Pokročilé informační systémy

Sémantický web a ontologie

Doc. Ing. Radek Burget, Ph.D.
burgetr@fit.vutbr.cz

Sémantický web

- Iniciativa směřující k webu obsahujícím strojově zpracovatelné informace
  - Tim Berners Lee, cca. 1999
- Představa inteligentních aplikací „agentů“ umožňujících využívat informace na webu bez nutnosti číst celé dokumenty
  - Kontextové vyhledávání
  - Doplnění údajů k vyhledávání
  - …

Technické řešení

- Vývoj technologií pro vhodnou reprezentaci dat
  - Možnost sdílení dat i s jejich sémantikou
  - Použitelné technologie jsou již dlouho k dispozici
- Integrace s existujícím webem
  - Anotace ve webových stránkách
  - Poněkud vázne, ale zlepšuje se
- Viz přednáška UPA http://www.fit.vutbr.cz/~burgetr/upa/05_webscraping/#/35

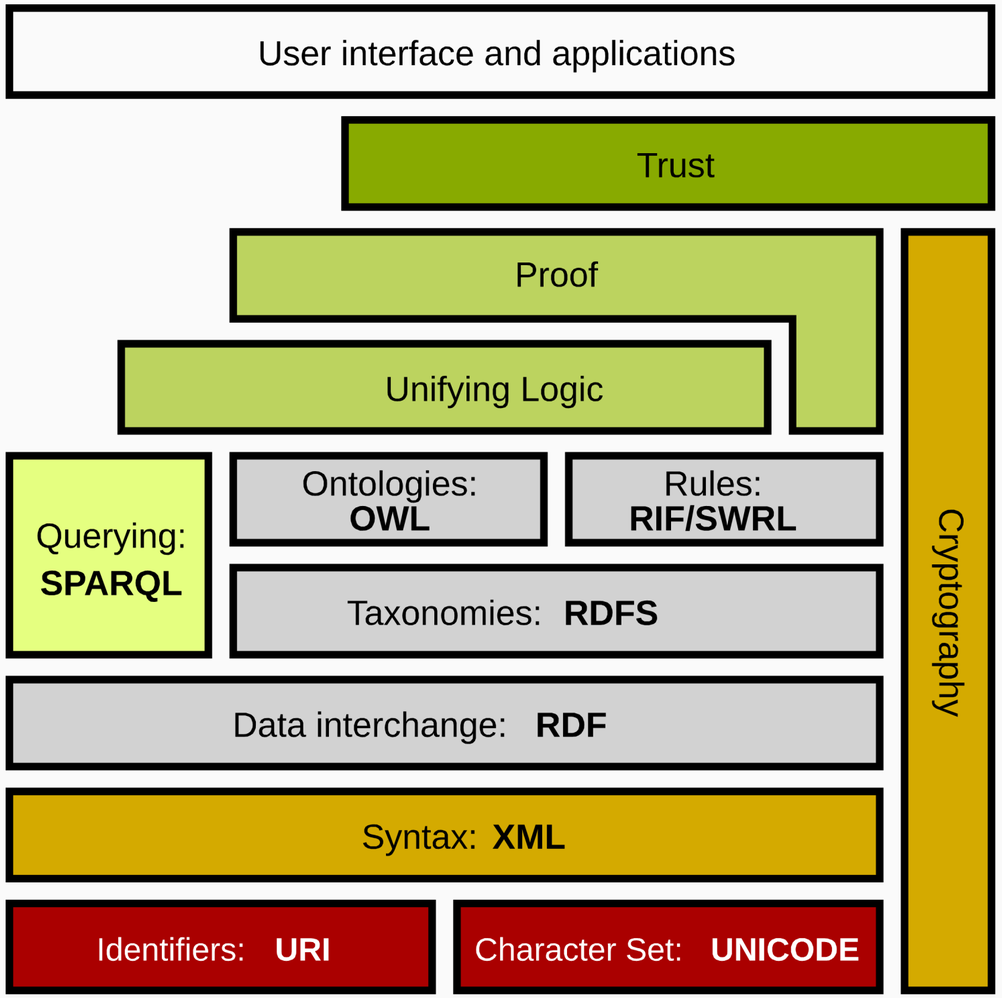
Web a sémantický web

- World Wide Web (web)
  - Základní jednotkou je dokument
  - „Web of documents“
- Semantic Web (sémantický web)
  - Základními jednotkami jsou data
  - „Web of Data“, „Linked data“

