

OWL

Iztok Savnik

Pregled

- Omejitve RDFS
 - Izrazna moč
 - Problemi s plastmi
- Jezik OWL
 - Ontologije
 - Zasnova OWL
 - OWL plasti
 - OWL in opisna logika
 - OWL sintaksa

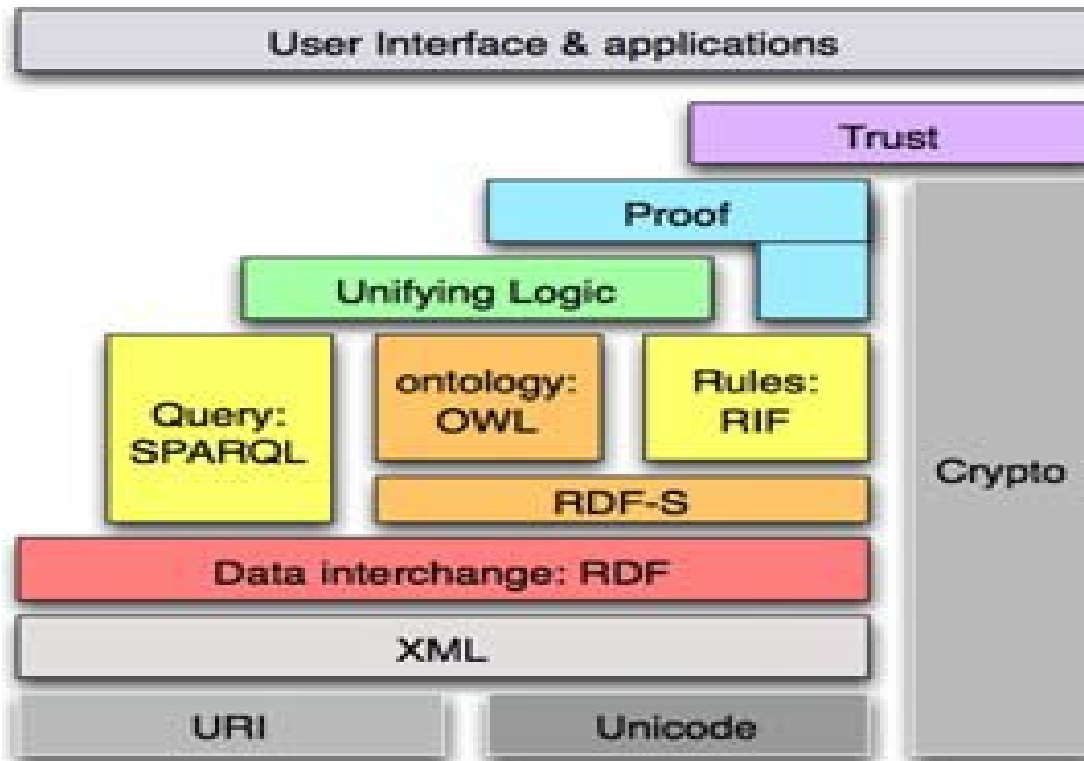
RDFS kot jezik za predstavitev ontologij

- Razredi
- Lastnosti
- Hierarhije razredov
- Hierarhija lastnosti
- Omejitve domene in zaloge vrednosti

Plasti RDF

- Sintaksa
 - Binarne relacije v RDF
 - Ni omejitev nad grafi
 - Vsi grafi so veljavni
- Semantika
 - Izrojeni grafi
 - Uporaba slovarja v jeziku
 - `rdfs:Class`, `rdfs:subClassOf`, itd
 - Meta-razredi
 - `ex:a`, `rdf:type`, `ex:a`

Kje smo?



Sklad jezikov

- XML
 - Sintaksa, ni semantike
- XML shema
 - Opisuje strukture XML dokumenti
- RDF
 - Podatkovni model za “relacije” med “objekti”
- RDF shema
 - RDF jezik za definicijo slovarjev
 - Podatkovni model
- OWL
 - Logika!

