



Od izvornih podatkov (1 ★) do povezanega znanja (5 ★)

<https://github.com/FRI-MDP/semanticne-delavnice/>

dr. **Dejan Lavbič** in dr. **Slavko Žitnik** (UL FRI)



Agenda

1. Uvod: Zakaj semantično opisovanje podatkov?
2. Od izvornih podatkov (1 ★) do odprtih podatkov (3 ★)
3. Raven 4 ★: semantični opis podatkov z RDF
4. Formalizacija podatkovnega modela z ontologijami
5. Raven 5 ★: povezani podatki in zunanji viri



1. Uvod: Zakaj semantično opisovanje podatkov?

Namen delavnice

Problem pred tehnologijo

- Zakaj objava podatkov ni dovolj
- Kaj dejansko rešuje semantika
- Pomen za javni sektor



Motivacijski problem

Podatki obstajajo – odgovori na vprašanje pa ne

- Razpršeni podatki
- Različni viri
- Različni pomeni



Od podatkov do znanja

Struktura ≠ pomen

- **Podatek** (npr. 298000)
- **Informacija** (Število prebivalcev Ljubljane.)
- **Metapodatek** (Kaj ta številka pomeni, za katero leto, vir, metodologija.)
- **Znanje** (Sklepanje, povezovanje, odgovori na kompleksna vprašanja, npr. Ljubljana je največje mesto v Sloveniji z 298.000 prebivalci (podatki iz leta 2023, SURS).)



Zakaj je to pomembno za javni sektor?

Interoperabilnost in ponovna uporaba

- Veliko podatkov
- Veliko podvajanja
- Malo povezovanja



Kaj semantično opisovanje ni?

Pogoste zmote:

- ❌ *“To je samo za raziskovalce.”*
- ❌ *“To pomeni, da moramo najprej narediti ontologijo.”*
- ❌ *“To je preveč kompleksno za praktično uporabo.”*



Ključni pojmi - poenostavljeno

Enostavne analogije

- URI
- RDF
- Ontologija
- SPARQL
- Linked Data





DEMO: Primer iz prakse

Semantika že danes

- Spletni članki
- Iskanje
- Povezovanje vsebin



2. Od izvornih podatkov (1 ★) do odprtih podatkov (3 ★)



Model 5 ★ odprtosti podatkov

★	Raven	Značilnosti podatkov
1 ★	Objavljeni podatki	PDF, HTML, poročila
2 ★	Strukturirani podatki	Excel, lastni formati
3 ★	Odprti podatki	CSV, strojno berljivi formati
4 ★	Semantično opisani podatki	RDF, URI-ji, ontologije
5 ★	Povezani podatki	Zunanji viri, Linked Data



DEMO: Izvorni podatki (1 ★ in 2 ★): ko podatki obstajajo, a niso uporabni

- podatki CRP iz [NIO](#),
 - Nabor podatkov NIO CRP.xls
 - Sifranti IO CRP.xls
- pomanjkljivosti
 - razpršeni po datotekah in zavihkih,
 - pomeni stolpcev ni jasen,
 - brez konteksta,
 - iste vrednosti → različen koncept



DEMO: Prehod v odprti format (3 ★)

- pretvorba v CSV (odprt in strojno berljiv format), kot prvi korak k tehnični interoperabilnosti,
- preprost 2D besedilni zapis (vrstice, stolpci, ločila),
- izboljša **dostopnost**, ne pa še **pomena**.





DEMO: Podatki SURS (3 ★)

- Podatki o prebivalcih po občinah ([SURS](#), 2025),
- prenos v **odprtem formatu (CSV/TSV)**,
- vrednosti in kratice stolpcev (`data.tsv`),
- pomen stolpec zapisan **ločeno** (`info.txt`),
- **odprti podatki brez vgrajene semantike.**



Omejitve ravni 3 ★

- Odprti in strojno berljivi podatki (CSV)
- Brez skupnih identifikatorjev
- Različna imena in strukture
- Pomen ni formalno opredeljen
- **Struktura ≠ semantika**





3. Raven 4 ★: semantični opis podatkov z RDF

Zakaj CSV ni več dovolj?

- Odprti in strojno berljivi podatki (3 ★)
- Brez pomena za računalnik
- Enaki nizi ≠ ista entiteta
- Manjkajo enolični identifikatorji
- Potrebujemo **URI-je**, **RDF** in **formalni pomen**





RDF: opis podatkov v obliki trojčkov

- **Osebek** – kaj opisujemo
- **Povedek** – katero lastnost opisujemo
- **Predmet** – vrednost ali povezava
- Eksplicitno izražen pomen
- Podatki kot **graf**, ne tabela

URI: enolična identifikacija pojmov

Od ID-ja v tabeli do globalnega pomena:

- enolični in stabilni identifikatorji
- globalno naslovljivi (splet)
- osnova povezovanja podatkov



Zapis RDF v obliki Turtle (TTL)

- RDF ima več zapisov (serializacij)
- **Turtle (TTL)** je najbolj berljiv
- Pogosto uporabljen v praksi



TSV → RDF

Vsaka vrstica → več trojčkov

- Vrstica = entiteta
- Stolpci = lastnosti
- RDF preseže tabelo

data.tsv

ob_id	ob_ime	tot_p
001	Ajdovščina	19895
213	Ankaran	3446
...



