

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**V A R A Ž D I N**

**Iva Udovčić**  
**David Kajzogaj**

**Učenje ontologija**

**SEMINARSKI RAD**

**Varaždin, 2025.**

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**V A R A Ž D I N**

**Iva Udovčić, 0016148057**

**David Kajzogaj, 0016146827**

**Studij: Organizacija poslovnih sustava**

**Učenje ontologija**

**SEMINARSKI RAD**

**Mentor/Mentorica:**

Prof. dr. sc. Sandra Lovrenčić

**Varaždin, siječanj 2025.**

# Sadržaj

Sadržaj .....	iii
1. Uvod .....	1
2. Definicija ontologije .....	2
2.1. Ontologije na semantičkom webu .....	4
3. Učenje ontologija .....	6
3.1. Slojevi učenja ontologije .....	10
3.2. Arhitektura učenja ontologije .....	10
3.3. Zadatci učenja ontologije .....	12
3.4. Otvoreni problemi u učenju ontologija .....	12
4. Primjena ontologije .....	14
5. Primjeri alata i platformi .....	16
6. Primjer .....	18
7. Zaključak .....	34
Popis literature .....	35
Popis slika .....	37
Prilozi .....	39

# 1. Uvod

U brzom razvoju informacijskih tehnologija, organizacija i upravljanje podacima postaju ključni izazovi. S rastom količine informacija na internetu i u korporativnim sustavima, pojavljuje se potreba za tehnologijama koje mogu omogućiti učinkovito strukturiranje i razumijevanje podataka. Ontologije, kao formalne reprezentacije znanja pružaju okvir za semantičku obradu informacija, čineći podatke razumljivima ne samo ljudima, već i računalima.

Tema ovog rada je učenje ontologija, a odnosi se na proces automatskog ili poluautomatskog otkrivanja i izrade ontološkog znanja iz raznovrsnih izvora podataka. Ovaj pristup predstavlja inovaciju u tradicionalnom procesu razvoja ontologija, koji često zahtijeva intenzivno uključivanje stručnjaka i ručno modeliranje. Kroz primjenu tehnika strojnog učenja, obrade prirodnog jezika i analize podataka, učenje ontologija omogućuje bržu izradu i održavanje složenih ontoloških sustava.

Razvoj semantičkog weba temeljen na standardima poput RDF-a i OWL-a, stvorio je temelje za uvođenje ontologija u svakodnevne primjene, od pretraživanja internetskih resursa do optimizacije poslovnih procesa i integracije podataka u biomedicini. U tom kontekstu, učenje ontologija postaje ključna tehnologija za rješavanje problema poput heterogenosti podataka, skalabilnosti i dinamičnosti domena.

Svrha ovog rada je pružiti pregled osnovnih koncepata učenja ontologija, istražiti metode i alate koji se koriste za njihovu primjenu te analizirati ključne izazove i mogućnosti koje se otvaraju razvojem ovog područja. Kroz razmatranje različitih primjera i aplikacija, rad će ilustrirati kako učenje ontologija može unaprijediti procese organizacije podataka, omogućiti strojno razumijevanje i podržati donošenje odluka u kompleksnim sustavima.

## 2. Definicija ontologije

Ontologije predstavljaju formalnu konceptualizaciju određenog područja interesa koja je zajednička grupi ljudi [1]. Pojam ontologija izvorno potječe iz filozofije, gdje označava znanost o postojanju, odnosno o onome što jest (grč. *ovtos*). [3]

Temelj je semantičkog weba. Kako bi ontologija bila korisna u dijeljenju informacija i upravljanju znanjem potrebni su učinkoviti i djelotvorni pristupi razvoju samih ontologija. Učenje ontologija, koje nastoji automatski ili poluautomatski otkriti ontološka znanja iz različitih oblika podataka, može prevladati usko grlo stjecanja ontologija u procesu njihova razvoja. Ontologija služi kao snažan semantički temelj za strojno razumijevanje digitalnog sadržaja. Široko je prisutna u informacijskim sustavima, omogućujući dodavanje meta-podataka dokumentima, unapređenje učinkovitosti pretraživanja i zaključivanja te osiguravanje interoperabilnosti podataka među različitim aplikacijama. Učenje ontologija podrazumijeva automatsko otkrivanje i kreiranje ontološkog znanja uz pomoć tehnika strojnog učenja. Za razliku od ručne izrade ontologija, učenje ontologija ne samo da omogućuje brže i opsežnije otkrivanje znanja, već također smanjuje pristranosti i nedosljednosti koje mogu unijeti ljudi. Ontologije su dizajnirane za pružanje znanja o određenim domenama na način koji je razumljiv i ljudima i računalima. One posebno definiraju pojmove unutar domene te odnose između tih pojmova, a mogu uključivati i jasnu definiciju svojstava, funkcija, ograničenja i aksioma. [2]

Ontologije u informatici obično se smatraju formalnim prikazima znanja, često specifičnim za određenu domenu. Ipak, ne postoji opći dogovor o tome koje točno zahtjeve formalna reprezentacija treba zadovoljiti da bi bila smatrana ontologijom. Ovisno o pristupu, ontologije mogu biti jednostavni rječnici, taksonomije, tezauri ili složeni top-level formalni sustavi. Ontologije igraju ključnu ulogu u integraciji podataka i znanja jer, stvarajući zajednički okvir, olakšavaju upite i zaključivanje među različitim izvorima podataka. [4]

S obzirom na brz rast podataka na internetu i unutar korporativnih mreža, potreba za strukturiranim pristupom podacima postaje sve važnija. Osim što omogućuju integraciju podataka, zaključivanje i postavljanje upita, ontologije služe i kao alat za dokumentiranje strukture specifičnih domena, što pomaže u stvaranju zajedničkog razumijevanja njihovih pojmova. [4]

Međutim, izrada ontologija predstavlja vrlo složen i skup proces koji u velikoj mjeri ovisi o dostupnosti stručnjaka. Za razvoj formalne ontologije za neku domenu, potrebno je prikupiti i formalizirati specifično znanje na način koji omogućuje automatizirane zaključke. To je moguće samo uz suradnju stručnjaka iz domene i inženjera ontologija koji su upoznati s

teorijama i praksama reprezentacije znanja. Nakon što je ontologija stvorena, njezino održavanje postaje nužno zbog promjena u znanju i potrebama primjene. [4]

Reprezentacija ontologija ključno je pitanje u procesu njihovog razvoja. Osim što omogućuje razumijevanje ontologija od strane računala i ljudi, jezik za njihovu reprezentaciju mora osigurati prikladnost u prikazu i visoku učinkovitost kod izvođenja zaključaka. Posljednjih godina postignut je značajan napredak u standardizaciji jezika za ontološku reprezentaciju, poput RDF-a, RDFS-a i OWL-a. Ovi jezici temelje se na okvirnom modelu reprezentacije znanja, pri čemu neki, poput OWL-a, koriste deskriptivne logike kako bi povećali izražajnu snagu i omogućili naprednije zaključivanje. [2]

Stvaranje ontologija obuhvaća kreiranje njihovog sadržaja, uključujući pojmove i njihove odnose. Budući da to zahtijeva duboko poznavanje određene domene, modeliranje ontologija tradicionalno obavljaju inženjeri znanja i stručnjaci za to područje. Međutim, zbog brzih promjena u domenama znanja i sve veće geografske raspršenosti stručnjaka, pristup specijaliziranim znanjima postaje sve izazovniji. Osim toga, stručna znanja često mogu biti nepotpuna, subjektivna ili zastarjela. Kako bi se prilagodili ovim izazovima, stručnjaci se sve češće okreću alternativnim izvorima podataka, poput rječnika, dokumenata s interneta i struktura baza podataka, kako bi popunili sadržaj ontologija. U tom smislu, učenje ontologija pruža značajne prednosti u procesu stjecanja ontološkog znanja. [2]

Evaluacija ontologija predstavlja još jedno ključno područje, osobito kako postaju sve češće korištene. Procjena kvalitete ontologija može poboljšati njihovu interoperabilnost među različitim sustavima te potaknuti širu primjenu. Ontologije se mogu ocjenjivati prema raznim kriterijima, uključujući njihov sadržaj, tehničke karakteristike, metode razvoja i njihovu praktičnu primjenu. Korištenje objektivnih mjera poput potpunosti, konzistentnosti i točnosti ključni je dio ovog procesa. Učenje ontologija prepoznato je kao alternativna metoda za procjenu sadržaja ontologija. [2]

Održavanje ontologija odnosi se na njihovu organizaciju, pretraživanje i ažuriranje. Kako broj ontologija i njihova složenost rastu, postaje sve važnije osigurati njihovu redovnu evaluaciju i prilagodbu promjenjivom okruženju. Na primjer, sustavi poput SWOOGLE, koji indeksiraju i pretražuju semantički web, upravljaju s milijunima dokumenata u RDF i OWL formatima. Ručno održavanje takve količine ontologija predstavlja iznimno zahtjevan zadatak, pa su potrebna automatizirana rješenja za njihovu integraciju i mapiranje. Učenje ontologija nudi obećavajući pristup za rješavanje ovog problema. [2]

Iz gore navedenog je vidljivo da brojna pitanja povezana s razvojem ontologija mogu imati koristi od učenja ontologija. Ova metoda omogućuje prikupljanje ontološkog znanja uz minimalnu ljudsku intervenciju. Umjesto isključivog oslanjanja na stručnjake za određene

domene, učenje ontologija koristi različite izvore podataka unutar domene i primjenjuje tehnike strojnog učenja te obrade prirodnog jezika za poluautomatsko ili potpuno automatsko otkrivanje znanja. Učenje ontologija ne samo da povećava učinkovitost razvoja ontologija, već omogućuje i otkrivanje novih spoznaja analizom postojećih baza podataka. [2]

## 2.1. Ontologije na semantičkom webu

Trenutni izvori informacija na internetu prvenstveno su dizajnirani za ljude i omogućuju pristup putem poveznica. S druge strane, semantički web ne fokusira se na strukturu tih poveznica, već na povezanost između elemenata i njihovih svojstava. Semantika proučava značenje jezičnih znakova, dok informacijska znanost istražuje strukture podataka bez interesa za njihovo značenje. U posljednje vrijeme, umjetna inteligencija, koja je postala ključna grana informacijskih znanosti, igra važnu ulogu u semantičkoj obradi podataka, a tehnike poput obrade prirodnog jezika, strojnog učenja i strojne translacije postale su bitne. [5]

Semantički web temelji se na konceptu upravljanja podacima i programima koji nisu izravno povezani s podacima. World Wide Web Consortium (W3C) odgovorno je za razvoj i standardizaciju web tehnologija. Krajem prošlog tisućljeća započeo je rad na semantičkom webu, a Tim Berners-Lee, osnivač interneta, 1996. godine ukazao je na problem pronalaženja podataka koji nisu povezani semantičkim oznakama. 1998. godine Berners-Lee je predstavio semantički web kao plan za razvoj povezanih aplikacija koje omogućuju stvaranje logičke mreže podataka koju računala mogu obraditi. Glavni cilj semantičkog weba je povezivanje ključnih područja ljudskog znanja i omogućavanje njihove obrade od strane računala koristeći jasno definirane, strojno razumljive termine. Razvoj inteligentnih agenata sposobnih za samostalno obavljanje složenih zadataka kroz komunikaciju između strojeva bio je ključna svrha semantičkog weba. [5]

Na semantičkom webu podaci će biti predstavljeni na način koji je strojno čitljiv, čime će računala moći interpretirati njihovo značenje. Semantički web omogućit će pretraživanje temeljen na značenju, a ne samo na ključnim riječima, što je karakteristično za današnji web. To zahtijeva semantičku i sintaktičku interoperabilnost. [5]

Izgradnja semantičkog weba temelji se na uspostavi nove razine međuoperabilnosti, koja je zasnovana na otvorenim standardima. Ovi standardi ne samo da definiraju sintaktički oblik dokumenata, već omogućuju i osnovu za njihov semantički sadržaj. Semantički web koristi osnovne tehnologije postojećeg weba za stvaranje novog internetskog okvira, koji bi trebao služiti kao univerzalni medij za razmjenu podataka, informacija i znanja. Ovaj koncept temelji se na slojevitoj arhitekturi koja se nadograđuje na postojeće mrežne protokole. Ključne komponente uključuju XML za sintaksu podataka, RDF za modeliranje podataka, RDF

Schema za definiranje rječnika, ontologije (OWL), te logičke i sigurnosne mehanizme (Logic, Proof, Trust). [5]

RDF (Resource Description Framework) nastao je kao standard za opisivanje metapodataka i odnosa između web resursa, i to u apstraktnom formatu. RDF koristi model "subjekt-predikat-objekt" za izražavanje semantičkih veza među resursima, gdje se svi dijelovi identificiraju putem URI-ja (Uniform Resource Identifier). Ovaj model omogućava stvaranje triplestore baza podataka koje pohranjuju i omogućuju pretragu podataka u obliku RDF trojki. [5]

