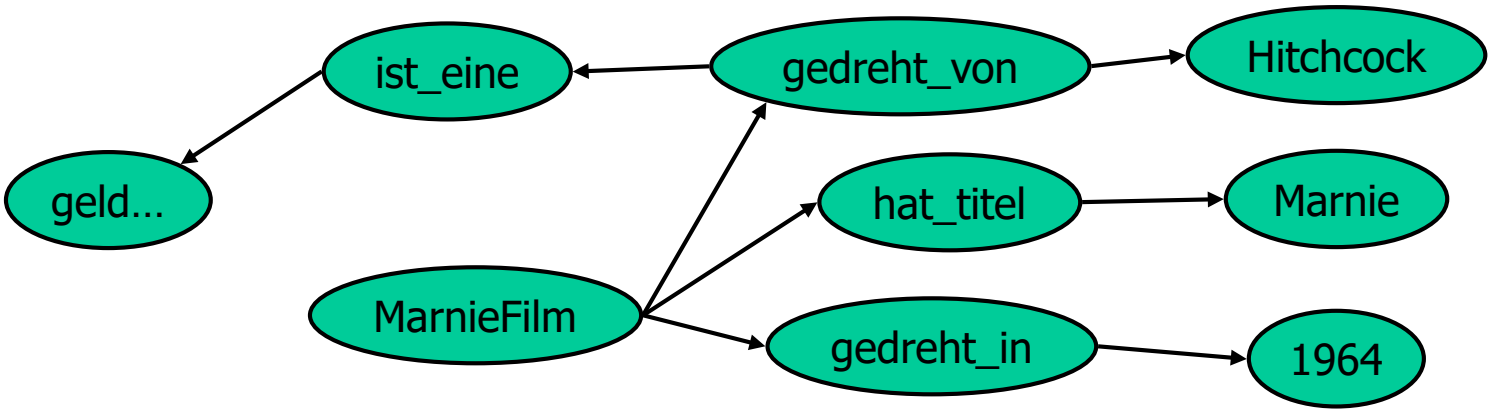


Warum ist `gedreht_von` kein Knoten?