DEMO: RDF trojčki – polni zapis

- Subjekt Povedek Predmet .

`https://onto.mdp.gov.si/obcina/ajdovscina`

`http://www.w3.org/1999/02/22-rdf-syntax-ns#type`

`https://onto.mdp.gov.si/shema/Obcina .`

`https://onto.mdp.gov.si/obcina/ajdovscina`

`https://onto.mdp.gov.si/shema/idObcinaSurs "001" .`

`https://onto.mdp.gov.si/obcina/ajdovscina`

`https://onto.mdp.gov.si/shema/naziv "Ajdovščina" .`

`https://onto.mdp.gov.si/obcina/ajdovscina`

`https://onto.mdp.gov.si/shema/steviloPrebivalcev 19895`

.





DEMO: Predpone (prefix)

- Skrajšanje URI-jev (`rdf:type` → `a`)
- Bolj berljiv zapis
- Standardna praksa

```
@prefix obcina: <https://onto.mdp.gov.si/obcina/> .
```

```
@prefix shema: <https://onto.mdp.gov.si/shema/> .
```

```
obcina:ajdovscina a shema:Obcina .
```

```
obcina:ajdovscina shema:idObcinaSurs "001" .
```

```
obcina:ajdovscina shema:naziv "Ajdovščina" .
```

```
obcina:ajdovscina shema:steviloPrebivalcev 19895 .
```



DEMO: Krajši zapis (TTL sintaksa)

- podpičje ; za isti subjekt
- bolj jedrnato

```
@prefix obcina: <https://onto.mdp.gov.si/obcina/> .
```

```
@prefix shema: <https://onto.mdp.gov.si/shema/> .
```

```
obcina:ajdovscina a shema:Obcina ;
```

```
    shema:idObcinaSurs "001" ;
```

```
    shema:naziv "Ajdovščina" ;
```

```
    shema:steviloPrebivalcev 19895 .
```

Od tabele do grafa

- **CSV** = 2D tabela
- **RDF** = graf
- Priprava na povezovanje



Kako določimo URI-je?

- Stabilni in trajni identifikatorji
- Berljivi in enolični
- Brez posebnih znakov
- Niso vezani izključno na interne ID-je



Vzorec URI-jev na delavnici

- Entitete: /obcina/{naziv}
- Pojmi (shema): /shema/{pojem}
- Jasna ločitev pomena





Kaj dobimo z ravnjo 4 ★?

- **Enolična identifikacija** entitet
- **Povezljivost** med različnimi viri
- **Grafovska struktura** podatkov
- Osnova za **poizvedovanje** in **sklepanje**
- Priprava na **ontologije**



4. Formalizacija podatkovnega modela z ontologijami

Zakaj RDF sam po sebi ni dovolj?

- RDF zagotovi zapis in povezljivost
- Pomen pojmov ni formalno opredeljen
- Pravila so implicitna
- Računalnik jih ne razume
- Potrebujemo **ontologijo**





Kaj je ontologija (v kontekstu delavnice)?

- Formalni opis pomena
- Razredi, lastnosti in relacije
- Pravila in omejitve
- Model področja (ne podatki)



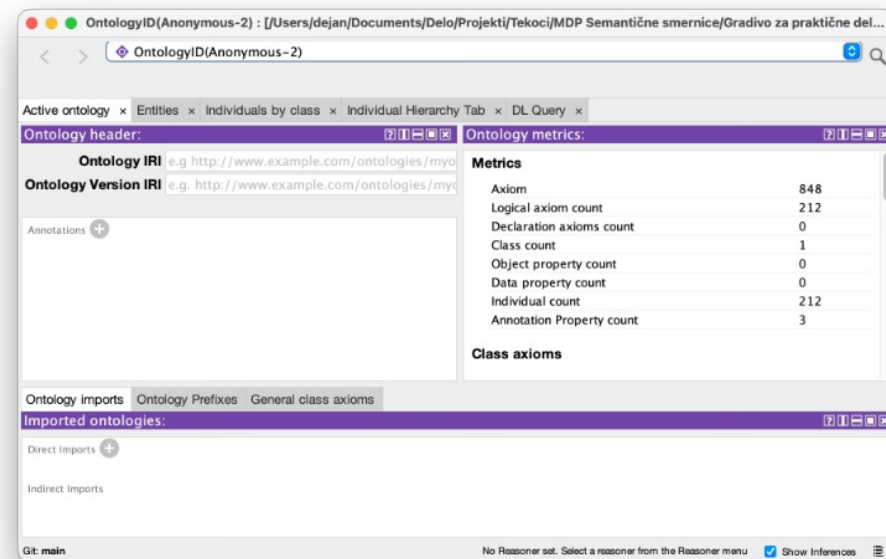
Ločevanje modela in podatkov: TBox in ABox

