Alati za izradu ontologija

Peurača, Davor

Sveučilište J.J. Strossmayera u Osijeku

Filozofski fakultet

Diplomski studij nakladništva i informacijske tehnologije

Davor Peurača

Alati za izradu ontologija

Diplomski rad

Mentor: izv. prof. dr. sc. Boris Bosančić

Osijek, 2021.

Sveučilište J.J. Strossmayera u Osijeku
Filozofski fakultet Osijek
Odsjek/samostalna katedra
Diplomski studij nakladništva i informacijske tehnologije

Davor Peurača

Alati za izradu ontologija

Diplomski rad

Društvene znanosti, informacijske i komunikacijske znanosti, informacijski sustavi i informatologija

Mentor: izv. prof. dr. sc. Boris Bosančić

Osijek, 2021.

Izjava o akademskoj čestitosti i o suglasnosti za javno objavljivanje

IZJAVA

Izjavljujem s punom materijalnom i moralnom odgovornošću da sam ovaj rad samostalno napravio te da u njemu nema kopiranih ili prepisanih dijelova teksta tuđih radova, a da nisu označeni kao citati s napisanim izvorom odakle su preneseni.
Svojim vlastoručnim potpisom potvrđujem da sam suglasan da Filozofski fakultet Osijek trajno pohrani i javno objavi ovaj moj rad u internetskoj bazi završnih i diplomskih radova knjižnice Filozofskog fakulteta Osijek, knjižnice Sveučilišta Josipa Jurja Strossmayera u Osijeku i Nacionalne i sveučilišne knjižnice u Zagrebu.

U Osijeku, datum

Ivana Perica, 0010203166
ime i prezime studenta, JMBAG

Sažetak

Svrha rada je pružiti osvrt na alate za izradu ontologija. Ciljevi rada su usporediti trenutno dostupne alate za izradu ontologija i dati osvrt na njihove funkcionalnosti. U prvom dijelu rada se obrađuju definicije vezane uz ontologije. Nakon toga se prikazuju i opisuju alati za izradu ontologija: Protégé, WebProtégé, Fluent, OWLGrEd. Navedeni alati se zatim uspoređuju s obzirom na ponuđene funkcionalnosti i opisano korisničko iskustvo u korištenju alata. U zaključku se navodi da se područje daljnjeg razvoja alata za izradu ontologija nalazi u polju mogućnosti zajedničkog uređivanja ontologija.

Ključne riječi: alati za izradu ontologija, ontologije, Protégé, WebProtégé, sustav za upravljanje znanjem.

Sadržaj

1. Uvod

Ontologije postaju sve važnije u poljima kao što su upravljanje znanjem, integracija informacija, kooperativni informacijski sustavi, pronalaženje informacija i elektronička trgovina. Jedno od područja primjene koje je 2000-ih doživjelo eksploziju interesa je semantički web, na kojem ontologije igraju ključnu ulogu u uspostavljanju zajedničke terminologije između ljudskih i računalnih agenata, čime se osigurava da različiti agenti imaju zajedničko razumijevanje pojmova koji se koriste u semantičkom označavanju i pretraživanju. Alati koji se koriste za izradu ontologija vremenom su se mijenjali, a u ovom diplomskom radu vrši se njihova analiza.

Svrha rada je dati osvrt na alate za izradu ontologija. Ciljevi koji proizlaze iz svrhe rada su dati općenito osvrt na pojam ontologije, a onda i na pojedine alate za izradu ontologija, uz prikaz njihove usporedbe.

U skladu sa svrhom i ciljevima rada u drugom poglavlju se govori o definicijama ontologije i konceptima koji su bitni za razumijevanje ovog pojma, kao i tehnologijama semantičkog weba koje su povezane s njim. Objašnjeno je što je RDF i koja je njegova uloga u ontologijama. Također se daje uvid u SPARQL (*SPARQL Protocol and RDF Query Language*) upitni jezik te se obrazlaže kako je postao standardni upitni jezik u polju semantičkog weba. Objašnjava se RDFS (*RDF Schema*) kao univerzalni jezik opisa putem vlastitog rječnika. Zatim se daje uvid u OWL (*Web Ontology Language*), ontologijski jezik za semantički web, odnosno osnovni jezik za izradu ontologija s formalno definiranim značenjem. U trećem poglavlju se donosi opis i analiza alata za izradu ontologija. Opisuju se alati OWLGrEd, Fluent Editor, Protégé i WebProtégé. OWLGrED je alat koji ima jednostavno sučelje s omogućenom funkcionalnošću vizualizacije ontologija. Fluent Editor je alat za uređivanje i za izmjenu ontologija koji koristi ljudsko čitljivi jezik za kreiranje sučelja koje znatno olakšava izradu ontologija. Desktop Protégé alat za izradu ontologija je najstariji i najpopularniji alat trenutno na tržištu. Jedan je od najfleksibilnijih alata zbog mogućnosti dodataka koji se mogu uključiti u aplikaciju i dodatno proširiti funkcionalnosti izrade ontologija. WebProtégé je nov alat za izradu ontologija, nastao na desktop Protégéu, koji je razvio isti tim stručnjaka sa sveučilišta Stanford kao i desktop Protégé. Ovaj alat je izrađen primarno kako bi se kroz jednostavno sučelje omogućilo zajedničko uređivanje ontologija.