Pregled RDF sheme

- RDFS **omogoča**
 - razredi
 - hierarhija razredov
 - lastnosti
 - hierarhija lastnosti
 - omejitve domene in zaloge vrednosti
- RDFS **ne omogoča**
 - značilnosti lastnosti (inverz, tranzitivnost, ...)
 - lokalne omejitve zaloge vrednosti
 - definicija kompleksnih konceptov
 - kardinalnost
 - aksiomi o praznem preseku

Razširitev RDF sheme

- OWL razširi RDF shemo na kompleten jezik za predstavitev znanja in podatkov na Web
 - logični izrazi (and, or, not)
 - (ne)enakost
 - lokalne lastnosti
 - obvezne/neobvezne lastnosti
 - obvezne vrednosti
 - naštevni razredi
 - simetrija, inverzi

Načrtovalski cilji OWL

- Na razpolago (za Web)
- Vsebina se spreminja skozi čas
- Interoperabilnost
- Odkrivanje inkonsistence
- Usklajevanje med izrazno močjo in učinkovitostjo
- Enostaven za uporabo
- Kompatibilno z obstoječimi standardi
- Internacionalizacija

Zahteve OWL

- Ontologije so objekti na Web
- Meta-podatki, verzije, itd...
- Ontologije so razširljive
- Razredi, lastnosti, podatkovni tipi, domene/zaloge vrednosti, objekti
- Enakost (razredi, objekti,...)
- Razredi kot instance
- Kardinalnost
- XML sinatksa

Namen OWL

- Plasti
- Kompleksni podatkovni tipi
- Digitalne signature
- Odločljivost (delno)
- Unikatna imena lokalno (delno)

Ne obravnava:

- Privzete vrednosti
- Predpostavka o zaprtem svetu
- Veriženje lastnosti
- Aritmetika
- Operacije nad nizi
- Delno uvažanje
- Definicija oken
- Postopkovno
- Priponke (attachments)

Jezikovni nivoji OWL

- OWL Lite
 - klasifikacijska hierarhija
 - enostavne omejitve
- OWL DL
 - maksimalna izraznost
 - ohranitev izračunljivosti
 - standardna formalizacija z DL
- OWL Full
 - zelo velika izraznost
 - ni izračunljiv
 - sintaktična svoboda RDF

Značilnosti OWL nivojev

OWL Lite

- (sub)razredi, objekti
- (sub)lastnosti, domena, zaloga vrednosti
- konjunkcija
- (ne)enakost
- kardinalnost 0/1
- podatkovni tipi
- inverzi, tranzitivnost, simetrične lastnosti
- someValuesFrom
- allValuesFrom

OWL DL

- negacija
- disjunkcija
- kompletna kardinalnost
- naštrevni tipi
- hasValue

OWL Full

- Meta-razredi
- spreminjanje podatkov

OWL Lite

- Ni eksplicitne negacije ali unije
- Omejena kardinalnost (0/1)
- Ni imenskih tipov (oneOf)
- Semantika osnovana na DL
 - sklepanje na osnovi DL sistemih (+podatkovni tipi)
- Semantično samo majhna omejitev OWL DL
 - ni imenskih tipov
 - ni poljubne kardinalnosti

OWL DL

- Uporaba slovarja je omejena
 - Ne more biti uporabljena za “grde trike” (npr. za spremembo OWL)
 - Razredi ne morejo biti instance
- Uporabljena teorija modelov osnovana za DL
 - Direktna preslikava iz DL
 - Sklepanje z DL sistemi

OWL Full

- Ni omejitev glede uporabe slovarja (dokler je legalno znotraj RDF)
 - Razredi in objekti (in več...)
- Teorija modelov za RDF
 - Sklepanje z uporabo FOL sistema
 - Semantika naj ustreza OWL DL za omejene baze znanja

OWL konstrukti

OWL konstrukt	DL	Primer
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	Human \sqcap Male
unionOf	$C_1 \sqcup \dots \sqcup C_n$	Doctor \sqcup Lawyer
complementOf	$\neg C$	\neg Male
oneOf	$\{o_1, \dots, o_n\}$	{john, mary}
allValuesFrom	$\forall P.C$	\forall hasChild.Doctor
someValuesFrom	$\exists P.C$	\exists hasChild.Lawyer
value	$\exists P.\{o\}$	\exists citizenOf .USA
minCardinality	$\geq n P.C$	≥ 2 hasChild.Lawyer
maxCardinality	$\leq n P.C$	≤ 1 hasChild.Male
cardinality	$= n P.C$	$= 1$ hasParent.Female

+ XML Schema tipi: int, string, real, etc...