- **TBox** – model, struktura in pravila
- **ABox** – konkretni podatki in primerki
- Isti model, različni podatki
- Lažje vzdrževanje in nadgradnje



DEMO: Uvoz podatkov TTL v Protégé

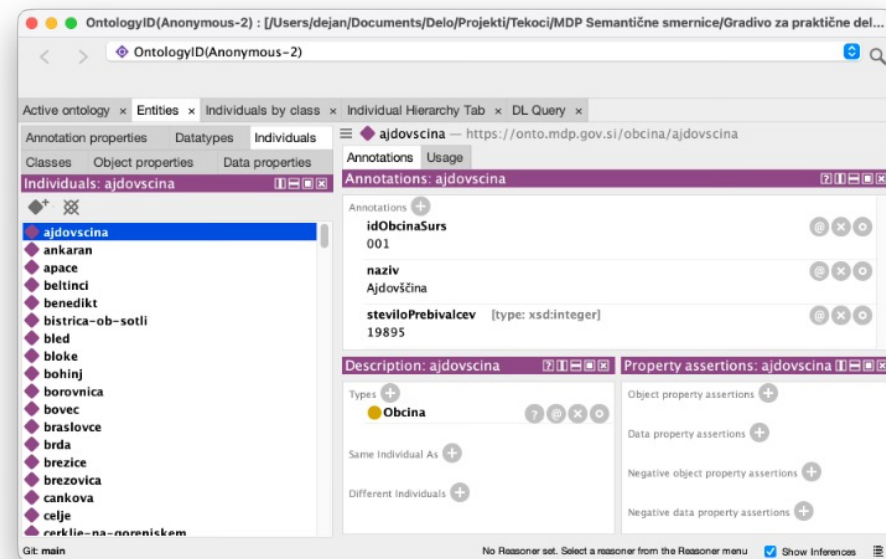
- Začnemo z **realnimi podatki SURS** (SURS_obcine.ttl)
 - 848 trditev (axioms)
 - 212 primerkov (občin)
 - 3 lastnosti (še brez formalnega pomena)
- RDF/TTL brez formalnega modela
- Fokus na podatkih (ABox)





DEMO: Pregled podatkov: Individuals

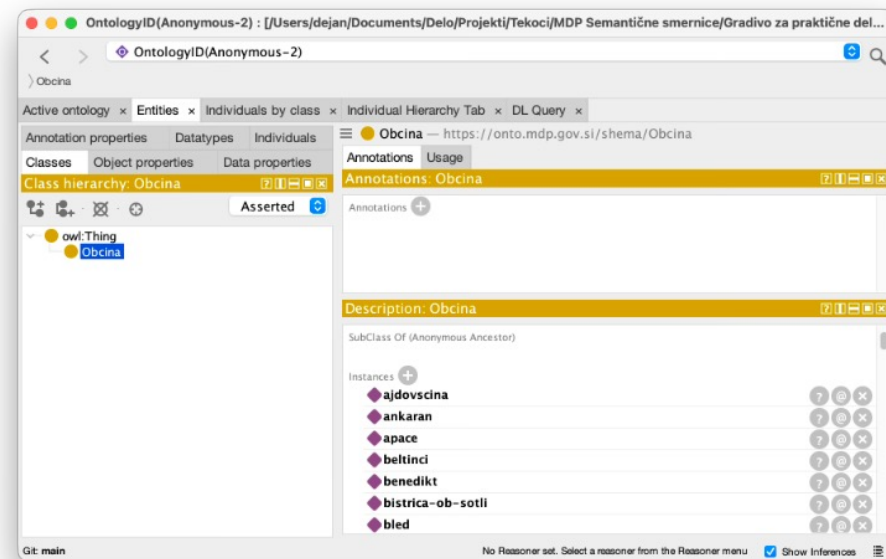
- Vsi primerki v zavihku **Individuals**
- Brez formalnega opisa modela





DEMO: Pregled podatkov: Classes

- En sam razred: `Obcina`
- Brez formalnega opisa modela



Prvi korak formalizacije: opis razreda

- Iz podatkov izluščimo pojme
- V središču: **Občina**
- Prehod iz implicitnega v formalni model