U zaključku se nalazi skraćeni prikaz analize alata i daje konačno mišljenje o trenutnom stanju razvoja alata za izradu ontologija.

2. Definicije i svrha ontologije

Riječ "ontologija" koristi se s različitim značenjima u različitim zajednicama. Najradikalnija je razlika između filozofskog značenja te riječi, koje ima dobro utemeljenu tradiciju, i računalnog značenja koje je nastalo u drugoj polovini 20. stoljeća u zajednici inženjera znanja. Rana neformalna definicija (računalnih) ontologija glasi: „eksplicitna specifikacija konceptualizacije".[1] U ovom radu se opisuju prethodni pokušaji pojašnjenja i formaliziranja definicije ontologije, te se raspravlja i o važnosti ostvarenja konsenzusa u razvoju ontologija.[2]

Što je konceptualizacija? Formalno znanje temelji se na konceptualizaciji predmeta, pojmova i drugih entiteta za koje se pretpostavlja da postoje u nekom području interesa i odnosa. Konceptualizacija je apstraktni, pojednostavljeni pogled na svijet koji se želi predstavljati s nekom svrhom. Svaka baza znanja, i sustav zasnovan na znanju je osmišljen na nekoj konceptualizaciji.[3] Ontologije su djelotvoran alat za sistematizaciju znanja iz različitih predmetnih područja. Kao izvorni sintetizatori znanja, ontologije bude veliko teorijsko i praktično zanimanje u području obrazovanja.[4] Kada se govori o ontologijama, mora se govoriti i o semantičkom webu koji je usko vezan s ontologijama.

Ideju semantičkog weba osmislio je Tim Berners-Lee, osnivač *World Wide Weba (WWW)*.[5] On je predvidio da će u budućnosti ogromna količina informacija na webu nositi strojno čitljive metapodatke, što će omogućiti računalima mogućnost automatske manipulacije sadržajem, bez ljudske intervencije. Stoga je semantički web zamišljen kao proširenje postojećeg weba, ali u kojem bi se informacijama dodijelilo značenje. To bi se postiglo primjenom naprednih tehnologija znanja na webu i distribuiranim sustavima općenito. Metode i tehnologije semantičkog weba omogućuju strojevima razumijevanje značenja ili "semantike" informacija na WWW-u. Da bi se to postiglo, informacije trebaju biti strukturirane, te popraćene nizom pravila koja računala mogu koristiti za provođenje automatiziranog logičkog rasuđivanja.

[1] Gruber, T.R. A Translation Approach to Portable Ontology Specifications. 2007. str. 2
[2] Usp. Handbook on ontologies, 2010. str. 22
[3] Isto. str. 24.
[4] Usp. Dolzhenkov, Valeriy Nikolaevich;Maltzagov, Daudovich Isa;Makarova, Igorevna, Aleksandra. Software Tools for Ontology Development. 2020. str. 936.
[5] Usp. Berners-Lee, T., Hendler, J., & Lassila, O. The semantic web. *Scientific american*, 2001, *284*(5), 34-43.

Prema Tim Berners Lee-ju, „…*web se razvio najbrže kao medij dokumenata za ljude, a ne informacija kojima se može manipulirati automatski.*“ [6] Popunjavanjem web stranica sa strojno razumljivim metapodacima i dodavanjem dokumenata koja će računala koristiti, postojeći web transformirat će se u semantički web. Računala će značenje semantičkih podataka pronaći putem hiperveza koje će ih povezati s definicijama ključnih pojmova te pravilima za logično rasuđivanje.[7]

Izrada ontologije može se podijeliti u nekoliko faza koje će arhitekta ontologije uputiti o obujmu i granicama ontologije. Dokumentacija izrade ontologije pruža informacije o fazama u razvoju ontologije i objašnjava komponente potrebne za predstavljanje informacija.