Dve sintaksi OWL

- RDF/XML documents
 - <http://www.cs.man.ac.uk/~horrocks/ISWC2003/Tutorial/people+pets.owl.rdf>
 - OWL je del Semantičnega spleta !
 - OWL naj bo razširitev RDF !
 - RDF aplikacije lahko delajo z OWL
- “abstraktna” sintaksa
 - <http://www.cs.man.ac.uk/~horrocks/ISWC2003/Tutorial/people+pets.abs>
 - enostavneje pisati in brati
 - bližje opini logiki in okvirjem

Peter FP Schneider

Razredi

- Primeri: person, pet, old
- Kolekcija individualnih objektov (stvari, ...)
- Način opisovanja dela sveta
- Objekt iz dane domene (OWL Full)

Primeri razredov

```
Class(pp:animal partial
      restriction(pp:eats someValuesFrom(owl:Thing)))
Class(pp:person partial pp:animal)
Class(pp:man complete
      intersectionOf(pp:person pp:male pp:adult))
Class(pp:animal+lover complete
      intersectionOf(pp:person
                    restriction(pp:has_pet minCardinality(3))))
```

Primeri razredov

```
Class(pp:vegetarian complete
      intersectionOf(pp:animal
                    restriction(pp:eats
                              allValuesFrom(complementOf(pp:animal)))
                    restriction(pp:eats
                              allValuesFrom(
                                complementOf(restriction(pp:part_of
                                                         someValuesFrom(pp:animal)))))))
DisjointClasses(pp:young pp:adult)
```

Lastnosti

- Kaj je lastnost?
 - Ima-očeta, ima-psa, številka-servisa, ...
 - Kolekcija razmerij med objekti (in podatki)
 - Način opisa neko vrsto razmerja med posamezniki
 - Objekt iz danega sveta (OWL Full)

Primeri lastnosti

ObjectProperty(pp:eaten_by)

ObjectProperty(pp:eats inverseOf(pp:eaten_by)
domain(pp:animal))

ObjectProperty(pp:has_pet domain(pp:person)
range(pp:animal))

ObjectProperty(pp:is_pet_of
inverseOf(pp:has_pet))

DataProperty(pp:service_number
range(xsd:integer))

SubPropertyOf(pp:has_pet pp:likes)

Primerki

- Objekti iz danega sveta
- Primerki razreda
- Povezani so z ostalimi objekti in podatkovnimi vrednostmi preko lastnosti

Instance razredov

```
Individual(pp:Tom type(owl:Thing))
Individual(pp:Dewey type(pp:duck))
Individual(pp:Rex type(pp:dog)
           value(pp:is_pet_of pp:Mick))
Individual(pp:Mick type(pp:male)
           value(pp:reads pp:Daily+Mirror)
           value(pp:drives pp:Q123+ABC))
Individual(pp:The42 type(pp:bus)
           value(pp:service_number "42"^^xsd:integer))
```

Kaj sledi iz primera?

```
Class(pp:old+lady complete
      intersectionOf(pp:elderly pp:female pp:person))
Class(pp:old+lady partial
      intersectionOf(
        restriction(pp:has_pet allValuesFrom(pp:cat))
        restriction(pp:has_pet someValuesFrom(pp:animal))))
```

Kaj sledi iz primera?

```
Class(pp:cow partial pp:vegetarian)
Class(pp:mad+cow complete
  intersectionOf(pp:cow restriction(pp:eats
    someValuesFrom(intersectionOf(pp:brain
      restriction(pp:part_of
        someValuesFrom pp:sheep))))))
```

Krave kot vegetarianci ne jejo nič kar je del živali.
Norih krav ni!

Kdo je Tom?

```
ObjectProperty(pp:has_pet domain(pp:person)  
               range(pp:animal))
```

```
Class(pp:old+lady complete  
      intersectionOf(pp:elderly pp:female pp:person))
```

```
Class(pp:old+lady partial  
      intersectionOf(restriction(pp:has_pet  
                                allValuesFrom(pp:cat))  
                    restriction(pp:has_pet someValuesFrom(pp:animal))))
```

```
Individual(pp:Minnie type(pp:elderly) type(pp:female)  
value(pp:has_pet pp:Tom))
```

Minnie je oseba.