Za razvoj semantičkog weba nužna su dodatna rješenja koja proširuju RDF, poput ontologija i OWL (Web Ontology Language), koje omogućuju bogatije semantičko opisivanje domena i njihovih odnosa. Ontologije predstavljaju sustave pojmova i njihovih veza, dok OWL definira standarde za strojno čitljivo predstavljanje ontologija. Također, u semantičkom webu koriste se jezici poput OIL (Ontology Inference Layer) i DAML (DARPA Agent Markup Language) koji proširuju RDF Schema i omogućuju preciznije semantičko razumijevanje podataka. [5]

Semantički web također koristi upitne jezike poput SPARQL, te mehanizme logičkog zaključivanja i sigurnosti, kako bi omogućio naprednu obradu i interpretaciju podataka na internetu. [5]



### 3. Učenje ontologija

Prošlo je više od desetljeća kako je uveden pojam učenje ontologija, koji označava novo istraživačko područje posvećeno automatskom generiranju ontologija. Održane su i radionice na tu temu koje su okupile stručnjake iz različitih znanstvenih zajednica te su predstavljeni radovi na toj radionici koji pokazuju tri pristupa: metode temeljene na pravilima prijenosa (ripple down rules), tehnike klasteriranja značenja riječi i metode za ekstrakciju informacija. Postoji nekoliko kategorija učenja ontologija [4]:

1. **Učenje ontologija iz teksta** prvenstveno se bavi automatskim ili poluautomatskim stvaranjem jednostavnih taksonomija pomoću tehnika rudarenja teksta i ekstrakcije informacija. Brojni pristupi korišteni u ovom procesu, poput leksikosintaktičkih uzoraka za identificiranje odnosa hiponimije ili klasifikacije imenovanih entiteta, temelje se na prethodnim istraživanjima iz područja računalne lingvistike, s glavnim ciljem olakšavanja prikupljanja leksičkih podataka iz velikih tekstualnih zbirki. Pojedini pristupi u učenju ontologija ne usredotočuju se na stvaranje shematskih struktura, već na razradu same razine podataka. Takve metode, poznate kao populacija ontologija, izvlače činjenice iz teksta. Primjer takvog pristupa je projekt Never-Ending Language Learning (NELL), koji kontinuirano pretražuje web kako bi obogatio svoju bazu znanja, a svoju učinkovitost unapređuje kroz vrijeme, među ostalim i putem povratnih informacija od korisnika.
2. **Rudarenje povezanih podataka** uključuje identificiranje ključnih obrazaca u RDF grafovima. Ovo istraživačko područje razvilo se zbog toga što izdavači povezanih podataka često ne izrađuju unaprijed precizne sheme, već se fokusiraju na objavu samih podataka. Prepoznavanje strukture unutar objavljenih RDF grafova može pojednostaviti kasnije definiranje shema i omogućiti otkrivanje zanimljivih odnosa među elementima grafa. Ovi ciljevi postižu se primjenom metoda poput statističke indukcije shema ili statističkog relacijskog učenja, koje analiziraju česte uzorke i povezanosti u velikim skupovima podataka. Uz to, tehnike klasteriranja koriste se za grupiranje srodnih resursa, čime se poboljšava organizacija osnovnih podataka.
3. **Učenje koncepata u opisnoj logici i OWL-u** usmjereno je na stvaranje aksioma shema, poput definicija klasa, iz postojećih ontologija i instanciranih podataka. Većina tehnika u ovom području temelji se na induktivnom logičkom programiranju. Iako algoritmi poput DL-FOIL i OCEL predstavljaju univerzalne pristupe nadziranom učenju u opisnoj logici, postoje i specifične prilagodbe za učenje ontologija, s naglaskom na poboljšanje učinkovitosti i jednostavnosti primjene. Blisko povezano s učenjem

konceptata u opisnoj logici je onto-relacijsko učenje, koje spaja metode za generiranje OWL aksioma s tehnikama učenja pravila.

4. **Crowdsourcing** ontologija predstavlja zanimljivu alternativu automatskim metodama jer spaja brzinu tehnologije s ljudskom preciznošću. Kada je zadatak dovoljno jednostavan, ključni je element pružiti odgovarajuće motivacije za angažman ljudi. Primjeri crowdsourcinga u kontekstu učenja ontologija uključuju izradu taksonomija putem Amazon Mechanical Turk-a i igre dizajnirane za obogaćivanje ontologija.

Drugi pristupi obuhvaćaju, primjerice, transfer učenje i ponovnu upotrebu ontologija, koji se fokusiraju na prilagodbu postojećih ontologija novim domenama, koristeći djelomično postojeće shematske strukture. Uz kombiniranje pravila i ontologija, istražuju se i izravni načini predstavljanja nesigurnosti, poput Markovljevih logičkih mreža [4].

Različiti pristupi, i veće i manje razlike među njima, otežavaju postizanje jedinstvene definicije ili podjelu na konkretne pod zadatke, koji ne bi trebali biti ni preširoki ni usmjereni samo na jedan specifičan pristup. Ipak, ta raznolikost čini zajednicu koja se bavi učenjem ontologija izuzetno bogatom i inspirativnom. Zabilježen je značajan napredak u svim tim područjima, a uočava se trend prema integriranim i hibridnim metodama. Izazovi u učenju ontologija [4]:

1. **Heterogenost.** Podaci na internetu mogu se znatno razlikovati, primjerice, u pogledu formata, jezika, domena i kvalitete. Pristupi učenju iz tih raznovrsnih izvora podataka mogu učinkovito iskoristiti ovu raznolikost, čime se poboljšava točnost i širina stvorenih ontologija. Međutim, integracija različitih metoda i homogenizacija podataka još uvijek nisu dovoljno istraženi u zajednici za učenje ontologija, što predstavlja izazov za njihovu praktičnu primjenu.
2. **Nesigurnost.** Podaci niske kvalitete ili nestrukturirani podaci, koji su teško interpretabilni računalnim metodama, kao i nesavršene tehnike za učenje ontologija, mogu dovesti do netočnih rezultata. Zbog toga su mnoge metode učenja ontologija dizajnirane tako da svaka odluka bude povezana s vrijednošću sigurnosti, koja ukazuje na povjerenje u točnost tih rezultata. Informacije o nesigurnosti, kao i podaci o porijeklu poput vremenskih oznaka ili bilješki o autoru, ključni su za ručno ili automatsko ispravljanje pogrešaka u naučenim ontologijama.
3. **Razmišljanje.** Ontologije se često razvijaju ili uče za primjene koje se oslanjaju na logičko zaključivanje. Ako takve primjene zahtijevaju logičku dosljednost ontologije, pristupi učenju ontologija trebaju biti sposobni generirati dosljedne (i koherentne) ontologije. Stoga, osim metoda za učenje konceptata u opisnim logikama, i drugi pristupi učenju ontologija koriste logičko zaključivanje. Eksperimenti su pokazali da bliska

povezanost između ispravljanja pogrešaka u ontologijama, kao što su dijagnoza i popravak nesuglasnosti, i procesa učenja ontologija može donijeti koristi.

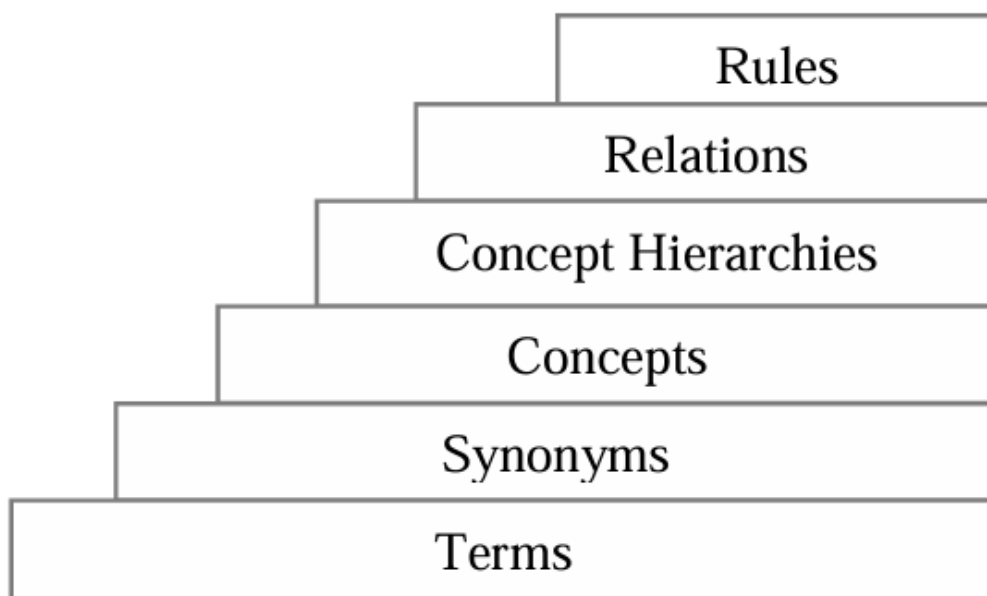
4. **Skalabilnost.** Izdvajanje znanja iz sve većih količina podataka na internetu – nestrukturiranih tekstualnih podataka s jedne strane, te strukturiranih podataka poput baza podataka, povezanih podataka ili ontologija s druge strane – zahtijeva skalabilne i učinkovite pristupe. Posebno kada se radi o učenju iz distribuiranih, slabije povezanih podataka i integraciji znanja iz više izvora, metode učenja ontologija suočavaju se s velikim izazovima. Kako bi se ti izazovi prevladali, razvijaju se različite strategije, poput distribuiranog računalstva za horizontalno skaliranje učenja ontologija, inkrementalnih pristupa za ponovnu upotrebu postojećeg znanja te uzorkovanja i modularizacije za povećanje učinkovitosti algoritama za učenje ontologija.
5. **Kvaliteta.** Kvaliteta automatski generiranih ontologija može se mjeriti na isti način kao i kod bilo koje druge ontologije, bilo da je stvorena ručno ili naučena. Ipak, evaluacija ontologija nije jednostavan. Formalna ispravnost, potpunost i dosljednost samo su neki od mogućih kriterija za procjenu kvalitete ontologije, a izbor tih kriterija te željeni standard kvalitete ovise o specifičnom kontekstu primjene ontologije. Idealno bi bilo da svaki korak u procesu učenja ontologije, uključujući izbor podataka, preprocesiranje i ekstrakciju odnosa, bude prilagođen specifičnoj domeni ili kontekstu u kojem će se ontologija koristiti. Metode za učenje ontologija i primjena obrazaca za dizajn ontologija mogu dodatno unaprijediti rezultate.
6. **Interaktivnost.** U praksi, kvaliteta naučene ontologije često ovisi o stupnju automatizacije. Manje ljudske intervencije u poluautomatskom procesu generiranja ontologije obično rezultira nižom kvalitetom. Ontologija koja nije dovoljno kvalitetna za predviđenu primjenu, kao što je slučaj s potpuno automatski generiranim ontologijama, zahtijevat će značajnu naknadnu obradu. Iako revizija naučenih ontologija obično nije dio samog procesa učenja, automatske metode generiranja ontologija mogu olakšati postprocesiranje pružanjem detaljnih podataka o porijeklu. Te informacije obično uključuju vrijednosti povjerenja i relevantnosti za pojedine aksiome, kao i vremenske oznake i ključne informacije o primijenjenim metodama učenja. Unatoč tome, količina naknadne obrade može predstavljati značajan izazov za inženjere znanja, zbog čega su potrebne inovativne metode za prevladavanje poznatog problema akvizicije znanja. Općenito, preporučuje se integracija metoda za učenje i reviziju ontologija u uobičajene okvire za inženjering ontologija, kako bi se smanjio ljudski angažman u cijelom procesu. Crowdsourcing i igre s ciljem mogu pomoći u smanjenju troškova revizije naučenih ontologija uključivanjem ne-stručnjaka, ali izazov je kako pretvoriti njihove interakcije u konkretne odluke za modeliranje ontologija. Sustavna ispitivanja stručnjaka, poznata kao relacijsko istraživanje, pokazala su se učinkovitim načinom postavljanja pravih

pitanja ljudima, dok istovremeno smanjuju broj odluka koje je potrebno donijeti. Konačno, eksperimenti su pokazali da integracija obrazaca dizajniranja ontologija, koji sažimaju iskustvo ljudskih inženjera ontologija, može značajno unaprijediti proces učenja ontologija.

### 3.1. Slojevi učenja ontologije

Učenje ontologije može se predstaviti kroz skup slojeva koji se obrađuju korak po korak. Cilj je identificirati ključne pojmove prepoznati njihove sinonime, svrstati ih u kategorije, definirati hijerarhijske strukture, uspostaviti odnose među konceptima i utvrditi pravila koja ograničavaju i reguliraju njihovo ponašanje. Slojevi učenja ontologije su sljedeći: [6]

- **Pojmovi:** Jezični prikazi specifičnih koncepata unutar određenog područja.
- **Sinonimi:** Prepoznavanje različitih jezičnih varijacija istog pojma, kako unutar istog jezika, tako i između različitih jezika.
- **Definicije koncepata:** Obuhvaćaju formalna i neformalna značenja.
- **Strukture hijerarhije:** Uspostavljaju odnose između svojstava koncepata i njihovih veza s drugim konceptima.
- **Relacije/Odnosi:** Definiraju tipove poveznica između koncepata u domeni
- **Pravila:** Omogućuju donošenje zaključaka i proširivanje znanja unutar ontološkog sustava.