DEMO: Razred Obcina

Implicitna opredelitev

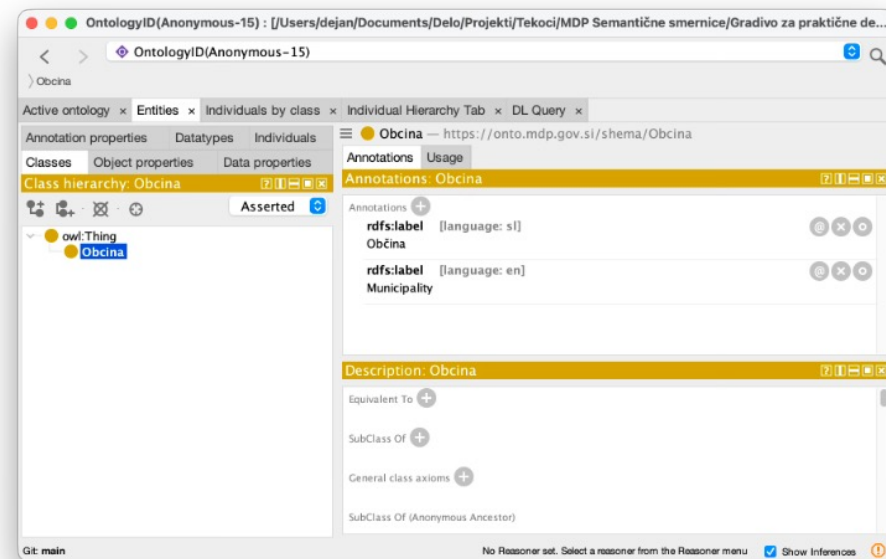
```
obcina:ajdovscina a  
schema:Obcina .
```

Formalna opredelitev

```
schema:Obcina a  
owl:Class ;
```

```
    rdfs:label  
    "Občina"@sl ;
```

```
    rdfs:label  
    "Municipality"@en .
```



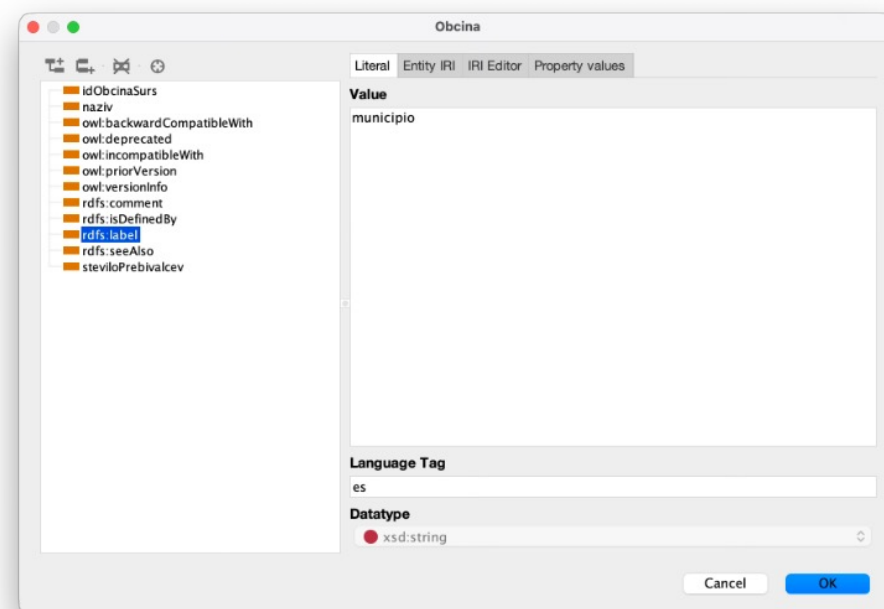


DEMO: Dodajanje oznak (večjezičnost)

- `rdfs:label`
- jezikovne oznake
 - ročno (`@sl`, `@en`)
 - GUI (`@es`)

```
shema:Obcina rdf:type  
owl:Class ;
```

```
    rdfs:label  
    "Municipality"@en ,  
    "Občina"@sl ,  
    "Municipio"@es .
```





DEMO: Kaj Protégé doda samodejno (1/2)?

- lastnosti še niso formalno opredeljene,
- zato se interpretirajo kot **oznake**.

```
schema:idObcinaSurs rdf:type owl:AnnotationProperty .
```

```
schema:naziv rdf:type owl:AnnotationProperty .
```

```
schema:steviloPrebivalcev rdf:type  
owl:AnnotationProperty .
```



DEMO: Kaj Protégé doda samodejno (2/2)?