Problem

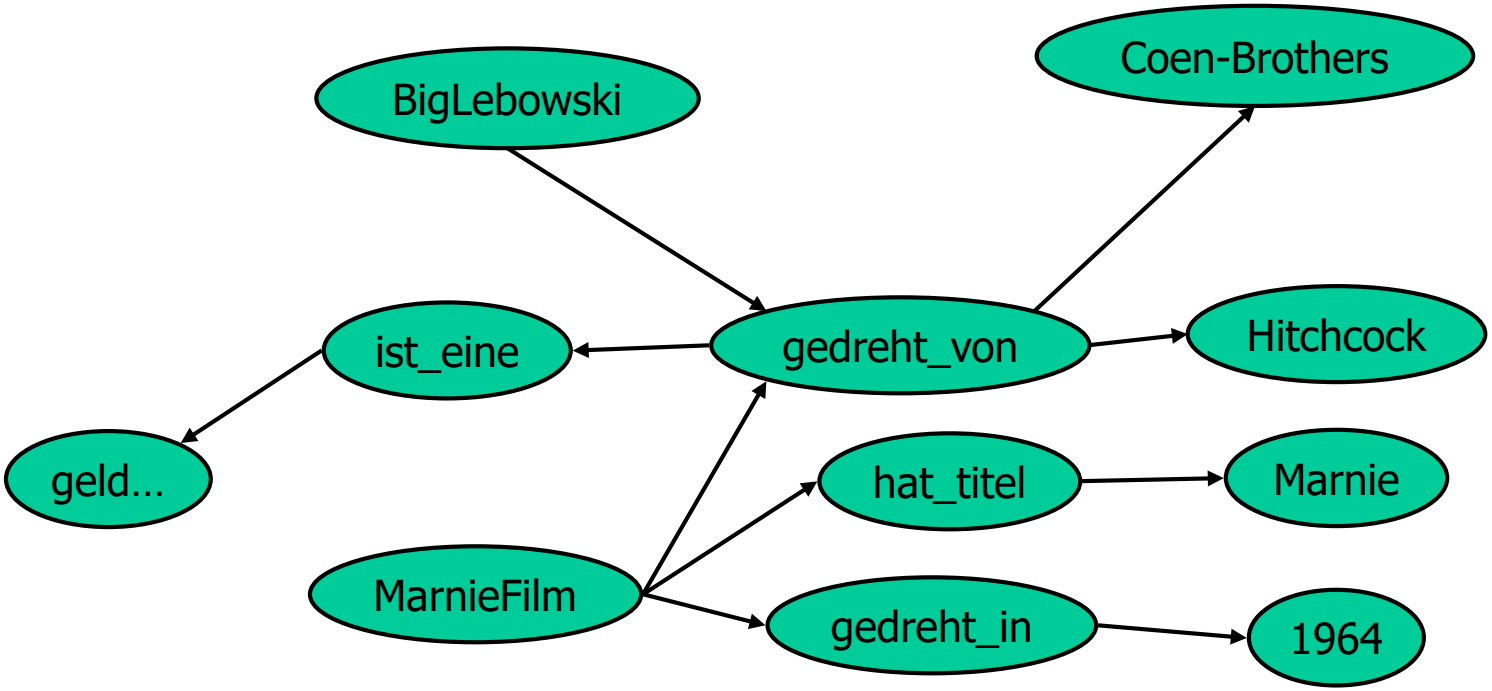
- Auch Prädikate sind Ressourcen
- Über die kann man Aussagen machen
- Erfordert Kanten von/auf Kantenlabel

```
(MarnieFilm gedreht_von Hitchcock)
(MarnieFilm hat_titel Marnie)
(MarnieFilm gedreht_in 1964)
(gedreht_von ist_eine gelderwerbstätigkeit)
```



Wo sind unsere Aussagen?
Was schreiben wir an die neuen Kanten?

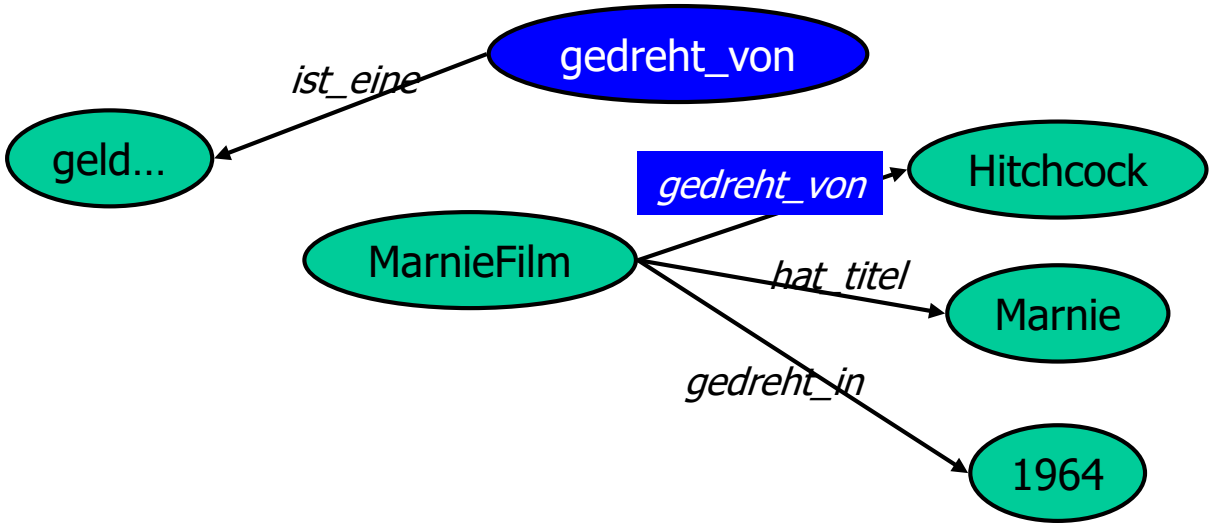
Problem



Wer hat was gedreht?