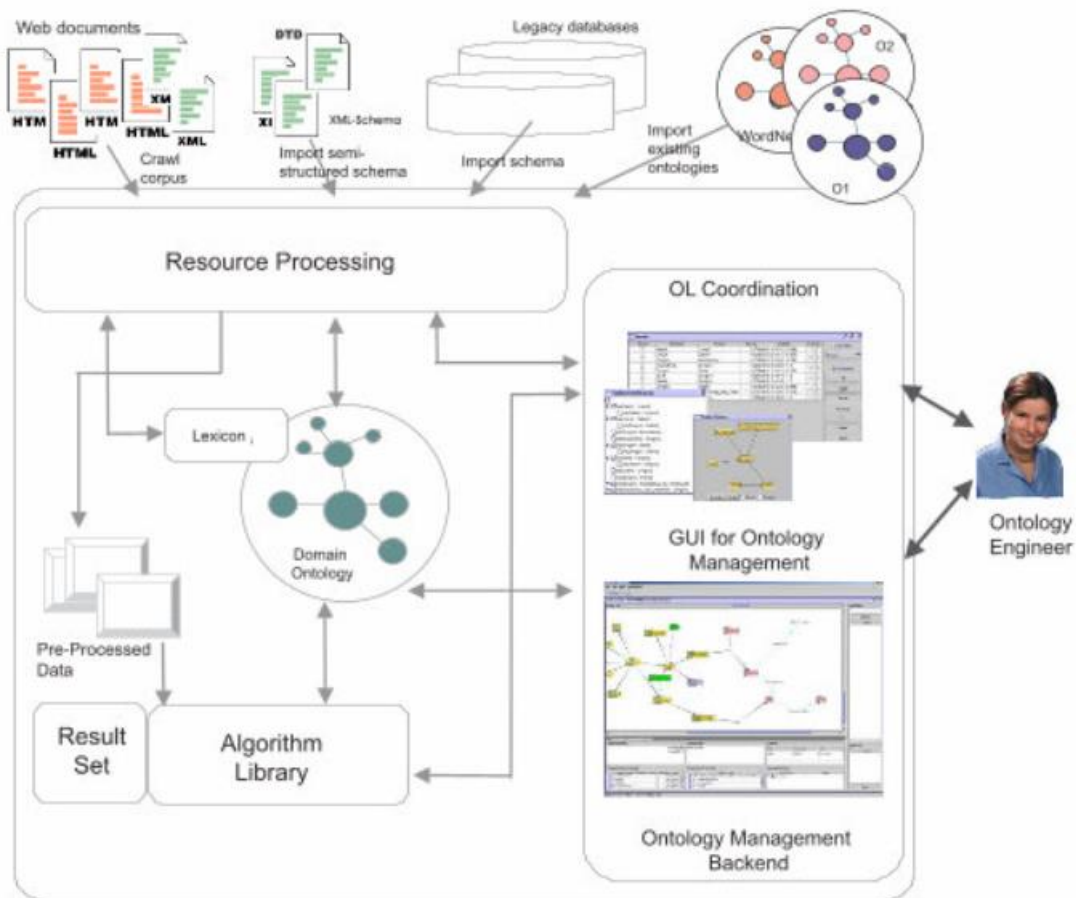


Slika 1. Slojevi učenja ontologije (l. (n.d.). , str. 1220 ).

### 3.2. Arhitektura učenja ontologije

Arhitektura učenja ontologija se sastoji od 4 glavne komponente. Te komponente su komponenta za upravljanje ontologijama, komponenta knjižnice algoritama, koordinacijska komponenta te komponenta za obradu resursa. U nastavku su objašnjene: [6]

- **Komponenta za upravljanje ontološkim sustavima:** Ovaj modul omogućuje stručnjacima za ontologiju ručno upravljanje i prilagodbu ontoloških struktura. Omogućuje dodavanje postojećih ontologija, njihovo pregledavanje, validaciju te prilagodbu i stalno unapređenje.
- **Modul za obradu podataka:** Zadužen je za identifikaciju, analizu, uvoz i pretvorbu ključnih ulaznih podataka. Ključna podkomponenta uključuje sustav za obradu prirodnog jezika. Glavni cilj ovog modula je osigurati pripremljene podatke kao ulaz za algoritamski sustav.
- **Algoritamski modul:** Središnji je element odgovoran za primjenu različitih algoritama u svrhu izgradnje i održavanja ontološkog modela. Uključuje razne metode prilagođene analizi i obradi podataka.
- **Modul za koordinaciju:** Ovaj dio omogućuje stručnjacima interakciju s komponentama za učenje ontologija, upravljanje resursima te algoritamske obrade. Korisnička sučelja omogućuju odabir podataka, primjenu metoda obrade i aktivaciju mehanizama za ekstrakciju. Dobiveni rezultati prikazuju se u različitim oblicima za analizu i pregled struktura ontologije.



Slika 2. Arhitektura učenja ontologija (I. (n.d.). , str. 1220).

### 3.3. Zadatci učenja ontologije

Učenje ontologije obuhvaća automatsko ili poluautomatsko izgrađivanje ontoloških struktura iz različitih izvora podataka. Ključni zadaci uključuju: [7]

#### 1. Izrada ontologije

- Kreiranje nove ontologije od temelja od strane tima stručnjaka.
- Korištenje strojnog učenja (ML) kako bi se pomoglo u procesu dizajniranja ontologije, uključujući prijedloge za odgovarajuće odnose među pojmovima i provjeru dosljednosti.

#### 2. Izvlačenje shema ontologije

- Proces ekstrakcije shema ontologije iz različitih izvora podataka.
- Uz pomoć strojnog učenja, ulazni podaci i meta-ontologije koriste se za izradu potpunih ontologija uz asistenciju ljudskih stručnjaka.

#### 3. Ekstrakcija instanci ontologije (populacija ontologija)

- Izdvajanje konkretnih instanci iz polustrukturiranih ili nestrukturiranih podataka kako bi se napunile postojeće ontološke sheme sa specifičnim pojedincima.

#### 4. Integracija i navigacija ontologijama

- Rekonstrukcija postojećih baza podataka i omogućavanje navigacije kroz njih, kao i prevođenje baza znanja u različite ontološke jezike, poput prebacivanja s FOL-a (logika prvog reda) u OWL 2 DL.

#### 5. Ažuriranje ontologije

- Proširenje i prilagodba već postojećih ontologija kako bi odgovarale promjenama u domeni, uključujući prilagodbu dijelova ontologije koji se mogu modificirati.

#### 6. Obogaćivanje ontologije

- Automatizirani postupak za dorađivanje određenih dijelova postojećih ontologija, bez promjena osnovnih koncepata i odnosa, već se preciznije definiraju.
- Ova aktivnost se razlikuje od ažuriranja jer ne zahtijeva promjenu osnovnih elemenata ontologije.

### 3.4. Otvoreni problemi u učenju ontologija

Iako je istraživanje u ovom području postiglo značajan napredak zahvaljujući doprinosima iz različitih disciplina, kao što su strojno učenje i obrada prirodnog jezika, još uvijek nije razvijen dovoljno sofisticiran skup alata za generiranje kvalitetnog naučenog znanja. Jedan od ključnih izazova je učenje iz ne tekstualnih izvora. Tekstualni dokumenti dominiraju u učenju ontologija, budući da su najčešći izvor podataka. Međutim, s razvojem multimedijalnih

sadržaja, potrebno je razmotriti i učenje ontologija iz slika, zvučnih zapisa i video materijala. Problemi u učenju ontologija su sljedeći: [8]

- Ontologije za ljude i ontologije za strojeve: Ontološki jezici nalaze se na spektru između prirodnog jezika koji je razumljiv ljudima i formalnih jezika razumljivih strojevima. Za približavanje ovih dvaju krajeva, važno je omogućiti ljudima da i dalje koriste prirodni jezik, dok se automatski mapiraju riječi i izrazi u ontološke koncepte.
- Učenje specifičnih relacija: Većina istraživanja do sada fokusirala se na opće odnose između konceptata, dok su specifične vrste odnosa, poput relacije dio-cjelina, često zanemarene. Za mnoge specifične domene, ove relacije imaju značajnu ulogu.
- Učenje više stupanjskih relacija: Binarne relacije, koje povezuju dva koncepta, proučavane su u većini istraživanja, no relacije višeg stupnja, kao što je **relacija** povjerenja, ostale su nedovoljno istražene.
- Učenje definicija: Različiti pristupi učenju definicija, od ekstrakcije novih definicija do generiranja alternativnih ili kombiniranih definicija, još uvijek nisu formalno evaluirani. U procesu se često koriste relacije poput "definira" ili "odnosi se na".
- Filtriranje pojmova: Tijekom ekstrakcije konceptata iz teksta za specifičnu domenu može doći do velikog broja pogrešaka, s obzirom na prisutnost nerelevantnih pojmova pa je stoga filtriranje unaprijed ključni korak u procesu.
- Kreiranje ontologija visokog stupnja: Proces izrade ontologija visokog stupnja na temelju podataka dobivenih metodama učenja ostaje problematičan. Ručno ili automatsko povezivanje lingvističkih entiteta s ontološkim konceptima, kao i integracija različitih ontologija, još uvijek predstavlja veliki izazov.
- Evaluacija ontologija: Većina ontologija razvijenih tehnikama učenja procjenjuje se ručno, što je zahtjevan zadatak zbog nedostatka stručnjaka za specifične domene. Potrebna je izrada korpusa s mjerama evaluacije za različite domene, što bi omogućilo automatiziranu validaciju ontologija.
- Inkrementalno učenje ontologija: Kako se znanje u određenoj domeni razvija, ontologije je potrebno povremeno ažurirati s novim podacima. Korištenje postojećih ontologija i njihovo inkrementalno ažuriranje mnogo je učinkovitije nego izrada novih ontologija od početka.



## 4. Primjena ontologije

Primjena ontologija obuhvaća širok raspon područja, uključujući e-učenje, biomedicinu, informacijske sustave i semantički web. Njihova uporaba pomaže u organizaciji podataka, olakšava integraciju informacija i omogućuje ponovno korištenje postojećeg znanja. Gdje se sve primjenjuju ontologije slijedi u nastavku: [9]

- **Industrija nafte i plina**

Industrija nafte i plina generira ogromne količine podataka svaki dan. To uključuje podatke o naftnim bušotinama, seizmičke podatke, podatke o bušenju, transportu, korisnicima i marketingu. S obzirom na značaj ove industrije na globalnoj razini, podaci i njihove semantičke informacije igraju ključnu ulogu u izvlačenju korisnih informacija. Primjer Fluor Corporation razvila je projekt koji koristi RDF/OWL formate za omogućavanje obrade podataka iz industrije nafte i plina. Integracija semantičkog weba s Internetom stvari može dodatno unaprijediti poslovne procese u ovoj industriji.

- **Vojna tehnologija**

Mnoge vojne tehnologije poput bespilotnih letjelica i autonomnih borbenih robota generiraju ogromne količine podataka tijekom bitaka. Semantički web pomaže u organizaciji tih podataka i pomaže u donošenju odluka tijekom borbi. Ontologije se koriste za strukturiranje podataka s bojišta, čime se omogućava brži pristup i analiza podataka. Na primjer, upotreba RDF reprezentacija i SPARQL protokola omogućava analizu podataka za prepoznavanje prijetnji.

- **E-uprava**

Korištenje ontologija i semantičkog weba u e-upravama može značajno poboljšati učinkovitost portala. Korištenjem semantičkih tehnologija, portal može razumjeti značenje korisničkog upita i pružiti relevantne, semantički ispravne rezultate. Ontologije pomažu u održavanju usklađenosti između različitih vladinih odjela, čime se omogućava bolje upravljanje resursima i učinkovitije donošenje politika.

- **E-poslovanje i e-trgovina**

Semantički web također pomaže u donošenju poslovnih odluka i razvoju pametnih sustava u e-poslovanju i e-trgovini. Ontologija GoodRelations modelira ključne e-commerce koncepte poput proizvoda, cijena i popusta, čineći e-commerce platforme učinkovitijima. Slično, platforma **LIB2CO** koristi ontologiju za organiziranje proizvoda, omogućujući bolju pretragu i pronalaženje informacija.

- **E-zdravlje i životne znanosti**

U industriji e-zdravlja, ontologije se koriste za pohranu i integraciju medicinskih podataka, čime se omogućava bolja obrada podataka i brži pristup informacija. Ontologije kao što su European Patient Summary i SNOMED CT koriste se za strukturiranje podataka i omogućavanje automatiziranih odluka temeljenih na tim podacima.

- **Multimedija i e-kultura**

S obzirom na ogromnu količinu multimedijalnog sadržaja koji se svakodnevno objavljuje, ontologije omogućuju organizaciju i pretraživanje tih podataka. Ontologije omogućuju preciznije preporuke i poboljšavaju sustave za pretraživanje multimedijskog sadržaja. Na primjer, ontologija za slike omogućava bolje pretraživanje relevantnih slika prema korisničkim upitima.