Technologie sémantického webu

- Technologie standardního webu
  - HTTP, URI
- Nástroje pro reprezentaci znalostí
  - Reprezentace dat (faktů)
    - XML, RDF, …
  - Sémantika
    - Ontologie
    - Technologie pro reprezentaci ontologie

Semantic Web Stack



Sémantický web – RDF

Reprezentace a výměna faktů

Cíle a prostředky

- Cíle
  - Reprezentace strukturovaných dat a jejich významu (sémantiky)
  - Možnost sdílet data a jejich sémantiku napříč aplikacemi
- Běžná reprezentace dat v IS:
  - Relační/objektové/NoSQL databáze – vázané na aplikaci
  - Veřejné API + serializace (JSON, XML) – není definována sémantika

Serializace – příklad

```
<nabidka>
    <polozka>
        <velikost>3+1</velikost>
        <lokalita>Brno-střed</lokalita>
        <cena mena="czk">2 200 000</cena>
    </polozka>
    <polozka>
        <velikost>2+1</velikost>
        <lokalita>Kuřim</lokalita>
        <cena mena="czk">450 000</cena>
    </polozka>
</nabidka>
```

Problémy

- Význam elementů je specifický pro danou aplikaci
  - Je definován v programovém kódu, který generuje nebo načítá serializovaná data
  - Obdobně jako např. sloupce v relační databázi
- Jiná aplikace může stejným elementům přiřadit jiný význam
  - Např. `<velikost>2+1<velikost>` vs. `<velikost>55m2</velikost>`
- Data jsou strojově čitelná (machine readable), ale ne srozumitelná (machine understandable)

Reprezentace sémantiky

- Odlišení značek v různých aplikacích
  - Např. XML namespaces
  - Řeší kolize značek – syntaktický problém
- Oddělená definice významu značek
  - Např. doprovodný dokument vysvětlující význam a případy použití
- Navíc ale potřebujeme definovat sémantické vztahy
  - Např. byt je věc, která má umístění, velikost a cenu
  - Pokud možno formálně => **Ontologie**

Reprezentace faktů

- XML
  - Mapování elementů na vlastnosti ontologií
  - Pouze hierarchická struktura – omezující
- RDF
  - Grafová struktura
  - Lze zapsat pomocí XML nebo jiných jazyků

RDF

- RDF je datový model standardizovaný W3C
  - Zaměřeno na data sdílená na venek
  - Snadné propojení dat z různých zdrojů a na různých schématech (linked data) (http://lod-cloud.net/)
- Lze integrovat do webových dokumentů
- Existují různá úložiště
  - RDF úložiště napsané v Javě http://rdf4j.org/
    - Původně známé jako *Sesame*, nyní pod Eclipse Foundation
  - Blazegraph (Java) https://www.blazegraph.com/
  - OpenLink Virtuoso (C++) https://virtuoso.openlinksw.com/