Offizielle Version

- W3C Spezifikationen
  - „The nodes of an RDF graph are its subjects and objects."
  - „A URI reference or literal used as a node identifies what that node represents. A URI reference used as a predicate identifies a relationship between the things represented by the nodes it connects. A predicate URI reference may also be a node in the graph."

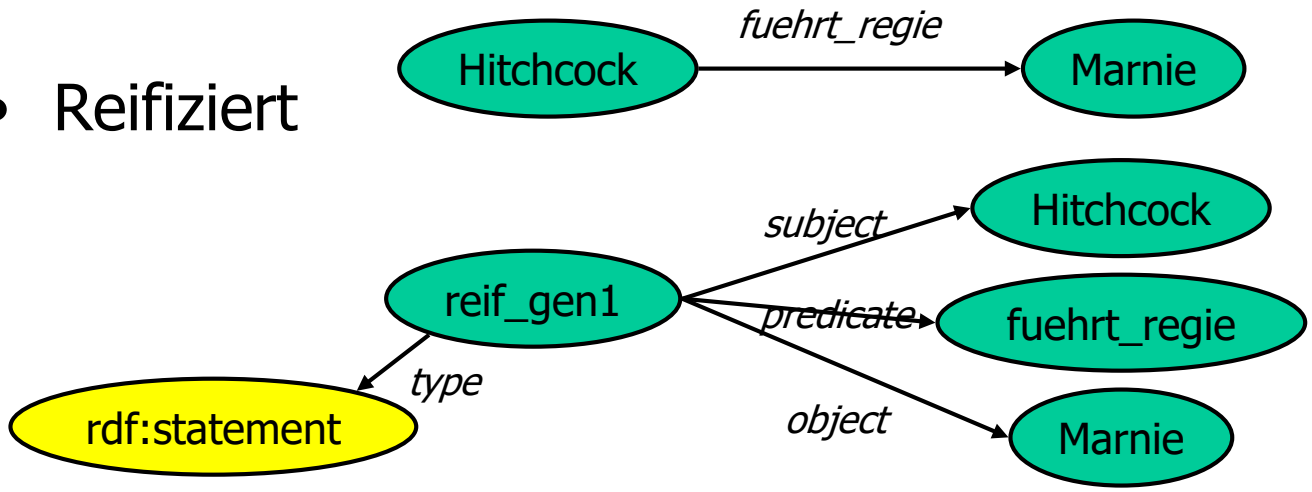


Aussagen über Aussagen

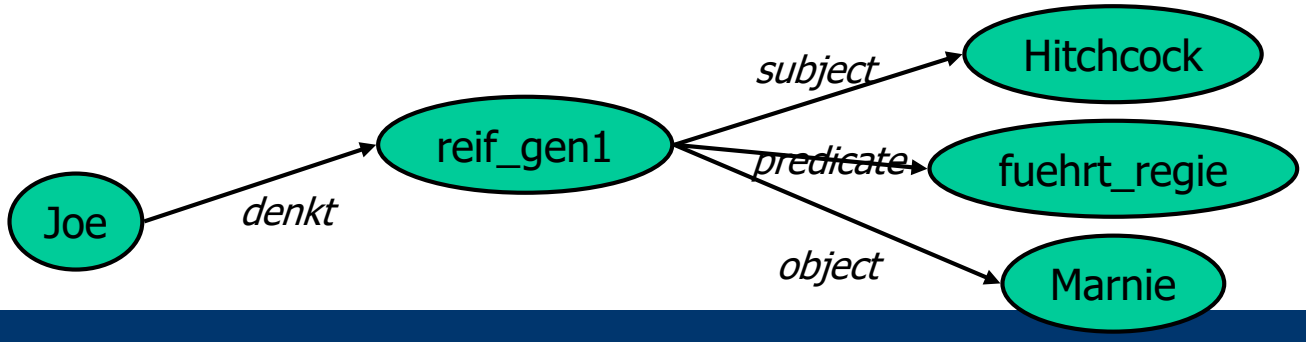
- In RDF kann man Aussagen über Aussagen treffen
  - Hitchcock ist der Regisseur von Marnie
  - Joe denkt, dass Hitchcock der Regisseur von Marnie ist
- Um eine Aussage über eine Aussage X machen zu können, muss man X reifizieren
  - Reification = „Verdinglichung“
  - Eine Aussage wird als Ressource behandelt
- Vorgehen für Aussage (S P O)
  - Man schafft einen neue Ressource R vom Typ `RDF:Statement`
  - Drei Aussagen: `(R subject S) (R predicate P) (R object O)`
  - R kann als Ressource verwendet werden: `(Joe denkt R)`