Primerki kot `owl:NamedIndividual`

```
obcina:ajdovscina rdf:type owl:NamedIndividual ,  
                    shema:Obcina .
```

Osnovni imenski prostor in ontologija

```
@base <https://onto.mdp.gov.si/shema/> .
```

```
<https://onto.mdp.gov.si/shema/> a owl:Ontology .
```




DEMO: Podatkovne lastnosti

- Formalizacija obstoječih lastnosti
- Prehod iz opisov v **podatkovne lastnosti**
- Lastnosti pripadajo razredu `Obcina`





DEMO: Dodajanje podatkovnih lastnosti v Protégé (1/2)

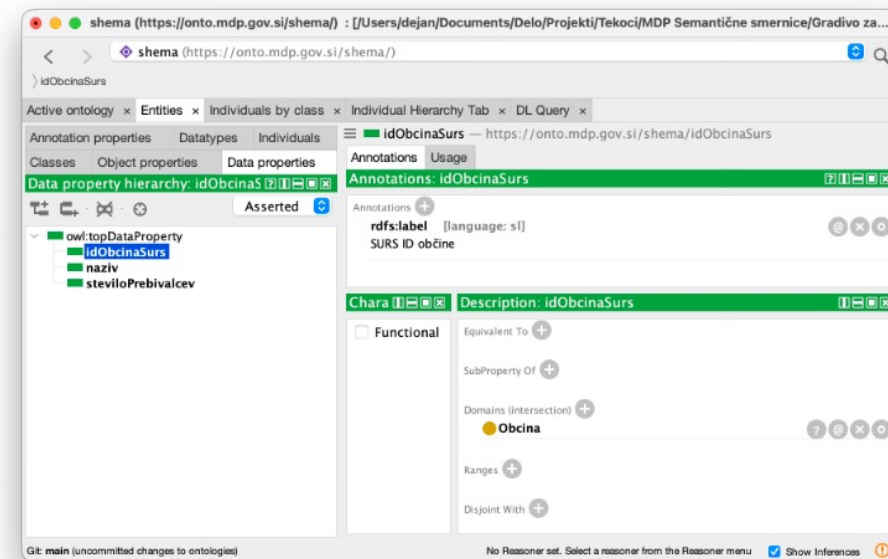
- Entities → Data properties
- Lastnosti imajo **domeno** in **obseg** (tip podatka)

```
schema:idObcinaSurs a  
owl:DatatypeProperty ;
```

```
    rdfs:domain  
    schema:Obcina ;
```

```
    rdfs:range  
    xsd:string ;
```

```
    rdfs:label "SURS ID  
občine"@sl .
```





DEMO: Dodajanje podatkovnih lastnosti v Protégé (2/2)

```
schema:naziv a owl:DatatypeProperty ;
```

```
    rdfs:domain schema:Obcina ;
```

```
    rdfs:range xsd:string ;
```

```
    rdfs:label "naziv"@sl .
```

```
schema:steviloPrebivalcev a owl:DatatypeProperty ;
```

```
    rdfs:domain schema:Obcina ;
```

```
    rdfs:range xsd:integer ;
```

```
    rdfs:label "število prebivalcev"@sl .
```


Zakaj potrebujemo objektne lastnosti?

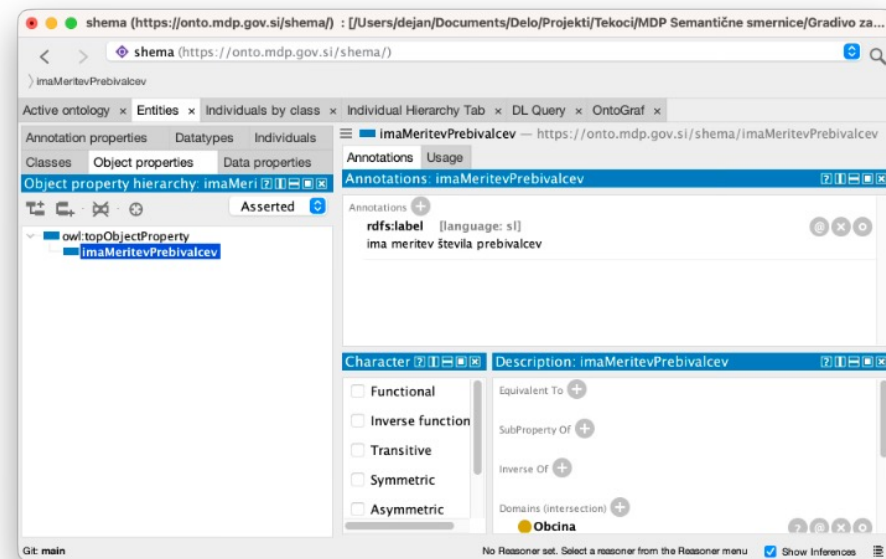
- Število prebivalcev ni trajna lastnost občine
- Gre za **opazovanje v času**
- Ena občina → več meritev
- Graf omogoča naravno razširitev





DEMO: Model z objektno lastnostjo

- Nov razred:
`MeritevPrebivalcev`
- Lastnosti: leto, vrednost
- Objektna lastnost:
`imaMeritevPrebivalcev`





DEMO: Preoblikovanje podatkov z novo strukturo

- Vrednosti → povezave na meritve
- Več let brez spremembe razreda `Obcina`
- Podatki postanejo časovno razširljivi

```
obcina:ajdovscina a schema:Obcina ;
```

```
    schema:imaMeritevPrebivalcev obcina:ajdovscina-  
stevilo-prebivalcev-2024, obcina:ajdovscina-stevilo-  
prebivalcev-2025 .
```

```
obcina:ajdovscina-stevilo-prebivalcev-2024 a  
schema:MeritevPrebivalcev ;
```

```
    schema:leto "2024"^^xsd:gYear ;
```

```
    schema:vrednost 19891 .
```





5. Raven 5 ★: povezani podatki in zunanji viri

Kaj pomeni raven 5 ★ v praksi?