RDF trojice

- Základním prvkem je **RDF trojice**

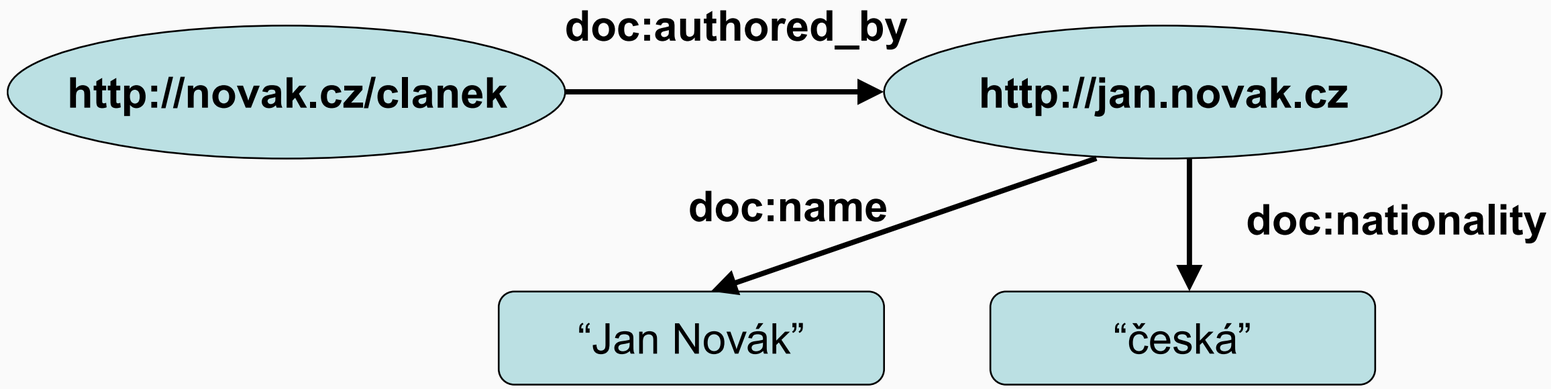
subjekt – predikát – objekt

RDF trojice – tvrzení (statement)

- *Autorem* **dokumentu X** je **pan Y**
  - Subjekt: **dokument X**
  - Predikát: *je autorem*
  - Objekt: *pan Y*
- Jednotlivé části tvrzení (zdroje) *(resources)* jsou reprezentované pomocí **URI** nebo **literálem**.

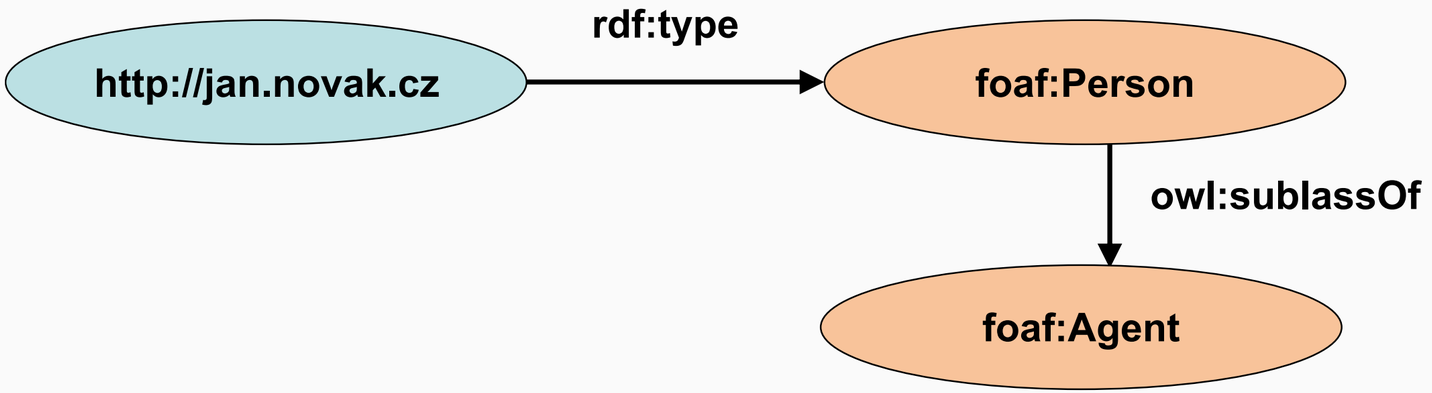
RDF tvrzení (II)

RDF Graf



- RDF graf lze rozložit na trojice subjekt – predikát – objekt
- Subjekt a predikát jsou vždy URI
  - doc: je prefix URI, který se expanduje
  - Např. doc:name => http://my.docs.com/#name
- Objekt je URI nebo literál (různých datových typů)

Schéma – Ontologie



- RDF data lze propojit s metadaty (ontologií, schématem)
  - Pomocí predikátu rdf:type
- Definice metadat opět pomocí RDF
  - Je možné (ale ne nutné) spojit data i metadata do jednoho grafu.