Grafisch

- „Hitchcock ist der Regisseur von Marnie“
- Reifiziert

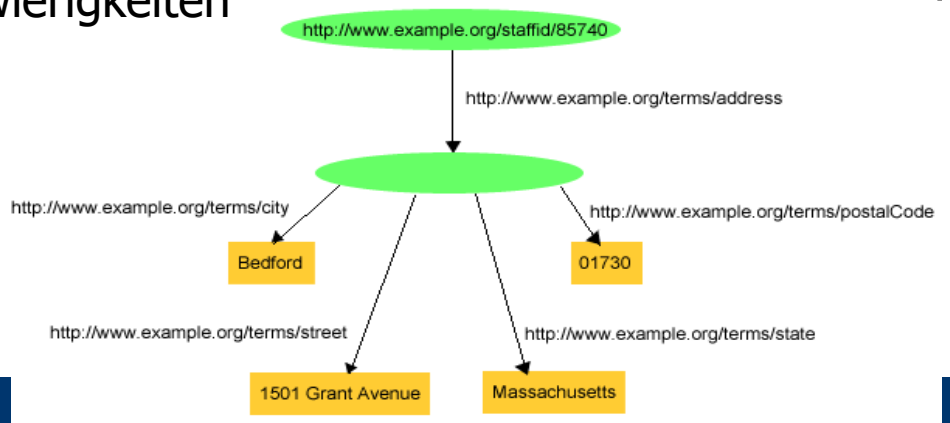


- „Joe denkt, dass Hitchcock der Regisseur von Marnie ist“



Weitere RDF Konzepte

- Aussagen über Mengen von Ressourcen
  - Gruppierung von Aussagen in Bags und (geordneten) Sequenzen
- Ressourcen können einen Typ haben
  - Wird im Allgemeinen nicht weiter interpretiert
  - Spezielle interpretierte Typen wie rdf:statement, rdf:bag, …
- „Blank Nodes"
  - Statt URIs und Literalen können auch „Blank Nodes" als Subjekt und Objekt verwendet werden
    - Bringt formal viele Schwierigkeiten
  - Gutartige Verwendung – n-arige Prädikate
  - Beispiel: Adresse mit 5 Elementen