Faze u razvoju ontologije su sljedeće: *određivanje domene znanja* (kao početak razvoja ontologije potrebno je odrediti domenu znanja ontologije); zatim, *utvrđivanje klasa ontologije* i njihovo uređenje u taksonomiji hijerarhija (potklasa – superklasa). Sljedeći korak je *definiranje svojstava ontologije*, da bi iza toga uslijedilo *utvrđivanje aspekata aksioma u svojstvima*, odnosno domene i raspona svojstava. Svojstva povezuju instance u domeni i instance u rasponu. Peti korak je *izrada instanci ontologije*, i posljednji - *međusobno povezivanje instanci putem svojstava*.[8]

2.1 RDF

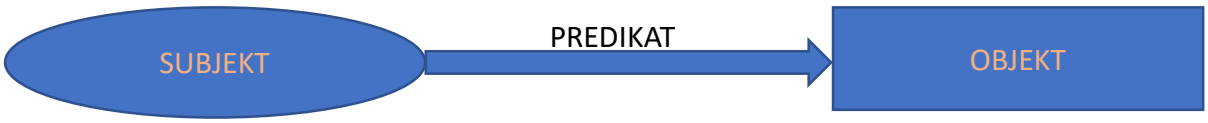
RDF je o domeni znanja neovisan označiteljski jezik koji osim strukturiranja nudi i mogućnost interpretacije podataka. Najčešći pristup serijalizaciji podataka odnosno njihovom označavanju ili enkodiranju u ontologijama je putem RDF-a (*Resource Description Framework*).[9]

RDF je strojno čitljiv i razumljiv format metapodataka koji nije namijenjen krajnjim korisnicima. RDF nije format pohrane, već predstavlja model za opis mrežnih izvora. RDF formati pohrane su: N3, N-Triples, Turtle i RDF/XML.

Osnovni koncepti RDF modela podataka su resursi, svojstva, izjave i grafovi. Resurs ili subjekt izjave je ono što se opisuje (mrežno sjedište, mrežna stranica). Svaki izvor mora imati URI/IRI (npr. https://web.ffos.hr). Svojstva predstavljaju specijalnu vrstu izvora, koja opisuju odnose između drugih izvora („napisana od”, „objavljena od” i sl.); svako svojstvo također mora imati

[6] Giri, Kaushal. Role of Ontology in Semantic Web. 2011. str. 117
[7] Usp. Giri, Kaushal. Role of Ontology in Semantic Web. 2011. str. 117
[8] Usp. Sari, Fitri, Riri. Implementation of Web Ontology and Semantic Application for Electronic Journal Citation System. 2010. str. 37.
[9] Usp. Handbook on ontologies, 2010. str. 489

URI/IRI. Izjave se sastoje od resursa, svojstva i vrijednosti svojstva ili objekta RDF izjave. Grafovi su imenovani elementi povezani imenovanim lukovima. Za RDF izjavu uobičajeno se kaže da se sastoji od subjekta, predikata i objekta (RDF Triplet - trolist).



Slika 1 – Grafički prikaz RDF izjave.

Na Slici 1 nalazi se grafički prikaz RDF izjave: subjekt se navodi u ovalnom obliku, strelica predstavlja predikat, dok se objekt RDF izjave nalazi u pravokutnom obliku ako je njegova vriejdnost slovna (engl. *literal*), a u ovalu, ako je neslovna (engl. *non-literal*).