Ukládání a přenos RDF dat

- Uložení do RDF úložiště (např. RDF4J)
  - Rozložení na trojice a uložení do interní struktury
  - Následně možnost dotazování (jazyk SPARQL)
- Serializace do souboru a zpět – několik variant
  - RDF/XML (standard W3C)
  - N-triples (N3)
  - Turtle (podmnožina N3)

Serializace do Turtle

```
@prefix doc: <http://dokumenty.cz/def#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

<http://novak.cz/clanek>
    doc:authored-by <http://jan.novak.cz> .

<http://jan.novak.cz>
    doc:name "Jan Novák" ;
    doc:nationality "česká" ;
    a foaf:Person .
```

XML Serializace

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:doc="http://dokumenty.cz/def\#">

    <rdf:Description rdf:about="http://novak.cz/clanek">
        <doc:authored-by
            rdf:resource="http://jan.novak.cz" />
      </rdf:Description>

      <rdf:Description rdf:about="http://jan.novak.cz">
        <doc:name>Jan Novák</doc:name>
        <doc:nationality>česká</doc:nationality>
```

RDF jako databáze

- Repozitář – úložiště RDF trojic
- Dotazování – jazyk SPARQL
- Lokální úložiště:
  - Virtuoso http://virtuoso.openlinksw.com/
  - RDF4J (dříve Sesame) http://rdf4j.org/
  - Blazegraph https://www.blazegraph.com/product/
- Globální
  - DBPedia http://dbpedia.org
  - http://dbpedia.org/resource/Berlin
  - http://dbpedia.org/sparql

RDF4J

- Java API (embedded) nebo samostatně běžící server přístupný přes HTTP REST API
- Různé druhy úložišť
  - Memory, Native, relační databáze, rozšiřitelné o další
- Strategie vyhodnocování SPARQL dotazů
  - Možnost implementace vlastní strategie
- Podpora kontextu (RDF čtveřice)
- Podpora transakcí

Dotazování – SPARQL

- Výsledkem dotazu je
  - CSV (tabulka) – dotaz SELECT
  - Nebo nový graf – dotaz CONSTRUCT

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX yago: <http://dbpedia.org/class/yago/>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
PREFIX dbprop: <http://dbpedia.org/property/>
SELECT ?place ?name ?label WHERE {
    ?place rdf:type dbpedia-owl:Country .
    ?place dbprop:commonName ?name .
    ?place rdfs:label ?label .
   OPTIONAL {?place dbprop:yearEnd ?yearEnd}
   FILTER (!bound(?yearEnd))
}
```

Veřejné báze znalostí

- DBPedia http://dbpedia.org
  - http://dbpedia.org/resource/Berlin
  - http://dbpedia.org/sparql
- Wikidata http://wikidata.org
  - http://wikidata.org/entity/Q42
- Mnoho dalších
  - http://lod-cloud.net/

Ontologie

Slovníky pro sémantický web

Pojem ontologie

- Původně obecnější význam (filozofie)
- Nástroj pro sdílení významu pojmů, které se vyskytují v cílové oblasti
- *„Formální, explicitní specifikace sdílené konceptualizace“*
- Definují základní pojmy modelovaného světa a vztahy mezi nimi
- Sdílené a opakovatelně použitelné

Účel ontologií

- Porozumění mezi lidmi (experty)
- **Porozumění mezi počítačovými aplikacemi**
  - **Dodání významu jednotlivým URI v sémantickém webu**
  - Možnost **integrace** dat z různých zdrojů
- Návrh znalostních aplikací