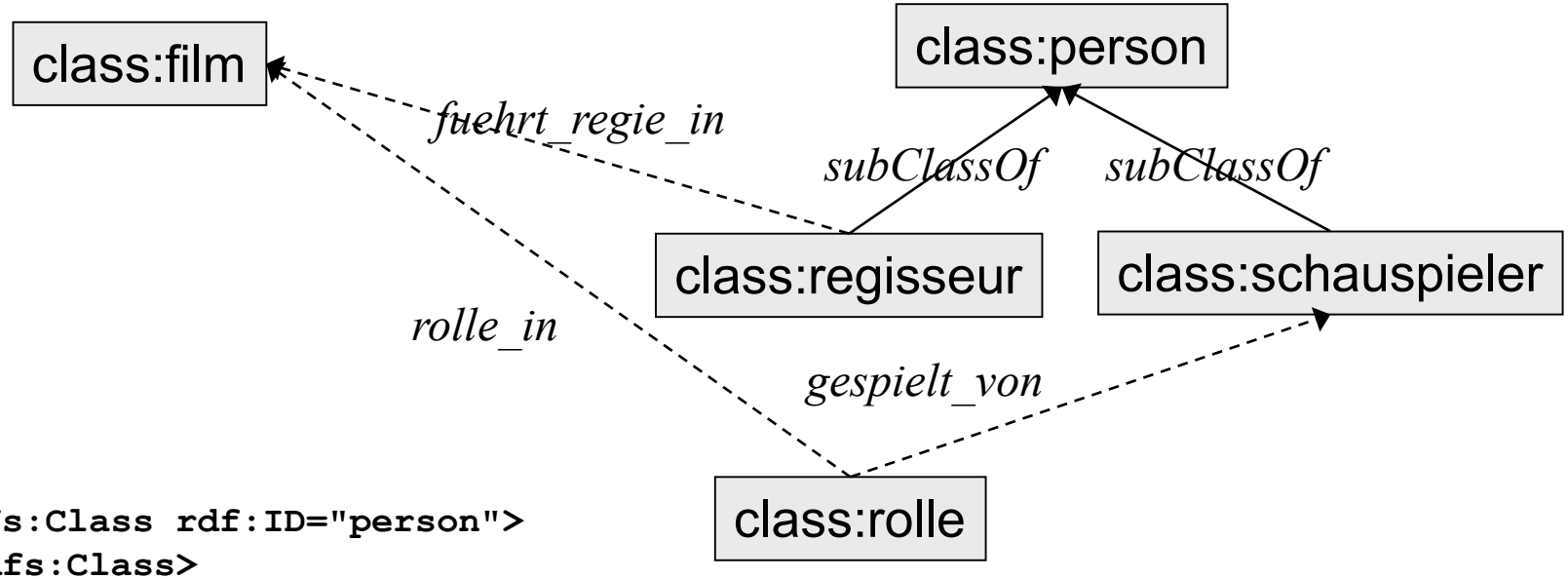
Bewertung von RDF

- Sehr flexibles Datenmodell
  - Keine Trennung von Dingen und Beziehungen
  - Blank nodes für n-arige Relationen
    - Sagen nur: Es existiert ein Knoten, der … (aber geben keine Identifikation)
    - RDF Graphen sind eine Mischung aus Extension und Intension
  - Reifikation
    - Kann man nicht auf Aussagen in PL-I abbilden
- Für Massendaten nicht geeignet (und nicht gedacht)
- Entworfen für heterogene, schwach strukturierte und wissensintensive Anwendungen

RDFS – Schemata für RDF

- Eine RDF Datenbasis ist vollkommen frei in den verwendeten Begriffen
  - Kein Schema – das erschwert die Analyse / Konsistenzprüfung
- RDFS
  - RDFS: RDF Vocabulary Description Language
  - Spezifikation von
    - Typen von Ressourcen (rdfs:class)
    - Subtypbeziehungen zwischen Typen (rdfs:subClassOf)
    - Eigenschaften eines Typen (rdfs:property)
    - Erlaubte Typen in Subjekt und Objekt von Prädikaten (rdfs:domain, rdfs:range)
    - …
- Damit: Inferenz in Typhierarchien

Filmontologie in RDFS




```
<rdfs:Class rdf:ID="person">
</rdfs:Class>
<rdfs:Class rdf:ID="regisseur">
  <rdfs:subClassOf rdf:resource="#person"/>
</rdfs:Class>
…
<rdfs:Property rdf:ID="fuehrtRegieIn">
  <rdfs:domain rdf:resource="#regisser"/>
  <rdfs:range rdf:resource="#film"/>
</rdfs:Property>
…
```