- **Istraživačko i digitalno novinarstvo**

Semantički web i ontologije omogućuju novinarima pretraživanje i analizu podataka koji bi inače bili teško dostupni. Na primjer, projekt Panama Papers koristio je ontologije za strukturiranje podataka kako bi novinarima omogućio brži pristup važnim informacijama. Slično, novinarstvo povezano s Trump World Data koristi ontologije za strukturalizaciju podataka, olakšavajući pristup relevantnim informacijama.

## 5. Primjeri alata i platformi

Ontologije su ključne za strukturiranje i organizaciju znanja u raznim domenama. Za njihovo učinkovito učenje i primjenu razvijeni su različiti alati i platforme. Primjeri alata i platformi za učenje ontologija:

- **Protégé**

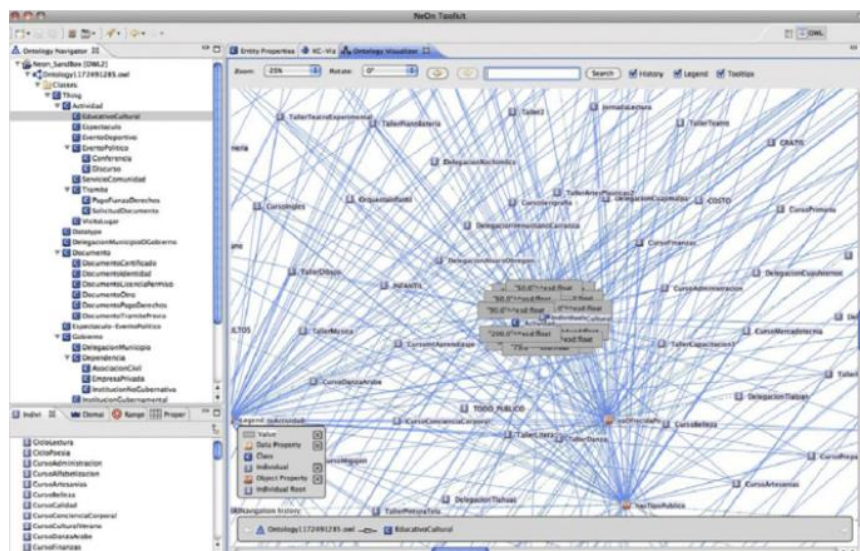
Besplatni open-source alat razvijen na Sveučilištu Stanford, omogućuje stvaranje, uređivanje i vizualizaciju ontologija. Podržava različite formate, uključujući OWL (Web Ontology Language). Neke od njegovih funkcionalnosti su: kreiranje, uređivanje i vizualizacija ontologije te sadrži bogatu biblioteku dodataka za naprednu analizu i pretragu. Najčešće se upotrebljava u medicini, biologiji i slično. U sljedećem poglavlju imamo primjer ontologije izrađene u alatu Protege. [13]

- **OntoGen**

Alat za poluautomatsko generiranje ontologija koji koristi metode strojnog učenja za pomoć korisnicima u izradi ontologija. Također olakšava kategorizaciju pojmova i njihovih odnosa te je pogodan za korisnike sa manje iskustva u modeliranju ontologija. Upotrebljava se za automatsku generaciju ontologija u istraživačkim projektima. [12]

- **NeOn Toolkits**

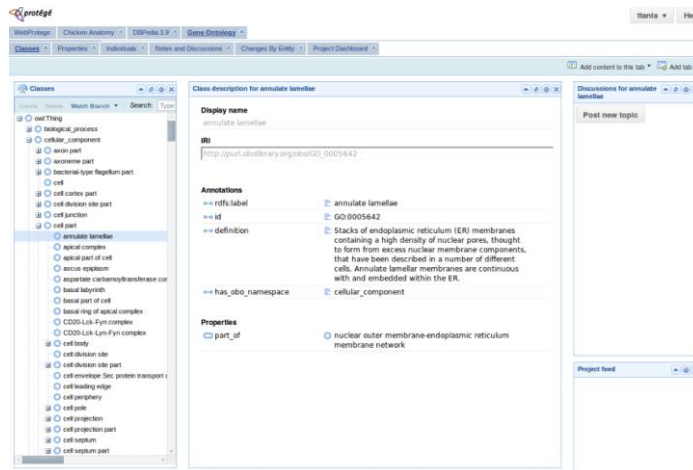
Open-source platforma za razvoj ontologija i semantičkih aplikacija, pruža integrirano okruženje za upravljanje ontologijama. Podržava modulaciju i ponovnu upotrebu postojećih ontologija te povezivanje s drugim alatima i bazama podataka. Najčešća uporaba ovog alata je pri razvoju kompleksnih ontoloških sistema. [10] [11]



Slika 3. NeOn Toolkits ([https://www.researchgate.net/figure/The-NeOn-Toolkit-illustrates-the-problems-of-occlusion-and-overcrowding-of-visual\\_fig3\\_234129385](https://www.researchgate.net/figure/The-NeOn-Toolkit-illustrates-the-problems-of-occlusion-and-overcrowding-of-visual_fig3_234129385))

- **WebProtégé**

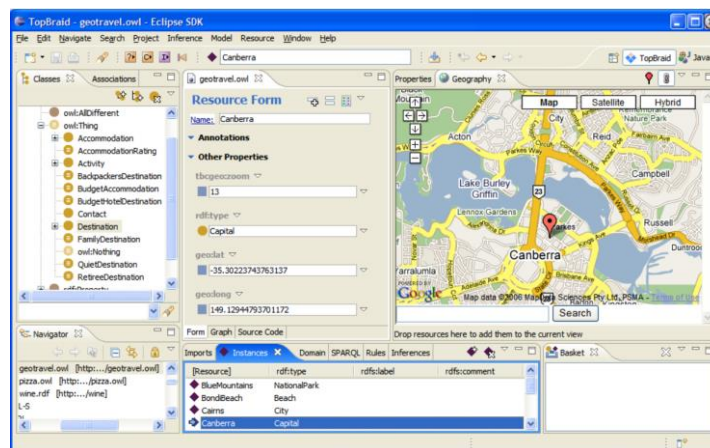
Web-bazirana verzija alata Protégé, omogućuje kolaborativno uređivanje ontologija putem web preglednika. Pogodna verzija jer nije potrebno instaliranje dodatnog softvera na računalo te omogućuje kolaborativno uređivanje ontologija putem web pretraživača. [13]



Slika 4. Web Protege (File:WP-classes.png—Protege Wiki, bez dat.)

- **TopBraid Composer**

Komercijalni alat za razvoj ontologija i semantičkih aplikacija, podržava vizualno modeliranje i integraciju podataka. Moguće je integrirati SPARQL upite te analizirati i transformirati podatke. Upotreba ove vrste alate je u svrhu poslovnih aplikacija i analitika. [14]



Slika 5. TopBraid (<https://franz.com/agraph/tbc/>)

Neki manje poznatiji alati, ali vrijedni spomena su [15]:

- RDF4J – Java framework za rad s RDF podacima.
- Apollo – Alat s fokusom na bio informatiku i genomiku.
- Swoop – OWL editor s fokusom na navigaciju i pretragu ontologija.
- Fluent Editor – Alat pomoću kojeg se definiraju ontologije na prirodnom jeziku.

## 6. Primjer

Za potrebe ovog rada napravljena je ontologija Namještaj u alatu Protege. Ontologija Namještaj ima tri klase koje su korijenske odnosno na razini 0. Jedna od klasa je klasa *MaterijalNamjestaja* koja se sastoji od podklasa: *MaterijalDrvo* i *MaterijalTkanina*. To su klase prve razine. Podklasa *MaterijalDrvo* ima svoje podklase: *MaterijalBukva*, *MaterijalHrast*, *MaterijalIverica* i *MaterijalTresnja*, a to su klase druge razine. *MaterijalTkanina* ima podklase: *MaterijalEkoKoza*, *MaterijalPlis*, *Materijalpoliester* i *MaterijalPravaKoza*, te klase su na drugoj razini. Još jedna klasa koja je na nultoj razini u ontologiji Namještaj je podklasa *MjestoMontazeNamjestaja* koja ima podklase *MontazaPod* i *MontazaZid*. Te podklase predstavljaju mjesto montaže namještaja koje može biti na zidu što vrijedi za viseće kuhinjske elemente te mjesto montaže pod koje se odnosi na namještaj poput ormara, stolica i stolova. Osim klasa *MaterijalNamjestaja* i *MjestoMontazeNamjestaja* na nultoj razini ontologije Namještaj nalazi se i klasa *Namještaj* koja ima jednu primitivnu klasu *NamjestajOdredjeni*, a ta klasa ima podklase: *NamjestajOrmar*, *NamjestajStol*, *NamjestajStolica* i *NamjestajViseciKuhinjskiElementi*. Definirane podklase u klasi *Namještaj* su: *NamjestajDrvo*, *NamjestajOdIverice* i *NamjestajOdlicnaKvaliteta*. Sve primitivne klase su međusobno razdvojene. Klasa *Namještaj* ima ograničenje: *imaMjestoMontaze some MjestoMontazeNamjestaja* što znači da ne postoji namještaj koji nema mjesto montaže. Kao nulta klasa postoji i klasa *ParticijaVrijednosti* koja ima podklasu *Kvaliteta* koja se sastoji od podklasa *Losa*, *Odlicna* i *Srednja*. Klasa *Kvaliteta* ima aksiom pokrivanja koji glasi *Odlicna or Srednja or Losa* što znači da klasu *Kvaliteta* u potpunosti pokrivaju njezine podklase odnosno ograničavaju joj doseg na točno te tri vrijednosti. Klasa *Drzava* sastoji se od instanci *BiH*, *Italija* i *Njemacka* te je prebrojiva klasa.

Definirana podklasa *NamjestajDrvo* ima ograničenje *imaMaterijal only MaterijalDrvo* što znači da se u toj podklasi nalaze samo one vrste namještaja koje su izrađene samo od drva. Uvjet *only* kod različitih vrsta namještaja (ormar, stol, stolica, viseći kuhinjski elementi) dodan je da članovi klase *NamjestajDrvo* mogu biti i individue koje nemaju vezu preko svojstva *imaMaterijal*.

Definirana podklasa *NamjestajOdIverice* ima ograničenje *imaMaterijal some MaterijalIverica* što znači da se u tu klasu svrstavaju samo one vrste namještaja koje su napravljene od iverice.

Definirana podklasa *NamjestajOdlicnaKvaliteta* ima uvjet *imaMaterijal some (MaterijalNamjestaja and (jeKvalitetan some Odlicna))* što znači da se u toj podklasi nalaze samo onaj namještaj koji je napravljen od barem jednog materijala čija je kvaliteta odlična.

Za svaki materijal postoje ograničenja vezana za državu porijekla te kvalitetu materijala. *MaterijalBukva* ima porijeklo iz Italije, a srednju kvalitetu, *MaterijalHrast* ima porijeklo iz Italije te odličnu kvalitetu. *MaterijalIverica* ima porijeklo iz Italije te srednju kvalitetu. *MaterijalTresnja* ima porijeklo iz BiH i lošu kvalitetu, *MaterijalEkoKoza* ima porijeklo iz BiH te lošu kvalitetu, *MaterijalPlis* ima porijeklo iz BiH te srednju kvalitetu. *MaterijalPoliester* ima porijeklo iz Italije te lošu kvalitetu, *MaterijalPravaKoza* ima porijeklo iz Italije i odličnu kvalitetu.

Klasa *NamjestajOrmar* ima egzistencijalna ograničenja te aksiom zatvorenja. Egzistencijalno ograničenje definira obvezne uvjete, a to su da je *NamjestajOrmar* napravljen od materijala *MaterijalBukva* i *MaterijalIverica*. Postoji i aksiom zatvorenja kod te klase, a točno definira sve moguće klase odnosno to je unija egzistencijalnih ograničenja. Instance te klase su *OrmarDvokrilni* i *OrmarTrokrilni* koji imaju podatkovno svojstvo *imaCijenu* te je određena cijena za obje instance.

Klasa *NamjestajStol* je ograničena egzistencijalnim svojstvom na materijale *MaterijalTresnja* i *MaterijalIverica* te aksiomom zatvorenja koji glasi *imaMaterijal only (MaterijalIverica or MaterijalTresnja)*. Instance su *StolBlagavaonski* i *StolRadni* koje imaju Data property assertions definiran kao *imaCijenu* te je navedena cijena za obje instance.

Klasa *NamjestajStolica* ima slijedeća ograničenja: *imaMaterijal some MaterijalHrast, imaMaterijal some MaterijalPravaKoza* te aksiom zatvorenja *imaMaterijal only (MaterijalHrast or MaterijalPravaKoza)*. Instance su *StolicaBarska* i *StolicaBlagavaonska* koje koriste podatkovno svojstvo *imaCijenu* kako bi se prikazala njihova cijena.