- RDF + URI + ontologija (4 ★)
- Podatki so še vedno izolirani
- Manjka povezava z drugimi viri
- 5 ★ = **povezani podatki**





DEMO: Zunanji viri za povezovanje (5 ★) (1/2)

- **Podatkovni modeli (TBox)**

- Centralni besednjak – Platforma za semantično interoperabilnost

- **Splošne ontologije (TBox)**

- Schema.org – <https://schema.org/>
- FOAF – <http://xmlns.com/foaf/spec/>
- Dublin Core – <http://purl.org/dc/elements/1.1/>
- SKOS – <http://www.w3.org/2004/02/skos/core#>
- DCAT – <http://www.w3.org/ns/dcat#>



DEMO: Zunanji viri za povezovanje (5 ★) (2/2)

- **Splošni podatkovni viri (ABox)**

- DBpedia – <https://wiki.dbpedia.org/>
- Wikidata – <https://www.wikidata.org/>
- LinkedGeoData – <http://linkedgeodata.org/>

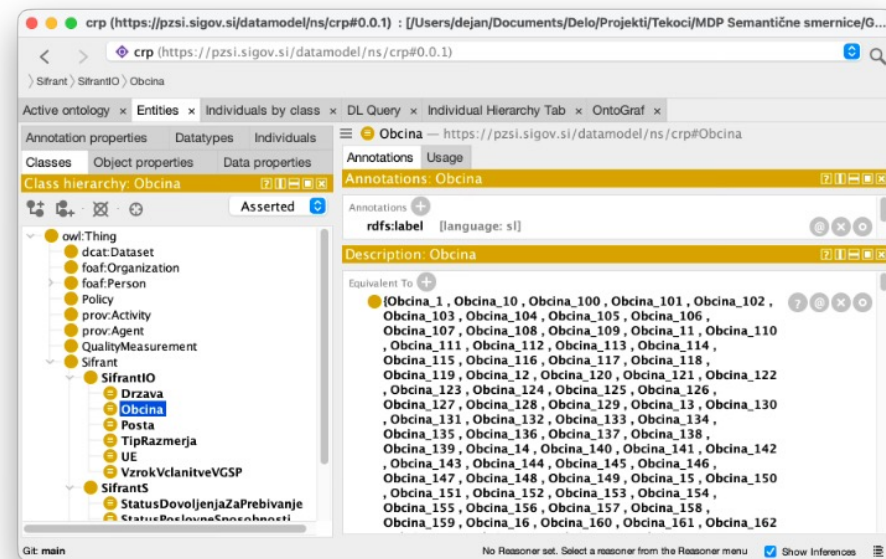
- **Domenski besednjaki**

- EUROVOC – <https://eurovoc.europa.eu/>
- GEMET – <https://www.eionet.europa.eu/gemet/>
- INSPIRE – <https://inspire.ec.europa.eu/glossary>
- GeoNames – <https://www.geonames.org/>



DEMO: Povezovanje s hierarhijo pojmov (rdfs : subClassOf) (1/2)

- Razredi so organizirani hierarhično
- Podrazredi dedujejo pomen nadrazredov
- Omogoča splošnejše poizvedbe
- Temelj za sklepanje
- `cnbs:Sifrant`
 - `:SifrantIO`
 - `:Obcina`





DEMO: Povezovanje s hierarhijo pojmov (rdfs:subClassOf) (2/2)

```
@prefix : <https://pzsi.sigov.si/datamodel/ns/crp#> .
```

```
@prefix cnbs: <http://onto.mju.gov.si/centralni-besednjak-sifranti#> .
```

```
cnbs:Sifrant rdf:type owl:Class .
```

```
:SifrantIO rdf:type owl:Class ;
```

```
    rdfs:subClassOf cnbs:Sifrant .
```