Einschätzung RDFS

- RDFS relativ nahe an objektorientierten Modellen
  - Und eher weiter weg von Description Logics
- Man kann z.B. nicht
  - Klassen auf Basis anderer Klassen definieren
    - Union, Schnitt, Komplement, …
  - Eigenschaften von Eigenschaften definieren
    - Transitivität, Symmetrie, …
- Formal ohne Trennung zwischen Modell und Metamodell
  - Eingebaute Sprachelemente können redefiniert werden
    - Z.B.: Range oder domain von rdfs:subClassOf einschränken
- Vorsitz: RDFS Validierung ist idR nicht strikt
  - RDFS definiert Klassen und deren Beziehungen
  - Die können in RDF benutzt werden

Inhalt dieser Vorlesung

- Grundidee des Semantic Web
- Layer Cake
- RDF und RDFS
- SparQL
- Die OWL Sprachfamilie

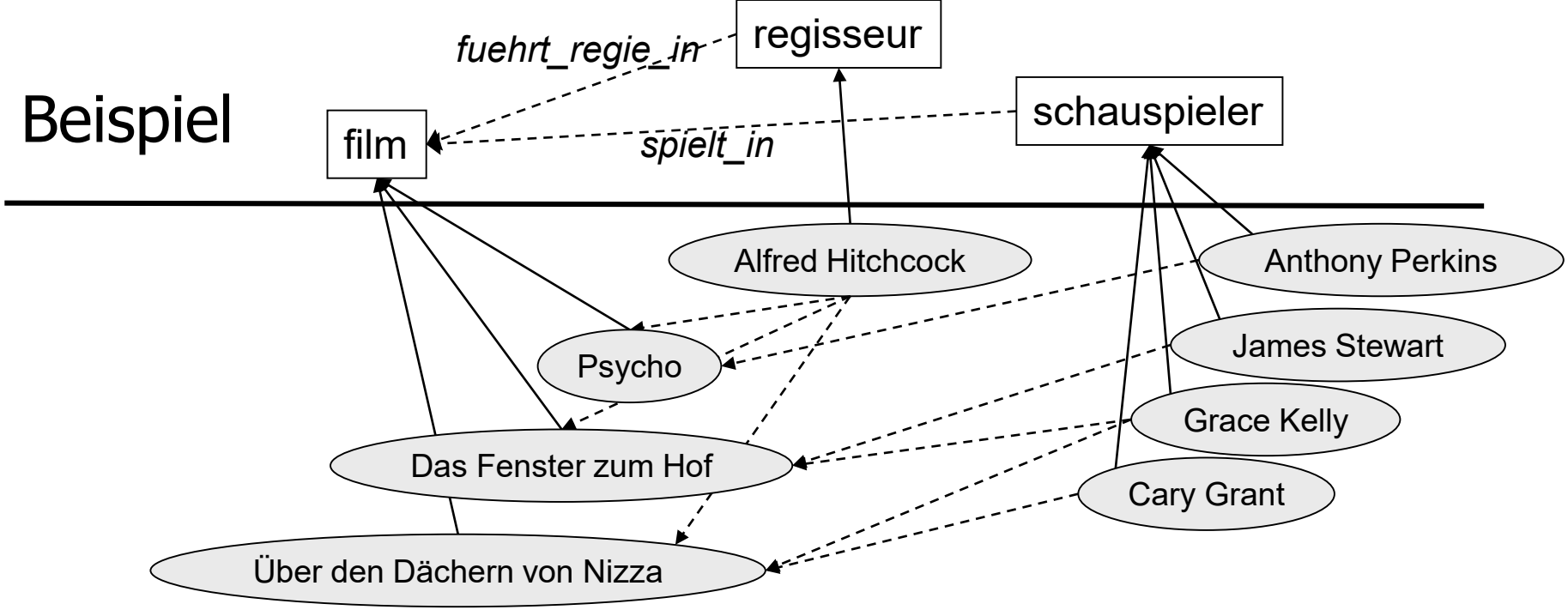
Anfragesprachen für RDF

- Historisch sind eine ganze Reihe von Sprachen entstanden
  - RQL, RDQL, SesamQL, …
- W3C standard: SPARQL
  - *SPARQL Protocol and RDF Query Language*
  - Wir behandeln nur den „QL" Teil
  - Eine SPARQL Anfrage Q ist im Kern ein Graphmuster aus Knoten und Kanten, beschriftet mit Konstanten oder Variablen
  - Q auf RDF-Datenbasis D: Alle zu Q isomorphen Subgraphen
- Grundkonzept: Anfragetripel (X Y Z)
  - X,Y,Z können Literale/URI's oder Variable sein
  - Anfragetripel Q matched ein RDF-Datentripel R, wenn es eine Funktion s gibt, die Konstante auf Konstante und Variable auf Konstante abbildet und für die gilt: s(Q)=R