Klasa *NamjestajViseciKuhinjskiElementi* kao i prethodne klase ima egzistencijalna ograničenja i aksiom zatvorenja. Materijali od koji se sastoji su *MaterijalIverica*, *MaterijalPoliester* i *MaterijalTresnja*. Aksiom zatvaranja glasi: *imaMaterijal only (MaterijalIverica or MaterijalPoliester or MaterijalTresnja)*. Instance su *ViseciElementiMali* i *ViseciElementiVeliki* koji kao i prethodne instance imaju definiranu cijenu preko podatkovnog svojstva *imaCijenu*.

Objektna svojstva su, kako se moglo i zaključiti iz gornjeg teksta, *imaPorijeklo*, *imaSvojstva* (sastoji se od podsvojstava *imaMaterijal*, *imaMjestoMontaže*), *jeKvalitetan*, *jeSvojstvoOd* (sastoji se od podsvojstava *jeMaterijalOd*, *jeMjestoMontazeOd*). Svojstva *imaSvojstva* i *jeSvojstvoOd* su inverzna kao i njihova podsvojstva, a uz to su i tranzitivna. Svojstvo *jeKvalitetan* je funkcionalno svojstvo. Još jedno objektno svojstvo je svojstvo *prodajeSeU* koje za svaku individuu pojedinog namještaja definira državu u kojoj se prodaje.

Domena svojstva *jeKvalitan* je klasa *Kvaliteta*. Domena svojstva *imaMaterijal* je *Namještaj*, a doseg je *MaterijalNamjestaja*. Domena svojstva *imaMjestoMontaze* je *Namještaj*, a doseg *MjestoMontazeNamjestaja*. Domena svojstva *jeMaterijalOd* je *MaterijalNamjestaja*, a doseg je *Namještaj*. Domena svojstva *jeMjestoMontazeOd* je *MjestoMontazeNamjestaja*, a doseg je *Namještaj*.

Podatkovno svojstvo *imaCijenu* ima domen *Namještaj*, a doseg je *xsd:decimal*.

Individue klase *NamjestajOrmar* su *OrmarDječji*, *OrmarDvokrilni* i *OrmarTrokrilni*. Individue klase *NamjestajStolica* su *StolicaBarska* i *StolicaBlagavaonska*. Klasa *NamjestajStol* ima individue *StolBlagavaonski* i *StolRadni*. Posljednja klasa koja ima individue je klasa *NamjestajViseciKuhinjskiElementi*, a individue su *ViseciElementMali* i *ViseciElementVeliki*. Za svaku od individua definirana je cijena te država u kojoj se prodaje. Za ovu kratku ontologiju napravljeni su i upiti. Napravljeno je ukupno 11 upita, od toga je šest DL upita, a 5 SPARQL.

Prvi upit glasi: *Namještaj and imaCijenu some xsd:decimal [>100]*. Ovaj upit dohvaća sve iz klase *Namještaj* ako imaju cijenu veću od 100. To je jednostavni DL upit. Slijedeći upit je također jednostavni te glasi: *MaterijalNamjestaja and jeKvalitetan some Srednja*. Ovaj upit dohvaća sve materijale čija je kvaliteta definirana ograničenjem na razinu *Srednja*. Treći jednostavni DL upit glasi: *imaMaterijal some (MaterijalDrvo and jeKvalitetan some Odlicna)*. Ovaj upit dohvaća sav namještaj koji spadaju u klasu *MaterijalDrvo* te imaju definiranu kvalitetu materijala kao *Odlicna*.

U nastavku će biti prikazani složeni DL upiti. Prvi takav glasi: *(MaterijalDrvo and jeKvalitetan some Srednja) and (MaterijalDrvo and imaPorijeklo value Italija)*. Ovaj upit kao rezultat daje sve one materijale koje su u klasi *MaterijalDrvo* te imaju srednju kvalitetu i porijeklo je Italija. Bitno je naglasiti da obje strane uvjeta moraju biti zadovoljene. Dakle, materijal mora biti i iz Italije i imati srednju kvalitetu. Drugi složeni DL upit glasi: *(NamjestajOdredjeni and prodajeSeU value Njemacka) and (NamjestajOdredjeni and imaCijenu some xsd:decimal [< 200])*. Ovaj upit u rezultatu daje sav namještaj koji se prodaje u Njemačkoj te ima cijenu manju od 200. Kao i kod prethodnog upita obje strane moraju biti zadovoljene. Treći složeni DL upit glasi: *((NamjestajOdredjeni and prodajeSeU value Njemacka) or (NamjestajOdredjeni and prodajeSeU value Italija)) and (NamjestajOdredjeni and imaCijenu some xsd:decimal [< 200])*. Ovaj upit u rezultatu vraća sve one individue koje se prodaju u Italiji ili Njemačkoj, cijena im je niža od 200.

Jednostavni SPARQL upiti su navedeni u nastavku. Prvi jednostavni upit je: *SELECT ?individua ?cijena WHERE {?individua rdf:type namjestaj:NamjestajOrmar; namjestaj:imaCijenu ?cijena}*, ovaj upit kao rezultat vraća individue klase *NamjestajOrmar* te njihovu cijenu. Drugi jednostavni upit je: *SELECT ?podsvojstvo ?domena ?doseg WHERE { ?podsvojstvo rdfs:subPropertyOf namjestaj:jeSvojstvoOd; rdfs:domain ?domena; rdfs:range ?doseg}*. Ovaj upit dohvaća podsvojstva, domen i doseg objektnog svojstva *jeSvojstvoOd*. Treći jednostavni SPARQL upit glasi: *SELECT ?potklasa WHERE {?potklasa rdfs:subClassOf namjestaj:MaterijalTkanina}*. Ovaj upit prikazuje sve podklase klase *MaterijalTkanina*.

Složeni SPARQL upiti navedeni su u nastavku. Prvi složeni upit glasi:

```

SELECT ?individua ?cijena ?prodaja
WHERE {
    ?individua rdf:type namjestaj:NamjestajOrmar;
        namjestaj:imaCijenu ?cijena.
    OPTIONAL { ?individua namjestaj:prodajeSeU ?prodaja. }
    FILTER (?cijena > 100)
}.

```

Ovaj upit dohvaća individuu klase *NamjestajOrmar*, cijenu te državu u kojoj se prodaje. Mjesto prodaje je stavljeno pod uvjet *OPTIONAL* što znači da će se prikazati ukoliko postoji, no ako ne postoji ostat će prazno polje. Za cijenu je korišten filter što znači da će se prikazati samo one individue koje imaju cijenu veću od 100. Drugi složeni upit glasi:

```

SELECT ?tipNamjestaja (SUM(?cijena) AS ?ukupnaCijena) (COUNT(?individua) AS ?brojIndividua)
WHERE {
    { ?individua rdf:type namjestaj:NamjestajStolica;
        namjestaj:imaCijenu ?cijena.
        BIND("Stolica" AS ?tipNamjestaja)
    }
    UNION
    { ?individua rdf:type namjestaj:NamjestajStol;
        namjestaj:imaCijenu ?cijena.
        BIND("Stol" AS ?tipNamjestaja)
    }
}

```

```

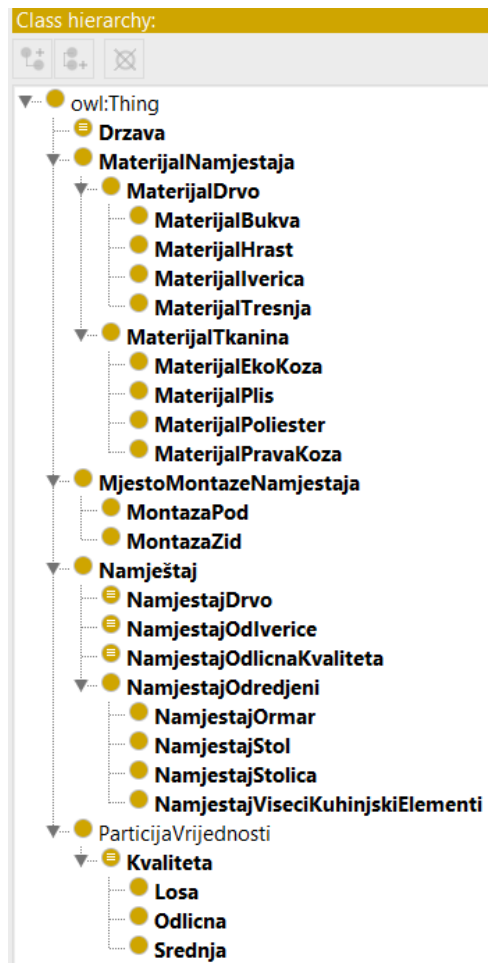
GROUP BY ?tipNamjestaja
ORDER BY ?tipNamjestaja

```

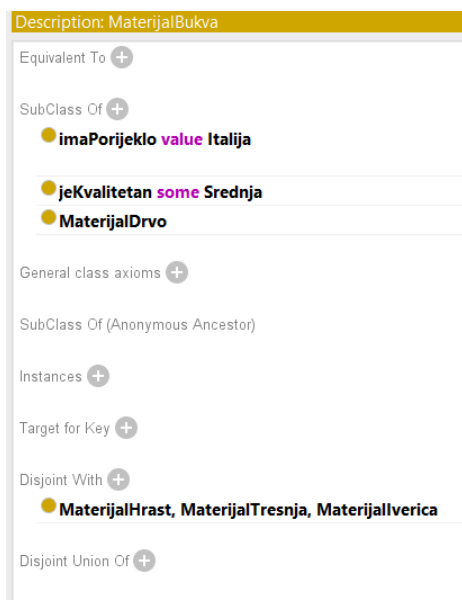
Ovaj upit sumira cijenu namještaja te ju prikazuje u stupcu ukupna cijena, a osim toga broji i broj individua klase *NamjestajStolica* i *NamjestajStol*. Agregira podatke o vrstama namještaja koristeći *UNION*. *BIND* stavlja vrijednost varijable *?tipNamjestaja* na *Stolica* ili *Stol*. *GROUP BY* grupira vrijednosti posebno za stolice i posebno za stol te se svakoj od tih grupa računa ukupna cijena i broj individua. *ORDER BY* omogućuju da rezultati budu poredani po vrijednosti varijable *?tipNamjestaja*.

U nastavku je prikazana ontologija kroz slike.

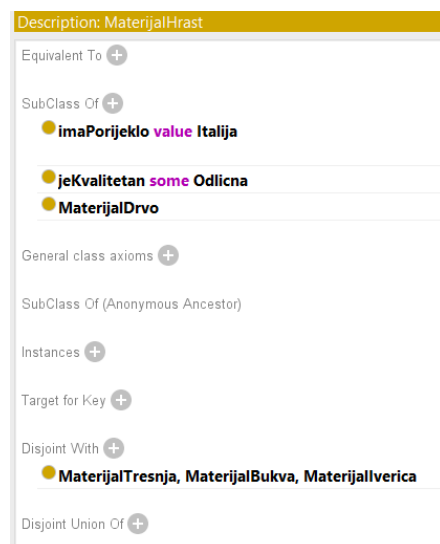




Slika 6. Hijerarhija (samostalna izrada)



Slika 7. MaterijalBukva (samostalna izrada)



Slika 8. MaterijalHrast (samostalna izrada)

Description: MaterijalIverica
Equivalent To <span>+</span>
SubClass Of <span>+</span>
<ul style="list-style-type: none"> <li>● imaPorijeklo <b>value</b> Njemacka</li> <li>● jeKvalitetan <b>some</b> Srednja</li> <li>● MaterijalDrvo</li> </ul>
General class axioms <span>+</span>
SubClass Of (Anonymous Ancestor)
Instances <span>+</span>
Target for Key <span>+</span>
Disjoint With <span>+</span>
<ul style="list-style-type: none"> <li>● MaterijalHrast, MaterijalTresnja, MaterijalBukva</li> </ul>
Disjoint Union Of <span>+</span>

Slika 9. MaterijalIverica (samostalna izrada)

Description: MaterijalTresnja
Equivalent To <span>+</span>
SubClass Of <span>+</span>
<ul style="list-style-type: none"> <li>● imaPorijeklo <b>value</b> BiH</li> <li>● jeKvalitetan <b>some</b> Losa</li> <li>● MaterijalDrvo</li> </ul>
General class axioms <span>+</span>
SubClass Of (Anonymous Ancestor)
Instances <span>+</span>
Target for Key <span>+</span>
Disjoint With <span>+</span>
<ul style="list-style-type: none"> <li>● MaterijalHrast, MaterijalBukva, MaterijalIverica</li> </ul>
Disjoint Union Of <span>+</span>

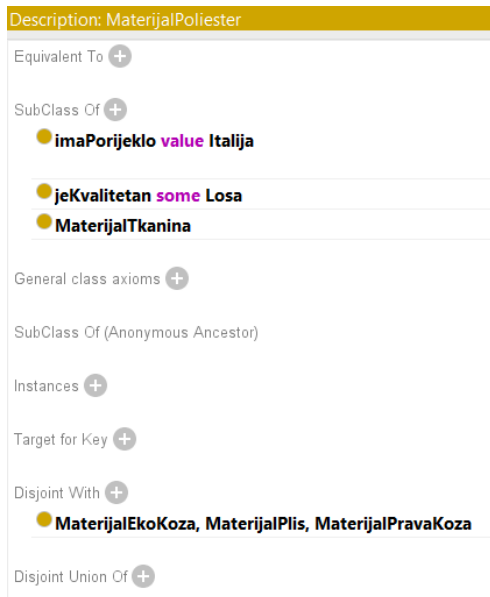
Slika 10. MaterijalTresnja (samostalna izrada)