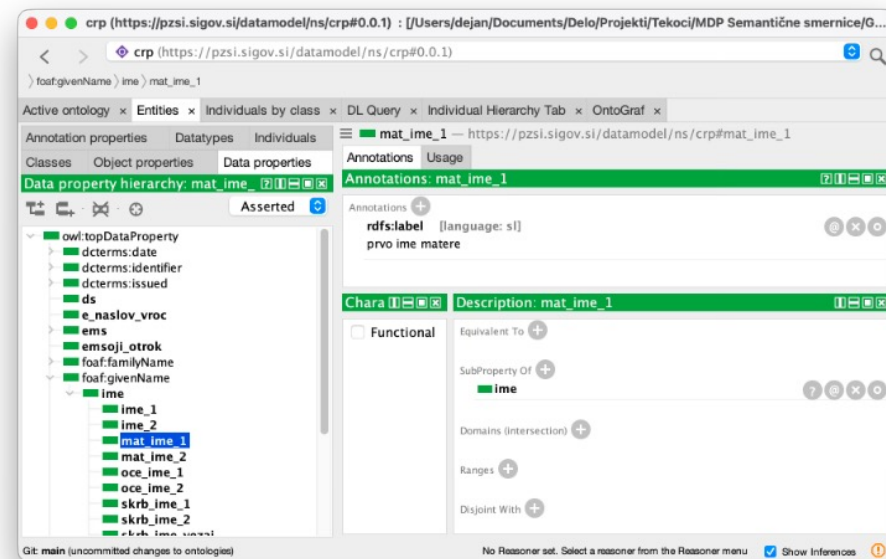
```
:Obcina rdf:type owl:Class ;
```

```
    rdfs:subClassOf :SifrantIO .
```




DEMO: Hierarhija lastnosti (`rdfs:subPropertyOf`) (1/2)

- Podobno kot pri razredih, tudi lastnosti tvorijo hierarhijo
- Splošna lastnost → bolj specifične podlastnosti
- Omogoča splošnejše poizvedbe in sklepanje
- `foaf:givenName`
 - `:ime`
 - `:mat_ime_1`





DEMO: Hierarhija lastnosti (`rdfs:subPropertyOf`) (2/2)

```
@prefix : <https://pzsi.sigov.si/datamodel/ns/crp#> .
```

```
@prefix cnb: <http://onto.mju.gov.si/centralni-  
besednjak-core#> .
```

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
```

```
:ime rdf:type owl:DatatypeProperty ;
```

```
    rdfs:subPropertyOf foaf:givenName ;
```

```
    rdfs:domain cnb:FizicnaOseba .
```

```
:mat_ime_1 rdf:type owl:DatatypeProperty ;
```

```
    rdfs:subPropertyOf :ime ;
```

```
    rdfs:label "prvo ime matere"@sl .
```

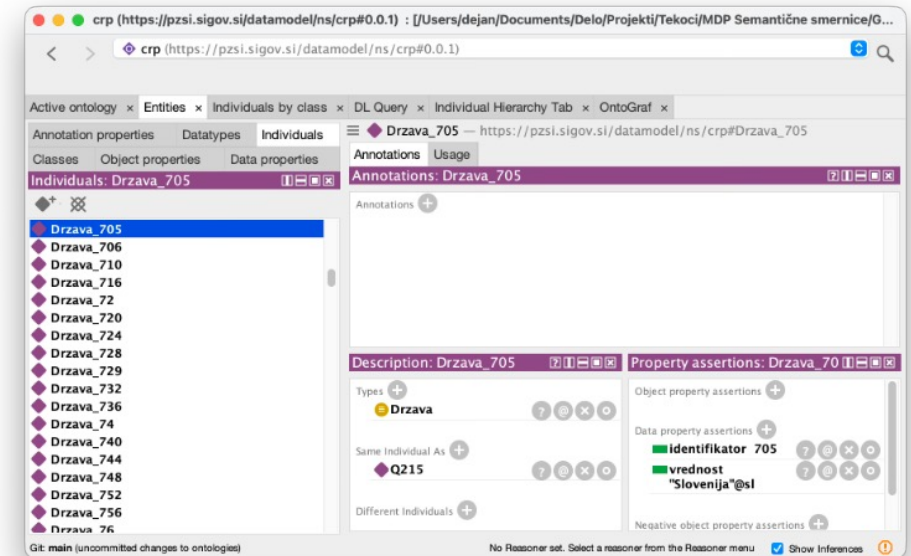



DEMO: Povezovanje istih primerkov (owl:sameAs) (1/2)

- Včasih vemo, da dva URI-ja opisujeta **isto entiteto**
- Entiteta je ista, razlikuje se le **vir podatkov**
- owl:sameAs pove: *to je popolnoma isto*

Primer:

- CRP: :Obcina_1
- Wikidata: wd:Q331701 (Ajdovščina)





DEMO: Povezovanje istih primerkov (owl:sameAs) (2/2)