SPARQL Grundaufbau

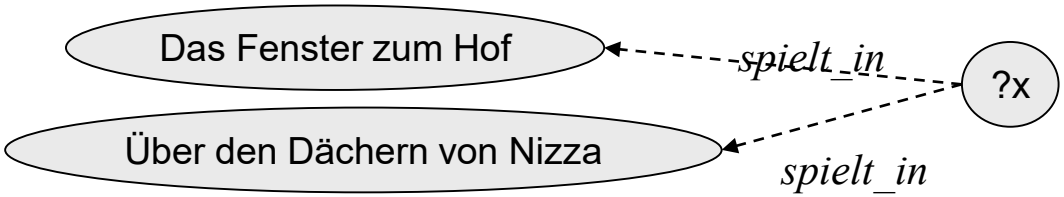
- Eine einfache SPARQL Anfrage ist eine Menge von Anfragetripeln
- Semantik
  - Anfragetripel werden an alle matchenden Datentripel gebunden
  - Bilde kartesisches Produkt aller Bindungen
  - Streiche alle Elemente, in denen eine Variable an verschiedene Werte gebunden wird
    - Gleiche Variable in verschiedenen Tripeln erzeugen also Joins
  - Alle übriggebliebenen Elemente bilden das ResultSet

Beispiel



- **SELECT ?X**
  WHERE (?X spielt_in „Über den Dächern von Nizza“
  ?X spielt_in „Das Fenster zum Hof“)

- Als Graph



- Berechnet: „Grace Kelly“

Erweiterungen

- WHERE Klausel
  - Optionals: Optionale Tripel
    - Wichtig wegen der fehlenden Strukturierung von RDF Daten
    - Vergleichbar Outer-Join (bzw. der Union von zwei Anfragen)
  - Filter für Wertebedingungen (=, <, >, REGEXP, …)
  - Union: Logisches ODER
- SELECT Klausel
  - Ausgabe von Variablenbindungen oder Tripelmengen
  - Sortierung der Ergebnisse
- FROM Klausel
  - Implizite Annahme einer Default RDF Datenbasis
  - Named Graphs – Queries über Tripel verschiedener Datenbasen

SparQL/RDF und Informationsintegration

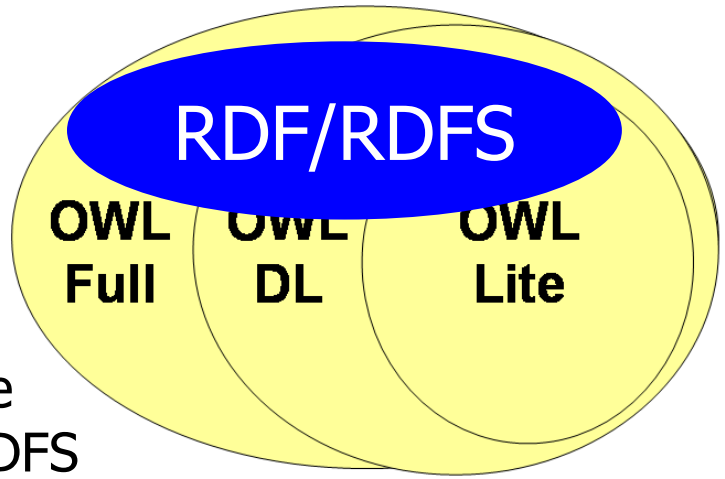
- Sehr flexibles Datenmodell
  - Subsumiert Tabellen (relational) und Bäume (XML)
  - Basiert auf RDF, nicht auf RDFS (oder OWL)
    - Keine Klassifikation / Beachtung von ISA Beziehungen
- Named Graphs
  - Verknüpfung mehrerer Datenquellen (Multi-DB-sprache)
- Schematische Heterogenität seltener
  - Knoten / Kanten
  - Ressource / Wert
- Optional und Union: Für strukturell heterogene Daten
  - Durch nur binäre Prädikate ist strukturelle Heterogenität selten
- Keine Gruppierung oder Aggregation
  - Schlecht für Duplikaterkennung und Fusion