Typy ontologií

- Terminologické (lexikální)
  - Seznam termínů v dané oblasti
  - Jejich vzájemné vztahy (taxonomie)
  - Např. *WordNet*
- Informační ontologie
  - Databázové systémy – pokročilejší schémata
- Znalostní ontologie
  - Aplikace umělé inteligence
  - Koncepty formálně definované pomocí logických formulí

Typy ontologií (II)

- Generické ontologie
  - Zákonitosti a vztahy mezi obecnými pojmy
  - „Upper ontology“, např. SUMO
- Doménové ontologie
  - Konkrétní oblast (např. podnikové, lékařství, …)
- Aplikační ontologie
  - Pro konkrétní aplikaci

Prvky ontologií

- **Třídy (koncepty)**
- **Individua (objekty, instance)**
- **Vlastnosti (role, atributy)**
- Meta-sloty (facety)
- Primitivní datové typy
- Axiomy (pravidla)

Definované prvky můžeme využít v RDF tvrzeních. Ontologie tedy definuje *slovní zásobu* (*vocabulary*) pro RDF.

Koncepty – třídy

- Množiny konkrétních objektů
- Žádné procedurální metody
- Třídy *definované* a *primitivní*
  - Podle definice příslušnosti individua
- Dědičnost tříd (často vícenásobná)

Individua – objekty – instance

- Konkrétní objekty reálného světa
- Individuum nemusí být nutně instancí třídy
- Vzhledem k určení ontologií se často nepoužívají
  - Reprezentují konkrétní data

Relace – atributy – sloty – vlastnosti

- Pojetí vlastnosti je jiné, než u OO modelování
- Vlastnost = relace
  - Samostatně definovaný prvek
  - Obvykle binární relace
- Možná dědičnost relací (má otce, má předka)
  - Nadřazená relace obsahuje všechny prvky podřazené relace
- Funkce – speciální relace
  - Hodnota argumentu *n* jednoznačně určena předchozími *n-1* argumenty

Meta-sloty, omezení na sloty

- Vlastnosti vlastností
  - Vztah podřízená – nadřízená vlastnost
- Globální omezení
  - Definiční obor a obor hodnot vlastnosti
- Lokální omezení – *facet*
  - Např. kardinalita
  - Hodnota vlastnosti **má-otce** aplikované na třídu **osoba** je **právě jedna** instance třídy **osoba**.

Primitivní hodnoty, datové typy

- Argumentem relace může být *primitivní hodnota* (ne objekt)
  - Číslo, řetězec, výčtová hodnota, …
  - Datatype slot vs. objektový slot
- Můžeme uvažovat dato-typové třídy (datové typy) a dato-typové instance (hodnoty)
- Dato-typové sloty obvykle deklarujeme jako funkční (mají pouze jednu hodnotu)

Axiomy, pravidla

- Logické formule vymezující vztahy tříd
  - Ekvivalence, subsumpce
- Obvykle součást definice tříd

Ontologické jazyky

RDF Schema, OWL

RDF Schema

- Sémantické rozšíření RDF
  - V podstatě (meta) **ontologie**
- Umožňuje definici
  - Tříd
  - Binární relace (definiční obor, obor hodnot)
  - Hierarchie nad třídami i relacemi
- Definice opět pomocí RDF tvrzení (trojic)
  - Např. `skola:Student rdfs:subClassOf skola:Osoba`
- Namespace (prefix obvykle **rdfs**)

  `http://www.w3.org/2000/01/rdf-schema#`

Třídy

- Třída je přiřazena ke zdroji pomocí rdf:type
  - `skola:Osoba rdf:type rdfs:Class`
  - V XML:

```
<rdf:Description rdf:about="&skola;Osoba">
<rdf:type rdf:resource="&rdfs;Class" />
</rdf:Description>
```

  - Nebo

```
<rdfs:Class rdf:about="&skola;Osoba" />
```

Odvozené třídy

- Podtřídy rdfs:subClassOf
  - Např. `skola:Student rdfs:subClassOf skola:Osoba`
  - V XML

```
  <rdfs:Class rdf:about="&skola;Student">
   <rdfs:subClassOf rdf:resource="&skola;Osoba" />
</rdfs:Class>
```