Minnie je stara gospa.

Tom je maček.

Kaj sledi iz naslednje ontologije?

```
Class(pp:van partial pp:vehicle)
Class(pp:driver partial pp:adult)
Class(pp:driver complete
      intersectionOf(restriction(pp:drives
                                someValuesFrom(pp:vehicle))
                    pp:person))
Class(pp:white+van+man complete
      intersectionOf(pp:man
                    restriction(pp:drives
                                someValuesFrom(intersectionOf(pp:white+thing pp:van))))))
Class(pp:white+van+man partial
      restriction(pp:reads allValuesFrom pp:tabloid))
```

Kaj sledi iz naslednje ontologije?

```
Individual(pp:Q123+ABC type(pp:white+thing)
  type(pp:van))
Individual(pp:Mick type(pp:male)
  value(pp:reads pp:Daily+Mirror)
  value(pp:drives pp:Q123+ABC))
```

Mick vozi kombi, zato mora biti odrasel.

(ker so vsi vozniki odrasli).

Ker je Mick moški in je tudi beli-kombi-mož,
bere tabloide, zato je Daily Mirror tabloid.

OWL pogled na življenje

- OWL ni sistem za delo s podatkovnimi bazami
 - Kaj je potem ?
 - Dodamo nove operacije v SQL DBMS ? ;)
- Razlogi za “stand-alone” sistem
 - Ni zahtev po tem, da so edine lastnosti objekta so tiste, ki so zapisane.
 - Ni predpostavke o tem, da je vse znano.
 - Razredi in lastnosti imajo lahko več “definicij”.
 - Stavki o posameznikih niso nujno skupaj v istem dokumentu.

Razlogi za ločen sistem

- Stroj za izvajanje **mehanizmov sklepanja**
- Deloma nekateri **novi aspekti podatkovnih modelov**
 1. ISA hierarhija lastnosti.
 2. Lastnosti lastnosti, ...
 3. First-class-citizen gradniki FOL v DL: ,
(SQL exists!)
 4. Urejenost (poset) med vsemi objekti:
 - Precej kompleksna implementacija.
 - Zelo težko spraviti v resen industrijski stroj
- Vmesnik do DBMS

Uporaba OWL (gradnja ontologij)

- Definiraj kako naj svet (domena) izgleda
 - Definiraj razrede in lastnosti v dani domeni
 - Definiraj domene in zaloge vrednosti lastnosti
 - Definiraj karakteristike razredov
 - Dodaj individualne objekte in relacije med objekti
 - Iteriraj dokler ni “zadosti dobro”
 - Zloži vse to v ontologijo
- Izgradi OWL ontologijo
 - Vprašaj se ali je ontologija konsistentna
 - Vprašaj se če so razredi koherentni

Uporaba OWL

- Naseli svet (za določeno opravilo)
 - Določi individualne objekte potrebne za delo
 - Definiraj razmerja med individualnimi objekti
 - Določi omejitve individualnih objektov
- Napiši podatke v OWL sintaksi
 - Vprašaj se če so podatki konsistentni
 - Vprašaj se kakšne podatke se da izpeljati

Kreiranje ontologij

- Podatki v OWL opisujejo v glavnem **ontologije**
 - ontologija—“a branch of metaphysics concerned with the nature and relations of being” [Merriam-Webster Dictionary]
 - Ontologija določa kaj je zanimivo znotraj dane domene in kako so podatki strukturirani
 - OWL ontologija je samo zbirka informacij v glavnem podatki o **razredih** in **lastnostih**
Ontology([name] ...)
- Ontologija lahko **vkluči** (import) podatke iz drugih ontologij
 - Ontology([name] owl:imports(<name>) ...)

Literatura

- <http://www.w3.org/TR/owl2-overview/>
- Jos de Bruijn, Semantics Web Technologies, Course at Free University of Bolzano, 2008.
- Peter F. Patel-Schneider, OWL-Tutorial, Bell Labs Research, Lucent Technologies