Inhalt dieser Vorlesung

- Grundidee des Semantic Web
- Layer Cake
- RDF und RDFS
- SparQL
- Die OWL Sprachfamilie

Ontology Web Language: OWL

- Historisch: DAML + OIL → DAML+OIL → OWL
- Formal basierend auf der DL SHIQ
  - DL mit transitiven und inversen Rolleneigenschaften sowie numerischen Kardinalitätseinschränkungen
- „Eigentlich" fußt OWL auf RDF und RDFS
  - Erweiterung um Inferenz auch außerhalb der subClass-Beziehung
  - Aber nicht durchgehalten
- Drei Stufen
  - OWL Lite – Einfache Sprache für Taxonomien plus Constraints
  - OWL DL – Subsumption entscheidbar
  - OWL Full – Unentscheidbar, als einzige Sprache abwärtskompatibel zu RDF/RDFS



OWL Lite

- Trennung von Klassen, Werten und Instanzen
  - Die macht RDFS nicht
- Großteil der RDFS-Elemente
  - class, subClassOf, property, range, domain, …
- Verhältnisse von Konzepten (class) und Rollen (property)
  - Zwischen Klassen: intersectionOf
  - Zwischen Klassen oder zwischen Beziehungen: equivalent
  - Zwischen Instanzen: sameAs, differentFrom
- Eigenschaften von Rollen
  - inverseOf, transitive, symmetric, functional
- Rolleneinschränkungen
  - allValuesOf, someValuesOf, max/minCardinality (0 oder 1)
- Es fehlen z.B. $\sqcup$, $\neg$

OWL Full

- Abwärtskompatibel zu RDF
- Vermischung von Klassen, Instanzen, Rollen und Werten
  - Klassen können Instanzen anderer Klassen sein
  - Das wird von RDFS „geerbt“
- Weitere Sprachelemente
  - disjointWith: Schnitt zweier Klassen muss leer sein
  - unionOf, complementOf, intersectionOf für Klassen
  - …
- Subsumption unentscheidbar
  - Man kann Antinomien formulieren: „Die Klasse K aller Dinge, die nicht zu K gehören“ (aus [HPvH03])

OWL DL

- Gegenüber OWL Full
  - Trennung von Klassen, Instanzen, Rollen und Werten
  - Diverse Einschränkungen
- Gegenüber OWL Lite
  - Neue Sprachelemente (abgeleitete Konzepte, Kardinalitätseinschränkungen, …)
- Entscheidbare Subsumption
  - Ziel: „So ausdrucksstark wie gerade noch möglich“
  - Exponentielle Komplexität Problem bei grossen Ontologien

Zusammenfassung Semantic Web

- Einige Technologien sind da, das Ziel bleibt Vision
- Stärke: Flexibler Framework zur Beschreibung komplexer Daten und Hintergrundwissen über diese Daten
  - Mit internen Brüchen, insb. RDF/SparQL – RDFS - OWL
- Hindernisse
  - Welche Benutzer können Sprachen wie OWL lernen und einsetzen?
  - Wer soll all die Ontologien schreiben?
  - Ontologieheterogenität als neue Art Heterogenität?
  - Welchen Vorteil bietet es für einen Webseitenbetreiber, Daten als RDF zu publizieren?
  - Wie integriert man 100 Millionen verteilte RDF Datenquellen?