Description: MaterijalEkoKoza
Equivalent To <span>+</span>
SubClass Of <span>+</span>
<ul style="list-style-type: none"> <li>● imaPorijeklo <b>value</b> BiH</li> <li>● jeKvalitetan <b>some</b> Losa</li> <li>● MaterijalTkanina</li> </ul>
General class axioms <span>+</span>
SubClass Of (Anonymous Ancestor)
Instances <span>+</span>
Target for Key <span>+</span>
Disjoint With <span>+</span>
<ul style="list-style-type: none"> <li>● MaterijalPoliester, MaterijalPlis, MaterijalPravaKoza</li> </ul>
Disjoint Union Of <span>+</span>

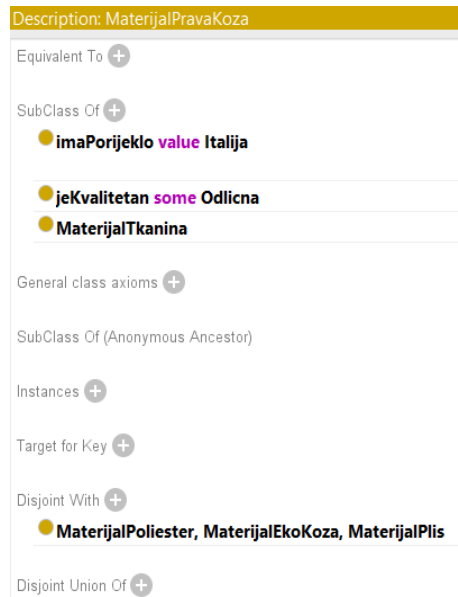
Slika 11. MaterijalEkoKoza (samostalna izrada)

Description: MaterijalPlis
Equivalent To <span>+</span>
SubClass Of <span>+</span>
<ul style="list-style-type: none"> <li>● imaPorijeklo <b>value</b> BiH</li> <li>● jeKvalitetan <b>some</b> Srednja</li> <li>● MaterijalTkanina</li> </ul>
General class axioms <span>+</span>
SubClass Of (Anonymous Ancestor)
Instances <span>+</span>
Target for Key <span>+</span>
Disjoint With <span>+</span>
<ul style="list-style-type: none"> <li>● MaterijalPoliester, MaterijalEkoKoza, MaterijalPravaKoza</li> </ul>
Disjoint Union Of <span>+</span>

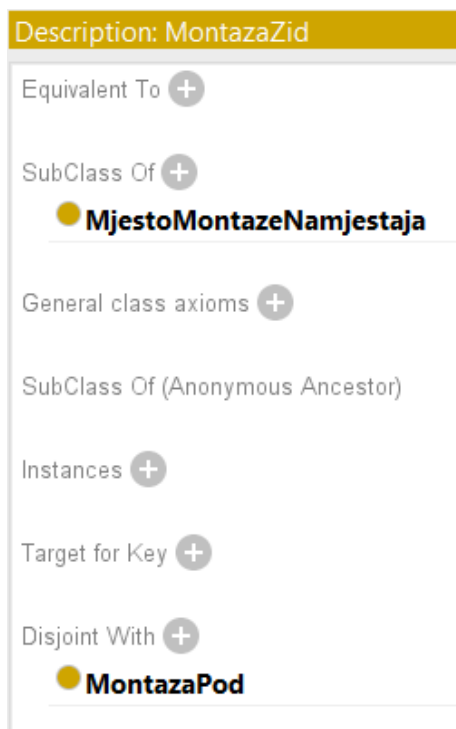
Slika 12. MaterijalPlis (samostalna izrada)



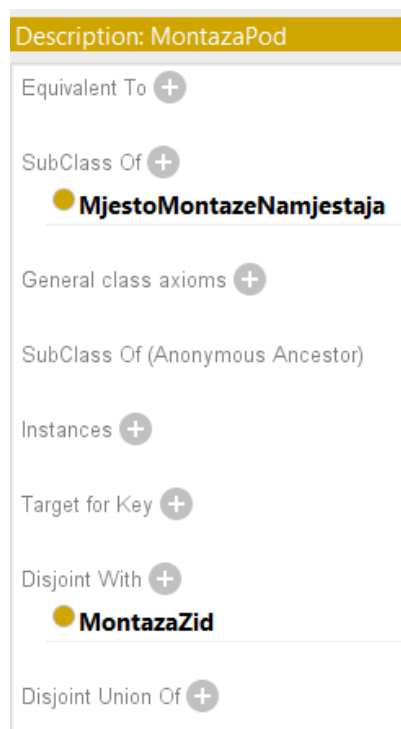
Slika 13. MaterijalPoliester (samostalna izrada)



Slika 14. MaterijalPravaKoza (samostalna izrada)



Slika 15. MontazaZid (samostalna izrada)



Slika 16. MontazaPod (samostalna izrada)

Description: NamjestajOrmar
Equivalent To <span>+</span>
SubClass Of <span>+</span>
<ul style="list-style-type: none"> <li>● imaMaterijal <b>only</b>(MaterijalIverica <b>or</b> MaterijalBukva)</li> <li>● imaMaterijal <b>some</b> MaterijalBukva</li> <li>● imaMaterijal <b>some</b> MaterijalIverica</li> <li>● NamjestajOdredjeni</li> </ul>
General class axioms <span>+</span>
SubClass Of (Anonymous Ancestor)
<ul style="list-style-type: none"> <li>● imaMjestoMontaze <b>some</b> MjestoMontazeNamjestaja</li> </ul>
Instances <span>+</span>
<ul style="list-style-type: none"> <li>◆ OrmarDvokrilni</li> <li>◆ OrmarTrokrilni</li> </ul>
Target for Key <span>+</span>
Disjoint With <span>+</span>
<ul style="list-style-type: none"> <li>● NamjestajStol, NamjestajStolica, NamjestajViseciKuhinjskiElementi</li> </ul>

Slika 17. NamjestajOrmar (samostalna izrada)

Description: NamjestajStol
Equivalent To <span>+</span>
SubClass Of <span>+</span>
<ul style="list-style-type: none"> <li>● imaMaterijal <b>only</b>(MaterijalIverica <b>or</b> MaterijalTresnja)</li> <li>● imaMaterijal <b>some</b> MaterijalIverica</li> <li>● imaMaterijal <b>some</b> MaterijalTresnja</li> <li>● NamjestajOdredjeni</li> </ul>
General class axioms <span>+</span>
SubClass Of (Anonymous Ancestor)
<ul style="list-style-type: none"> <li>● imaMjestoMontaze <b>some</b> MjestoMontazeNamjestaja</li> </ul>
Instances <span>+</span>
<ul style="list-style-type: none"> <li>◆ StolBlagavaonski</li> <li>◆ StolRadni</li> </ul>
Target for Key <span>+</span>
Disjoint With <span>+</span>
<ul style="list-style-type: none"> <li>● NamjestajStolica, NamjestajOrmar, NamjestajViseciKuhinjskiElementi</li> </ul>
Disjoint Union Of <span>+</span>

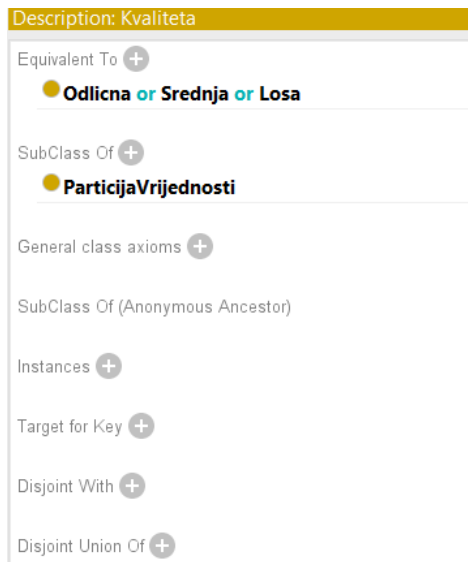
Slika 18. NamjestajStol (samostalna izrada)

Description: NamjestajStolica
Equivalent To <span>+</span>
SubClass Of <span>+</span>
<ul style="list-style-type: none"> <li>● imaMaterijal <b>only</b>(MaterijalHrast <b>or</b> MaterijalPravaKoza)</li> <li>● imaMaterijal <b>some</b> MaterijalHrast</li> <li>● imaMaterijal <b>some</b> MaterijalPravaKoza</li> <li>● NamjestajOdredjeni</li> </ul>
General class axioms <span>+</span>
SubClass Of (Anonymous Ancestor)
<ul style="list-style-type: none"> <li>● imaMjestoMontaze <b>some</b> MjestoMontazeNamjestaja</li> </ul>
Instances <span>+</span>
<ul style="list-style-type: none"> <li>◆ StolicaBarska</li> <li>◆ StolicaBlagavaonska</li> </ul>
Target for Key <span>+</span>
Disjoint With <span>+</span>
<ul style="list-style-type: none"> <li>● NamjestajStol, NamjestajOrmar, NamjestajViseciKuhinjskiElementi</li> </ul>

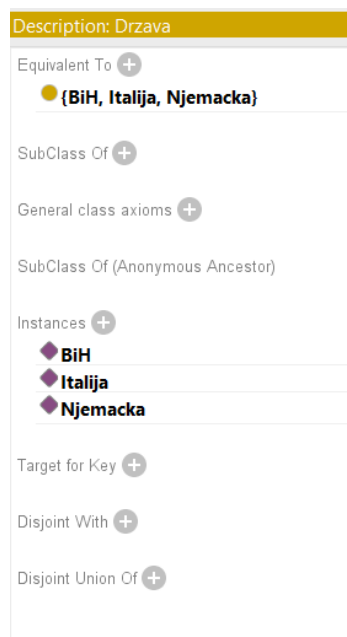
Slika 19. NamjestajStolica (samostalna izrada)

Description: NamjestajViseciKuhinjskiElementi
Equivalent To <span>+</span>
SubClass Of <span>+</span>
<ul style="list-style-type: none"> <li>● imaMaterijal <b>only</b>(MaterijalIverica <b>or</b> MaterijalTresnja <b>or</b> MaterijalPoliestar)</li> <li>● imaMaterijal <b>some</b> MaterijalIverica</li> <li>● imaMaterijal <b>some</b> MaterijalPoliestar</li> <li>● imaMaterijal <b>some</b> MaterijalTresnja</li> <li>● NamjestajOdredjeni</li> </ul>
General class axioms <span>+</span>
SubClass Of (Anonymous Ancestor)
<ul style="list-style-type: none"> <li>● imaMjestoMontaze <b>some</b> MjestoMontazeNamjestaja</li> </ul>
Instances <span>+</span>
<ul style="list-style-type: none"> <li>◆ ViseciElementMali</li> <li>◆ ViseciElementVeliki</li> </ul>
Target for Key <span>+</span>
Disjoint With <span>+</span>
<ul style="list-style-type: none"> <li>● NamjestajStol, NamjestajStolica, NamjestajOrmar</li> </ul>

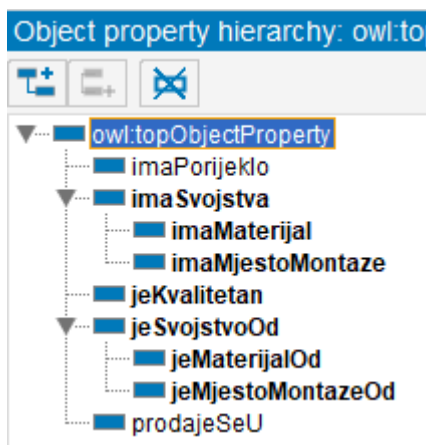
Slika 20. NamjestajViseciKuhinjskiElementi (samostalna izrada)



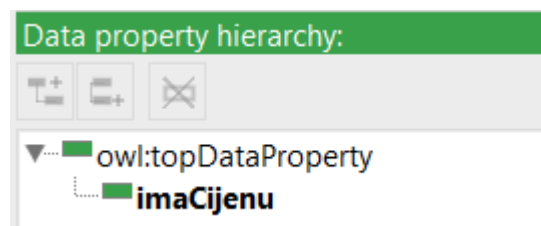
Slika 21. Kvaliteta (samostalna izrada)



Slika 22. Drzava (samostalna izrada)



Slika 23. Objektna svojstva (samostalna izrada)



Slika 24. Podatkovna svojstva (samostalna izrada)

The screenshot displays the Protégé ontology editor interface. On the left, the 'Class hierarchy' pane shows a tree structure starting from 'owl:Thing', with 'NamjestajStol' highlighted. The main workspace is divided into three panes: 'Annotations' (empty), 'Description: StolBlagavaonski' (showing 'Types' with 'NamjestajStol' selected), and 'Property assertions: StolBlagavaonski' (showing 'Object property assertions' with 'prodajeSeU BiH' and 'Data property assertions' with 'imaCijenu 460'). The bottom pane shows 'Direct instances: StolBlagavaonski' with 'StolBlagavaonski' and 'StolRadni' listed.