```
@prefix : <https://pzsi.sigov.si/datamodel/ns/crp#> .
```

```
@prefix wd: <https://www.wikidata.org/entity/> .
```

```
:Obcina_1 rdf:type owl:NamedIndividual ,
```

```
        :Obcina ;
```

```
        :identifikator 1 ;
```

```
        :vrednost "Ajdovščina"@sl ;
```

```
owl:sameAs wd:Q331701 .
```

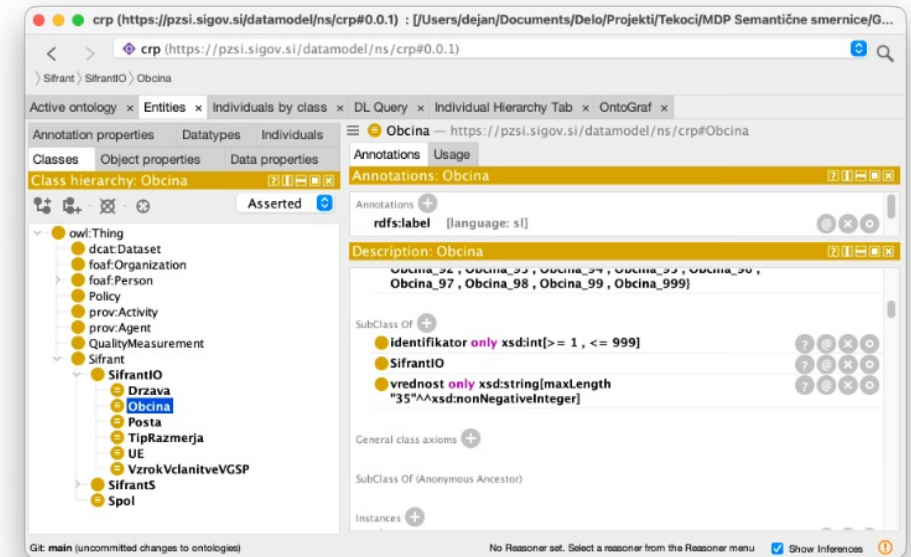


DEMO: Povezovanje pojmov (owl:equivalentClass) (1/2)

- Povezujemo **pojme / razrede**, ne primerkov
- Različni slovarji → isti pomen
- owl:equivalentClass pomeni: *pojma sta semantično enakovredna*

Primer:

- :Obcina (CRP)
- Municipality v drugem slovarju / ontologiji
- opredelitev vseh vrednosti





DEMO: Povezovanje pojmov (`owl:equivalentClass`) (2/2)

```
:Obcina rdf:type owl:Class ;  
  
    owl:equivalentClass [ rdf:type owl:Class ;  
                            owl:oneOf ( :Obcina_1  
                                           :Obcina_10  
                                           :Obcina_100  
                                           ... ) ] ;  
  
rdfs:subClassOf :SifrantIO ,  
    [ rdf:type owl:Restriction ; ... ] ,  
    [ rdf:type owl:Restriction ; ... ] .
```




Povezovanje CRP in SURS (cilj)

- 2 neodvisna vira:
 - **CRP** – uradni šifranti, stabilni identifikatorji
 - **SURS** – statistični kazalniki (npr. prebivalstvo)
- Opisujeta **iste občine**
- Cilj:
 - ne združujemo podatkov v eno datoteko
 - omogočimo **skupno uporabo** v poizvedbah in sklepanju
- Integracija v ločeni datoteki:
 - Integracija_CRP_SURS.ttl



DEMO: Povezovanje primerkov (CRP SURS)

- Najbolj neposreden pristop: **owl:sameAs**

- Primer:

- CRP: `crp:Obcina_1`
- SURS: `sursABox:ajdovscina`

- Pomen:

- gre za **isto občino**

@prefix crp:

<<https://pzsi.sigov.si/datamodel/ns/crp#>> .

@prefix sursABox: <<https://onto.mdp.gov.si/obcina/>> .

`crp:Obcina_1 owl:sameAs sursABox:ajdovscina .`



DEMO: Povezovanje pojmov in širši ekosistem

- Povezovanje **pojmov (TBox)**:
 - **previdno**: `rdfs:subClassOf`
 - **strogo**: `owl:equivalentClass`
- Primer (previdnejša varianta):

```
@prefix crp:
```

```
<https://pzsi.sigov.si/datamodel/ns/crp#> .
```

```
@prefix surstBox: <https://onto.mdp.gov.si/shema/> .
```

```
surstBox:Obcina rdfs:subClassOf crp:Obcina .
```




DEMO: Razširitev povezovanja – več občin, en ekosistem

- Ajdovščina že povezana z **Wikidata**
- Dodamo še druge občine: Celje, Ljubljana, Maribor
- CRP ↔ SURS ↔ Wikidata
- Podatkov **ne podvajamo**, ampak jih **logično povežemo**

```
sursTBox:Obcina rdfs:subClassOf crp:Obcina .
```

```
crp:Obcina_11 owl:sameAs sursABox:celje .
```

```
crp:Obcina_11 owl:sameAs wd:Q3441823 .
```