Třídy v RDFS

- `rdfs:Class` – třída (je instancí `rdfs:Class`)
- `rdfs:Resource` – třída jakéhokoliv zdroje
  - instance `rdfs:Class`
- `rdfs:Literal`
  - instance `rdfs:Class`
  - Podtřída `rdfs:Resource`
- `rdfs:Datatype`
  - instance i podtřída `rdfs:Class`
  - každá instance `Datatype` je podtřídou `Literal`

Třídy v RDFS (II)

- `rdfs:XMLLiteral`
  - instance `rdfs:Datatype`
  - podtřída `rdfs:Literal`
- `rdfs:Property`
  - instance `rdfs:Class`

Vlastnosti v RDFS

- Vlastnosti jsou instance `rdfs:Property`
  - `skola:maZapsano rdf:type rdfs:Property`
- **rdfs:Range** – typ objektů (obor hodnot)
  - `skola:maZapsano rdfs:range skola:Predmet`
- **rdfs:Domain** – typ subjektů (def. obor)
  - `skola:maZapsano rdfs:domain skola:Student`
- `rdfs:subPropertyOf`
  - Vlastnost je „podvlastností“ jiné vlastnosti

OWL

- Rozšíření RDFS o pokročilé vlastnosti
- Různé verze
  - OWL Lite – zjednodušená, kvůli implementaci
  - OWL DL – omezení RDF(S) pro podporu DL
  - OWL Full – max. kompatibilita s RDF(S)
- Namespace `http://www.w3.org/2002/07/owl#`

Definice tříd v OWL

- Kombinace s RDFS
- Třídu lze definovat pomocí logických podmínek
  - Identifikátorem třídy (žádné prvky)
  - Výčtem prvků (instancí)
  - Omezením vlastností
  - Sjednocením nebo průnikem dvou a více tříd
  - Doplňkem

Definice ontologie

```
<owl:Ontology rdf:about="">
    <rdfs:comment>An example OWL ontology</rdfs:comment>
    <owl:priorVersion>
          <owl:Ontology
              rdf:about="http://www.w3.org/TR/2003/WD-owl-guide-20030331/win
    </owl:priorVersion>
    <owl:imports rdf:resource="http://www.w3.org/TR/2003/CR-owl-guide-200308
    <rdfs:label>Wine Ontology</rdfs:label>
</owl:Ontology>
```

Definice třídy identifikátorem

Turtle

```
foaf:Person rdf:type owl:Class .
```

XML

```
<owl:Class rdf:about="&foaf;Person"/>
```

nebo

```
<rdf:Description rdf:ID="Person">
    <rdf:type resource="&owl;Class" />
</rdf:Description>
```

V Turtle s prefixy

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix foaf:<http://xmlns.com/foaf/0.1/>.

foaf:Person rdf:type owl:Class .
foaf:Person a owl:Class .
```

Další možnosti definice tříd

- Výčtem prvků
- Průnikem, sjednocením, doplňkem

Definice doplňkem

```
<owl:Class>
  <owl:complementOf>
    <owl:Class rdf:about="#Student"/>
  <owl:complementOf/>
</owl:Class>
```

Ostatní operátory nad třídami

- `owl:equivalentClass`
  - Stejná třída (např. z jiné ontologie)
- `owl:disjointWith`
  - Disjunktní třída

Definice vlastností

- RDFS konstruktory

```
<owl:ObjectProperty rdf:ID="studuje">
  <rdfs:domain rdf:resource="#Student"/>
  <rdfs:range rdf:resource="#Obor"/>
</owl:ObjectProperty>
```

- Vztahy mezi vlastnostmi
  - `owl:equivalentProperty` – stejné hodnoty
  - `owl:inverseOf` – inverzní vlastnost