Slika 25. Instance NamjestajStol (samostalna izrada)

The screenshot displays the Protégé ontology editor interface. On the left, the 'Class hierarchy' pane shows a tree structure starting from 'owl:Thing', with 'NamjestajOrmar' highlighted. The main workspace is divided into three panes: 'Annotations' (empty), 'Description: OrmarDvokrilni' (showing 'Types' with 'NamjestajOrmar' selected), and 'Property assertions: OrmarDvokrilni' (showing 'Object property assertions' with 'prodajeSeU Italija' and 'Data property assertions' with 'imaCijenu 320'). The bottom pane shows 'Direct instances: OrmarDvokrilni' with 'OrmarDječji', 'OrmarDvokrilni', and 'OrmarTrokrilni' listed.

Slika 26. Instance NamjestajOrmar (samostalna izrada)

Class hierarchy: NamjestajStolica

Annotations: StolicaBlagavaonska

Annotations: +

Description: StolicaBlagavaonska

Property assertions: StolicaBlagavaonska

Types: +

NamjestajStolica

Same Individual As: +

Different Individuals: +

Object property assertions: +

prodajeSeU Njemacka

Data property assertions: +

imaCijenu 150

Negative object property assertions: +

Negative data property assertions: +

Direct instances: StolicaBlagava

For: NamjestajStolica

StolicaBarska

StolicaBlagavaonska

Slika 27. Instance NamjestajStolica (samostalna izrada)

Class hierarchy: NamjestajViseci

Annotations: ViseciElementMali

Annotations: +

Description: ViseciElementMali

Property assertions: ViseciElementMali

Types: +

NamjestajViseciKuhinjskiElementi

Same Individual As: +

Different Individuals: +

Object property assertions: +

prodajeSeU Italija

Data property assertions: +

imaCijenu 120

Negative object property assertions: +

Negative data property assertions: +

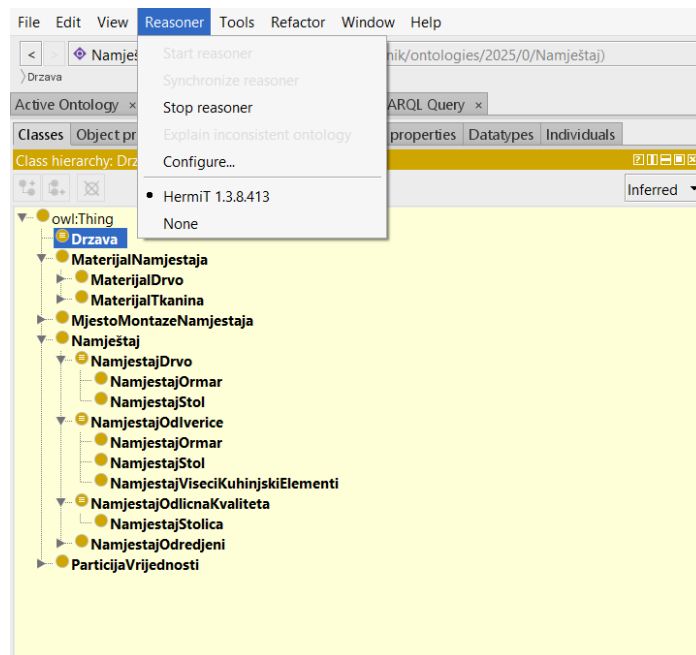
Direct instances: ViseciElementM

For: NamjestajViseciKuhinjskiElementi

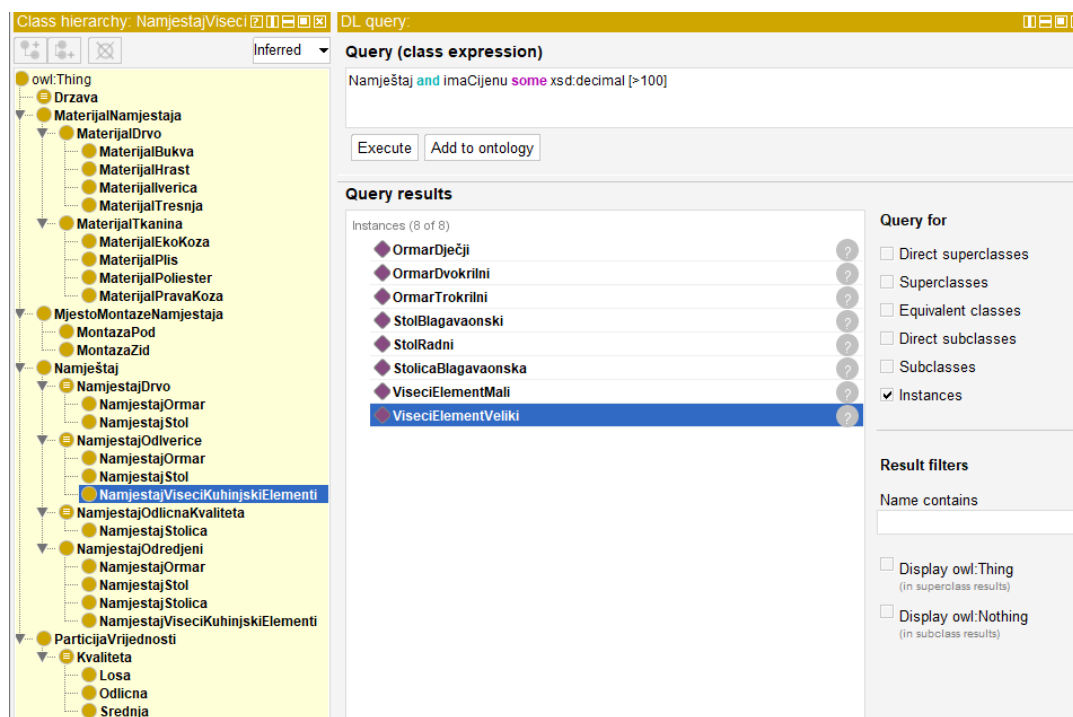
ViseciElementMali

ViseciElementVeliki

Slika 28. Instance NamjestajViseciKuhinjskiElementi (samostalna izrada)



Slika 29. Pokrenut reasoner (samostalna izrada)



Slika 30. Prvi DL jednostavni upit (samostalna izrada)



Class hierarchy: NamjestajViseci

DL query:

Query (class expression)

MaterijalNamjestaja and jeKvalitetan some Srednja

Execute Add to ontology

Query results

Subclasses (3 of 4)

- MaterijalBukva
- MaterijalVerica
- MaterijalPilis

Query for

- ☐ Direct superclasses
- ☐ Superclasses
- ☐ Equivalent classes
- ☐ Direct subclasses
- ☒ Subclasses
- ☐ Instances

Result filters

Name contains

☐ Display owl:Thing (in superclass results)

☐ Display owl:Nothing (in subclass results)

Slika 31. Drugi DL jednostavni upit (samostalna izrada)

Class hierarchy: MaterijalPravaKoza

DL query:

Query (class expression)

imaMaterijal some (MaterijalDrvo and jeKvalitetan some Odlicna)

Execute Add to ontology

Query results

Direct subclasses (1 of 1)

- NamjestajStolica

Subclasses (1 of 2)

- NamjestajStolica

Instances (2 of 2)

- StolicaBarska
- StolicaBlagavaonska

Query for

- ☐ Direct superclasses
- ☐ Superclasses
- ☐ Equivalent classes
- ☒ Direct subclasses
- ☒ Subclasses
- ☒ Instances

Slika 32. Treci DL jednostavni upit (samostalna izrada)

Class hierarchy: NamjestajOrmar

DL query:

Query (class expression)

(MaterijalDrvo and jeKvalitetan some Srednja) and (MaterijalDrvo and imaPorijeklo value Italija)

Execute Add to ontology

Query results

Equivalent classes (0 of 0)

Superclasses (2 of 3)

- MaterijalDrvo
- MaterijalNamjestaja

Direct superclasses (1 of 1)

- MaterijalDrvo

Direct subclasses (1 of 1)

- MaterijalBukva

Subclasses (1 of 2)

- MaterijalBukva

Instances (0 of 0)

Query for

- ☒ Direct superclasses
- ☒ Superclasses
- ☒ Equivalent classes
- ☒ Direct subclasses
- ☒ Subclasses
- ☒ Instances

Result filters

Name contains

☐ Display owl:Thing (in superclass results)

☐ Display owl:Nothing (in subclass results)

Slika 33. Prvi DL slozeni upit (samostalna izrada)

Class hierarchy: NamjestajOrmai

Inferred

- owl:Thing
  - Drzava
    - MaterijalNamjestaja
      - MaterijalDrvo
        - MaterijalBukva
        - MaterijalHrast
        - MaterijalIverica
        - MaterijalTresnja
        - MaterijalTkanina
          - MaterijalEkoKoza
          - MaterijalPlis
          - MaterijalPoliester
          - MaterijalPravaKoza
        - MjestoMontazeNamjestaja
          - MontazaPod
          - MontazaZid
        - Namjestaj
          - NamjestajDrvo
            - NamjestajOrmar**
            - NamjestajStol
          - NamjestajOdliverice
            - NamjestajOrmar
            - NamjestajStol
          - NamjestajViseciKuhinjskiElementi
            - NamjestajOdlisnaKvaliteta
              - NamjestajStolica
            - NamjestajOdrjednjeni
              - NamjestajOrmar
              - NamjestajStol
              - NamjestajStolica
              - NamjestajViseciKuhinjskiElementi
          - ParticijaVrijednosti
            - Kvaliteta
              - Losa
              - Odlisna
              - Srednja

DL query

Query (class expression)

(NamjestajOdrjednjeni and prodajeSeU value Njemacka) and (NamjestajOdrjednjeni and imaCijenu some xsd:decimal [< 200])

Execute

Add to ontology

Query results

Instances (2 of 2)

- StolicaBarska
- StolicaBlagavaonska

Query for

☐ Direct superclasses  
☐ Superclasses  
☐ Equivalent classes  
☐ Direct subclasses  
☒ Subclasses  
☒ Instances

Result filters

Name contains

☐ Display owl:Thing  
(in superclass results)  
☐ Display owl:Nothing  
(in subclass results)

Slika 34. Drugi DL složeni upit (samostalna izrada)

Class hierarchy: NamjestajOrmai

Inferred

- owl:Thing
  - Drzava
    - MaterijalNamjestaja
      - MaterijalDrvo
        - MaterijalBukva
        - MaterijalHrast
        - MaterijalIverica
        - MaterijalTresnja
        - MaterijalTkanina
          - MaterijalEkoKoza
          - MaterijalPlis
          - MaterijalPoliester
          - MaterijalPravaKoza
        - MjestoMontazeNamjestaja
          - MontazaPod
          - MontazaZid
        - Namjestaj
          - NamjestajDrvo
            - NamjestajOrmar**
            - NamjestajStol
          - NamjestajOdliverice
            - NamjestajOrmar
            - NamjestajStol
          - NamjestajViseciKuhinjskiElementi
            - NamjestajOdlisnaKvaliteta
              - NamjestajStolica
            - NamjestajOdrjednjeni
              - NamjestajOrmar
              - NamjestajStol
              - NamjestajStolica
              - NamjestajViseciKuhinjskiElementi
          - ParticijaVrijednosti
            - Kvaliteta
              - Losa
              - Odlisna
              - Srednja

DL query

Query (class expression)

((NamjestajOdrjednjeni and prodajeSeU value Njemacka) or (NamjestajOdrjednjeni and prodajeSeU value Italija)) and (NamjestajOdrjednjeni and imaCijenu some xsd:decimal [< 200])

Execute

Add to ontology

Query results

Instances (3 of 3)

- StolicaBarska
- StolicaBlagavaonska
- ViseciElementMali

Query for

☐ Direct superclasses  
☐ Superclasses  
☐ Equivalent classes  
☐ Direct subclasses  
☐ Subclasses  
☒ Instances

Result filters

Name contains

☐ Display owl:Thing  
(in superclass results)  
☐ Display owl:Nothing  
(in subclass results)

Slika 35. Treći DL složeni upit (samostalna izrada)

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX namjestaj: <http://www.semanticweb.org/korisnik/ontologies/2025/0/Namjestaj#>
SELECT ?individua ?cijena
WHERE {
?individua rdfs:type namjestaj:NamjestajOrmar;
namjestaj:imaCijenu ?cijena
}

```

individua	cijena
OrmarTroklini	"600"^^<http://www.w3.org/2001/XMLSchema#decimal>
OrmarDvoklini	"320"^^<http://www.w3.org/2001/XMLSchema#decimal>
OrmarDječji	"350"^^<http://www.w3.org/2001/XMLSchema#integer>

Slika 36. Prvi SPARQL jednostavni upit (samostalna izrada)

SPARQL query:				
<pre> PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX owl: &lt;http://www.w3.org/2002/07/owl#&gt; PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt; PREFIX namjestaj: &lt;http://www.semanticweb.org/korisnik/ontologies/2025/0/Namjestaj#&gt; SELECT ?podvojstvo ?domena ?doseg WHERE { ?podvojstvo rdfs:subPropertyOf namjestaj:jeSvojstvoOd; rdfs:domain ?domena; rdfs:range ?doseg; </pre>				
	podvojstvo		domena	doseg
jeMjestoMontazeOd		MjestoMontazeNamjestaja		Namjestaj
jeMaterijalOd		MaterijalNamjestaja		Namjestaj

Slika 37. Drugi SPARQL jednostavni upit (samostalna izrada)

SPARQL query:	
<pre> PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX owl: &lt;http://www.w3.org/2002/07/owl#&gt; PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt; PREFIX namjestaj: &lt;http://www.semanticweb.org/korisnik/ontologies/2025/0/Namjestaj#&gt; SELECT ?potklasa WHERE { ?potklasa rdfs:subClassOf namjestaj:MaterijalTkanina </pre>	
	potklasa
MaterijalEkoKoza MaterijalPlis MaterijalPravaKoza MaterijalPoliester	

Slika 38. Treći SPARQL jednostavni upit (samostalna izrada)

SPARQL query:			
<pre> PREFIX rdf: &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX owl: &lt;http://www.w3.org/2002/07/owl#&gt; PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; PREFIX xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt; PREFIX namjestaj: &lt;http://www.semanticweb.org/korisnik/ontologies/2025/0/Namjestaj#&gt; SELECT ?individua ?cijena ?prodaja WHERE { ?individua rdf:type namjestaj:NamjestajOrmar; namjestaj:imaCijenu ?cijena.  OPTIONAL { ?individua namjestaj:prodajeSeU ?prodaja. }  FILTER (?cijena &gt; 100) } </pre>			
individua	cijena	prodaja	
OrmarTrokrilni	"600"^^<http://www.w3.org/2001/XMLSchema#decimal>	BiH	
OrmarDvokrilni	"320"^^<http://www.w3.org/2001/XMLSchema#decimal>	Italija	
OrmarDjeGji	"350"^^<http://www.w3.org/2001/XMLSchema#integer>	Italija	

Slika 39. Prvi SPARQL složeni upit (samostalna izrada)

SPARQL query:		
<pre> PREFIX xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt; PREFIX namjestaj: &lt;http://www.semanticweb.org/korisnik/ontologies/2025/0/Namjestaj#&gt; SELECT ?tipNamjestaja (SUM(?cijena) AS ?ukupnaCijena) (COUNT(?individua) AS ?brojIndividua) WHERE {   {     ?individua rdf:type namjestaj:NamjestajStolica;       namjestaj:imaCijenu ?cijena.     BIND("Stolica" AS ?tipNamjestaja)   }   UNION   {     ?individua rdf:type namjestaj:NamjestajStol;       namjestaj:imaCijenu ?cijena.     BIND("Stol" AS ?tipNamjestaja)   } } GROUP BY ?tipNamjestaja ORDER BY ?tipNamjestaja </pre>		
tipNamjestaja	ukupnaCijena	brojIndividua
"Stol"	"710"^^<http://www.w3.org/2001/XMLSchema#decimal>	"2"^^<http://www.w3.org/2001/XMLSchema#integer>
"Stolica"	"225"^^<http://www.w3.org/2001/XMLSchema#decimal>	"2"^^<http://www.w3.org/2001/XMLSchema#integer>

Slika 40. Drugi SPARQL složeni upit (samostalna izrada)

## 7. Zaključak

Kroz analizu teorijskih i praktičnih aspekata učenja ontologija ovaj rad naglašava značaj ovih metoda u suvremenim informacijskim sustavima. Učenje ontologija ne samo da ubrzava proces razvoja formalnih ontologija već i smanjuje ljudske pogreške te omogućuje otkrivanje novih znanja. Predstavljeni primjeri alata i platformi kao što su Protégé i OntoGen ilustriraju napredak u automatizaciji i standardizaciji ovog procesa. Iako izazovi poput heterogenosti podataka, skalabilnosti i evaluacije ontologija ostaju otvoreni, kontinuirani napredak u strojnome učenju i obradi prirodnog jezika obećava daljnji razvoj ovog područja. U konačnici, učenje ontologija predstavlja ključnu komponentu u stvaranju pametnih sustava koji podržavaju interoperabilnost podataka, zaključivanje i donošenje odluka u različitim industrijama.

U radu je prikazan primjer ontologije napravljene u alatu Protege. Za tu ontologiju provedeni su jednostavniji i složeniji upiti kako bi se pokazalo samo zaključivanje u tom alatu, a uz to u ontologiju su uključene i definirane klase kroz koje se može vidjeti da alat na osnovu određenih ograničenja donosi zaključke te na taj način omogućava učenje ontologija.

## Popis literature

- [1] Maedche, A., Staab, S., (2004). *Ontology Learning*, Preuzeto 30.12.2024. s [https://link.springer.com/chapter/10.1007/978-3-540-24750-0\\_9](https://link.springer.com/chapter/10.1007/978-3-540-24750-0_9)
- [2] Zhou, L., (24.3.2007). *Ontology learning: state of the art and open issues*, Preuzeto 30.12.2024. s <https://link.springer.com/article/10.1007/s10799-007-0019-5>
- [3] Banek, M., (2013). Ontologije i semantički web, Preuzeto 30.12.2024. s [https://www.fer.unizg.hr/download/repository/Ontologije\\_i\\_Semanticki\\_Web-MBanek%5B2%5D.pdf?](https://www.fer.unizg.hr/download/repository/Ontologije_i_Semanticki_Web-MBanek%5B2%5D.pdf?)
- [4] Lehmann, J., Volker, J., (2014). An Introduction to Ontology Learning, Preuzeto 30.12.2024. s [https://jens-lehmann.org/files/2014/pol\\_introduction.pdf](https://jens-lehmann.org/files/2014/pol_introduction.pdf)
- [5] Putica, M., (20.2.2018). Semantički web, Preuzeto 30.12.2024. s <https://hrcak.srce.hr/file/310630>
- [6] Journal, I. (n.d.). *ONTOLOGY LEARNING*. Preuzeto 12.01.2025. s [https://www.academia.edu/34458362/ONTOLOGY\\_LEARNING](https://www.academia.edu/34458362/ONTOLOGY_LEARNING)
- [7] (73) 5.3 *Ontology Learning—YouTube*. (n.d.). Preuzeto 12.01.2025. s [https://www.youtube.com/watch?v=IXgex750Acs&ab\\_channel=OpenHPITutorials](https://www.youtube.com/watch?v=IXgex750Acs&ab_channel=OpenHPITutorials)
- [8] Juri, D. (n.d.). *Metode učenja ontologija—Trenutno stanje i problemi*. Preuzeto 12.01.2025. s <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=7eeccefe03a9358f6217c54e37ff4c1c1e493710>
- [9] Asim, M. N., Wasim, M., Khan, M. U. G., Mahmood, W., & Abbasi, H. M. (2018). A survey of ontology learning techniques and applications. *Database*, 2018, bay101. Preuzeto 12.01.2025. s <https://doi.org/10.1093/database/bay101>
- [10] d'Aquin, M. (2012). Modularizing Ontologies. In M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, & A. Gangemi (Eds.), *Ontology Engineering in a Networked World* (pp. 213–233). Springer Berlin Heidelberg. Preuzeto 12.01.2025. s [https://doi.org/10.1007/978-3-642-24794-1\\_10](https://doi.org/10.1007/978-3-642-24794-1_10)
- [11] Motta, E., Peroni, S., Gómez-Pérez, J. M., d'Aquin, M., & Li, N. (2012). Visualizing and Navigating Ontologies with KC-Viz. In M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, &

A. Gangemi (Eds.), *Ontology Engineering in a Networked World* (pp. 343–362). Springer Berlin Heidelberg. Preuzeto 12.01.2025. s [https://doi.org/10.1007/978-3-642-24794-1\\_16](https://doi.org/10.1007/978-3-642-24794-1_16)

[12] Otto, M. (n.d.). Ontogen. Ontogen. Preuzeto 12.01.2025. s <https://ontogen.io/>

[13] Protégé. (n.d.) Preuzeto 12.01.2025. s <https://protege.stanford.edu/>

[14] Topbraid Composer. (n.d.). AllegroGraph. Preuzeto 12.01.2025. s <https://allegrograph.com/topbraid-composer/>

[15] Kapoor, B. (2010). *A Comparative Study of Ontology building tools in Semantic Web Applications*. Preuzeto 12.01.2025.  
[https://www.academia.edu/124473502/A\\_Comparative\\_Study\\_of\\_Ontology\\_building\\_Tools\\_in\\_Semantic\\_Web\\_Applications](https://www.academia.edu/124473502/A_Comparative_Study_of_Ontology_building_Tools_in_Semantic_Web_Applications)

Slike:

Slika 1. Journal, I. (*Fig. 2: Ontology Learning Cycle*) Preuzeto 12.01.2025. s [https://www.academia.edu/34458362/ONTOLOGY\\_LEARNING](https://www.academia.edu/34458362/ONTOLOGY_LEARNING)

Slika 2. Journal, I. (*Fig. 3: Ontology Learning Architecture*) Preuzeto 12.01.2025. s [https://www.academia.edu/34458362/ONTOLOGY\\_LEARNING](https://www.academia.edu/34458362/ONTOLOGY_LEARNING)

Slika 4. *File:WP-classes.png—Protege Wiki*. (n.d.). Preuzeto 12.01.2025. s <https://protegewiki.stanford.edu/wiki/File:WP-classes.png>

## Popis slika

Slika 1. Slojevi učenja ontologije (I. (n.d.). , str. 1220 ).	10
Slika 2. Arhitektura učenja ontologija (I. (n.d.). , str. 1220 ).	11
Slika 3. NeOn Toolkits ( <a href="https://www.researchgate.net/figure/The-NeOn-Toolkit-illustrates-the-problems-of-occlusion-and-overcrowding-of-visual_fig3_234129385">https://www.researchgate.net/figure/The-NeOn-Toolkit-illustrates-the-problems-of-occlusion-and-overcrowding-of-visual_fig3_234129385</a> )	16
Slika 4. Web Protege (File:WP-classes.png—Protege Wiki, bez dat.)	17
Slika 5. TopBraid ( <a href="https://franz.com/agraph/tbc/">https://franz.com/agraph/tbc/</a> )	17
Slika 6. Hijerarhija (samostalna izrada).	22
Slika 7. MaterijalBukva (samostalna izrada)	22
<i>Slika 8. MaterijalHrast (samostalna izrada)</i>	22
Slika 9. MaterijalIverica (samostalna izrada)	23
<i>Slika 10. MaterijalTresnja (samostalna izrada)</i>	23
Slika 11. MaterijalEkoKoza (samostalna izrada)	23
Slika 12. MaterijalPlis (samostalna izrada)	23
Slika 13. MaterijalPoliester (samostalna izrada)	24
Slika 14. MaterijalPravaKoza (samostalna izrada)	24
Slika 15. MontazaZid (samostalna izrada)	24
Slika 16. MontazaPod (samostalna izrada).	24
Slika 17. NamjestajOrmar (samostalna izrada)	25
Slika 18. NamjestajStol (samostalna izrada)	25
Slika 19. NamjestajStolica (samostalna izrada)	25
Slika 20. NamjestajViseciKuhinjskiElementi (samostalna izrada)	25
Slika 21. Kvaliteta (samostalna izrada)	26
Slika 22. Drzava (samostalna izrada)	26
Slika 23. Objektna svojstva (samostalna izrada)	26
Slika 24. Podatkovna svojstva (samostalna izrada)	26
Slika 25. Instance NamjestajStol (samostalna izrada)	27
Slika 26. Instance NamjestajOrmar (samostalna izrada)	27
Slika 27. Instance NamjestajStolica (samostalna izrada)	28
Slika 28. Instance NamjestajViseciKuhinjskiElementi (samostalna izrada)	28
Slika 29. Pokrenut reasoner (samostalna izrada)	29
Slika 30. Prvi DL jednostavni upit (samostalna izrada)	29
Slika 31. Drugi DL jednostavni upit (samostalna izrada)	30
Slika 32. Treci DL jednostavni upit (samostalna izrada)	30
Slika 33. Prvi DL složeni upit (samostalna izrada)	30



Slika 34. Drugi DL složeni upit (samostalna izrada) .....	31
Slika 35. Treći DL složeni upit (samostalna izrada) .....	31
Slika 36. Prvi SPARQL jednostavni upit (samostalna izrada).....	31
Slika 37. Drugi SPARQL jednostavni upit (samostalna izrada) .....	32
Slika 38. Treći SPARQL jednostavni upit (samostalna izrada).....	32
Slika 39. Prvi SPARQL složeni upit (samostalna izrada) .....	32
Slika 40. Drugi SPARQL složeni upit (samostalna izrada) .....	33

## Prilozi

Ucenje\_Ontologija-Kajzogaj\_Udovcic.owl