```
<owl:ObjectProperty rdf:ID="maStudenta">
    <owl:inverseOf rdf:resource="#studuje"/>
</owl:ObjectProperty>
```

Definice vlastností (II)

- Omezení kardinality

```
<owl:FunctionalProperty rdf:about="studuje"/>
```

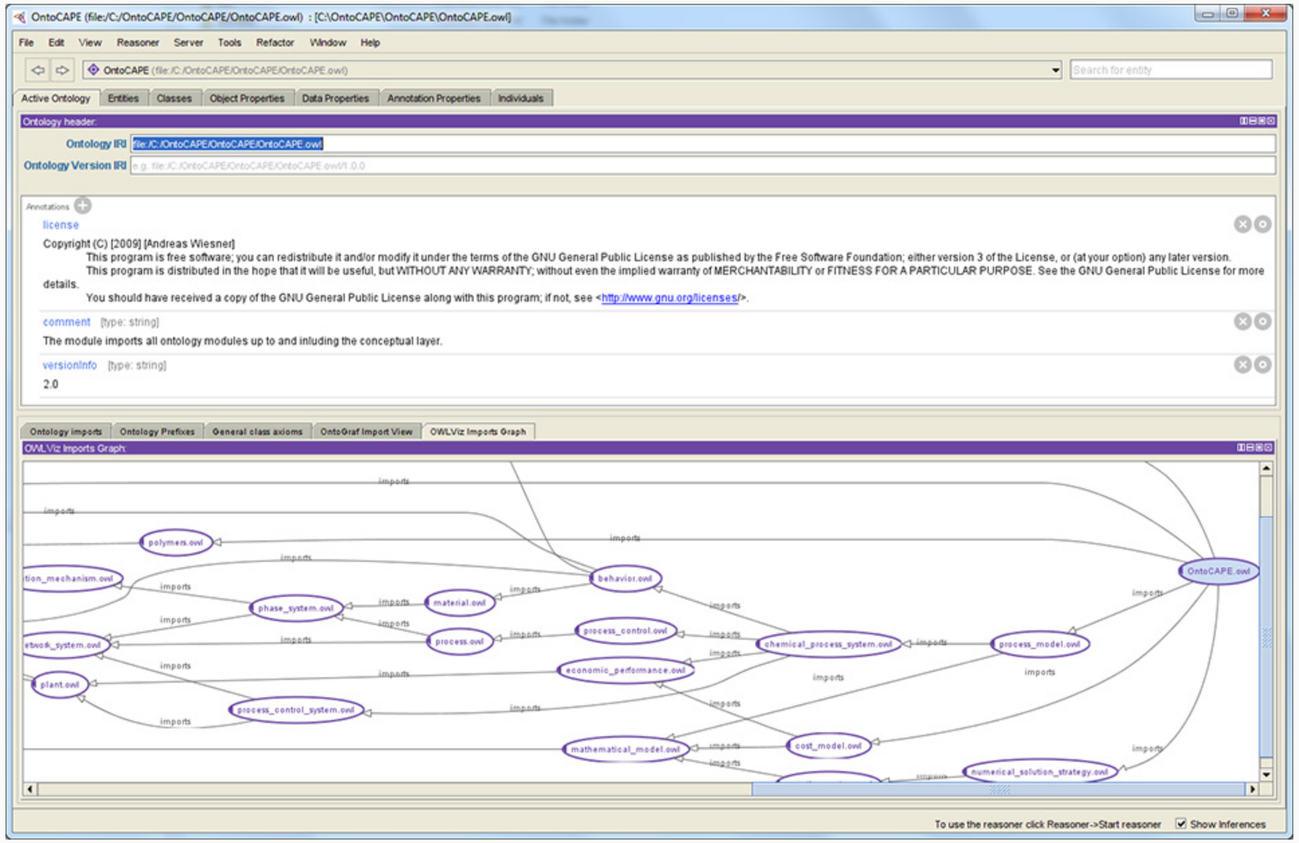
- Symetrická vlastnost
  - `owl:SymetricProperty`
- Tranzitivní vlastnost
  - `owl:TransitiveProperty`

Data-typové vlastnosti

- RDF Literály
- XSD datové typy
  - Namespace `http://www.w3.org/2001/XMLSchema`
- `xsd:string`, `xsd:normalizedString`, `xsd:boolean`, `xsd:decimal`, `xsd:float`, `xsd:double`, `xsd:integer`, `xsd:nonNegativeInteger` `xsd:positiveInteger`, `xsd:nonPositiveInteger`, `xsd:negativeInteger`, `xsd:long`, `xsd:int`, `xsd:short`, `xsd:byte`, `xsd:unsignedLong`, `xsd:unsignedInt`, `xsd:unsignedShort`, `xsd:unsignedByte`, `xsd:hexBinary`, `xsd:base64Binary`, `xsd:dateTime`, `xsd:time`, `xsd:date`, `xsd:gYearMonth`, `xsd:gYear`, `xsd:gMonthDay`, `xsd:gDay`, `xsd:gMonth`, `xsd:anyURI`, `xsd:token`, `xsd:language`, `xsd:NMTOKEN`, `xsd:Name`, `xsd:NCName`

Editor Protegé

http://protege.stanford.edu/



Existující ontologie

- Důraz na maximální využití existujících ontologií
  - Je možno kombinovat koncepty a vlastnosti z různých ontologií
- Přehled
  - https://lov.linkeddata.es/dataset/lov/

Dublin core

- Metadata dokumentů
- Použití zejména v knihovnictví
- Definuje vlastnosti dokumentů:

```
<rdf:Description rdf:about="http://www.w3schools.com">
   <dc:description>W3Schools</dc:description>
   <dc:publisher>Refsnes Data as</dc:publisher>
   <dc:date>2008-09-01</dc:date>
   <dc:type>Web Development</dc:type>
   <dc:format>text/html</dc:format>
   <dc:language>en</dc:language>
 </rdf:Description>
```

Friend-of-a-friend (FOAF)

- Ontologie pro popis osob a jejich vzájemných vztahů http://www.foaf-project.org/
- Třídy pro popis osob
  - `foaf:Agent`, `foaf:Person`, …
- Vlastnosti
  - `foaf:name`, `foaf:knows`, …

FOAF příklad

```
@prefix foaf:<http://xmlns.com/foaf/0.1/>.
@prefix dbr:<http://dbpedia.org/resource>.

dbr:Luke_Skywalker foaf:knows dbr:Han_Solo .
dbr:Luke_Skywalker foaf:name "Luke Skywalker" .
```

SKOS

- Simple Knowledge Organization System
- Umožňuje organizaci pojmů v nějaké doméně
  - Koncepty: Concept
  - Vztahy mezi nimi: broader, narrower, related, …
  - …

Schema.org

- Primárně pro anotování webových stránek
  - https://schema.org
- Základní slovníky pro různé obecné domény
  - https://schema.org/docs/gs.html#schemaorg

Další ontologie

- Music ontology
  - http://musicontology.com/
- Event ontology
  - http://motools.sourceforge.net/event/event.html
- Time ontology
  - http://www.w3.org/TR/2006/WD-owl-time-20060927/
- Geo ontology
  - http://www.w3.org/2003/01/geo/

Ontologie a RDF znalostní báze

- DBPedia.org
  - Vlastní ontologie + použití existujících
  - http://dbpedia.org/resource/Berlin
  - http://dbpedia.org/page/Novak_Djokovic
- Např.
  - Vlastnost Birth place
  - Podobně Wikidata

SPARQL

- Jazyk pro dotazování v RDF datech
- Syntax SQL + Turtle
- `SELECT ?x ?y WHERE { vzor RDF stromu }`
  - Výsledkem je tabulka (CSV)
- `CONSTRUCT { ?x :prop ?y } WHERE { … }`
  - Výsledkem je jiný RDF graf
- Např.
  - http://dbpedia.org/sparql

A to je vše!

Dotazy?