2.2 SQL

Za isključivo relacijske podatke (za razliku od polustrukturiranih podataka), SQL je jedan od rasprostiranijih upitnih jezika, koji uključuje podršku za pohranu velike količine podataka, učinkovite sheme indeksiranja, optimizatore upita itd. Stoga bi bilo poželjno kada bi ovu robusnu i široko dostupnu tehnologiju mogli koristiti za svrhe slanja upita polustrukturiranim podacima. Nažalost, to može biti učinjeno samo otežano zbog velikog jaza između RDF podatkovnog modela i relacijskog modela podataka na kojem se temelji jezik upita. Da bi to ilustrirali, može se vidjeti kakav će scenarij tražiti XML implementacija u relacijskoj bazi podataka: kao prvi korak XML model podataka trebao bi se kodirati u relacijski model. To je moguće učiniti ako bi se svakom čvoru u XML stablu dodijelio jedinstveni identifikator uz svaki unos u relacijskoj bazi podataka koja povezuje takav čvor sa svim njegovim potomcima i atributima. Problemi počinju kada se to koristi kao osnova za postavljanje upita XML strukturi: svaki XML upit treba kompilirati u SQL upit na relacijskim tablicama. U pravilu, jedan XML upit (poput: "vratiti sve potomke zadanog čvora") mora se kompilirati u složen skup SQL upita. Tu nije jasno je li konačan skup SQL upita moguće generirati za svaki smislen XML upit. Iako kao najprikladnije kratkoročno rješenje, smatra se kako dugoročno to ipak nije prikladno rješenje. Dapače, tehnike za pohranu velikih podataka, sheme indeksiranja,

optimizatori upita itd., trebali bi počivati na drugačijem modelu podataka, umjesto da se oslanjaju na navedene tehnike koje su predviđene za relacijski model podataka.[10]

2.3 SPARQL

W3C preporučuje SPARQL upitni jezik za podatke pohranjene u RDF-u koji ubrzo postaje standardni upitni jezik na semantičkom webu. U rujnu 2006. gotovo svi glavni alati za izradu ontologija imaju mogućnost pohrane podataka u RDF-u, te započinju s implementacijom podrške za SPARQL. To je narušilo interoperabilnost. Upitni jezik SPARQL temelji se na podudaranju varijabli, onih u postavljenom upitu i onih koji čine sastavne dijelove RDF tripleta – subjekt, predikat i objekt.

Kao jednostavan primjer može se razmotriti sljedeći upit:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?c
WHERE
{
?c rdf:type rdfs:Class .
}
```

Gornji upit dohvaća sve RDF triplete u kojima je svojstvo rdf: type, a objekt rdfs: Class. Ovaj upit, kad se izvrši, dohvatit će sve klase u ontologiji.[11]

U sljedećem upitu kao rezultat prikazat će se sve instance klase Osoba (*Person*).

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?i
WHERE
{
?i rdf:type foaf:Person .
}
```

SPARQL se ne obvezuje podržati RDFS semantiku. Stoga rezultat upravo navedenog upita ovisi o tome je li sustav podržava RDFS semantiku. Ako podržava, rezultat će obuhvaćati i sve potklase klase Osobe.

[10] Usp. Isto str. 499
[11] Usp. Handbook on ontologies, 2010. str. 501

Kao i u SQL-u, SPARQL upiti imaju strukturu SELECT – FROM – WHERE:

SELECT određuje projekciju: broj i redoslijed dohvaćenih podataka. FROM se koristi za određivanje izvora iz kojeg se podaci dohvaćaju. Ova klauzula nije obavezna; kada nije određena, može se jednostavno pretpostaviti da se ispituje baza znanja određenog sustava. WHERE nameće ograničenja na moguća rješenja odnosno grafički prikaz rezultata.
Na primjer, kako bi pronašli sve adrese e-pošte osoba, može se napisati:

```
SELECT ?x ?y
WHERE
{
?x foaf:mbox ?y .
}
```

Ovdje su? x i? y varijable, a ?x foaf: mbox ?y predstavlja RDF triplet izvor-svojstvo-vrijednost (engl. *resource property-value*).[12]

Većina RDF spremišta (*RDF Triplestore*) koristi jedan od uobičajenih RDF upitnih jezika, poput RQL (*RDF Query Language*), RDQL (*RDF Data Query Language*) ili SPARQL-a. To znači implementirati analizator rečenice (*parser*) koji će analizirati sintaksu upitnog jezika. Potencijalni međukorak za prevođenje raščlanjenog upita u relacijsku jednadžbu je grafikon ili objektni model za dohvaćanje semantike upita. Nakon toga, SQL upit se formira i šalje u bazu podataka. Sintaksa kreiranog SQL upita obično ovisi o temeljnom sadržaju DBMS-a (*Database Management System*). To je razlog zašto je potrebno stvaranje dodatnog srednjeg sloja koji apstrahira stvarni mehanizam za pohranu koji nudi funkcije pohrane i pronalaženja. Primjer za to je sloj pohrane i zaključivanja (SAIL) u *Sesame*. Sloj se može zamijeniti ovisno o korištenom DBMS-u, a može se postaviti i na drugi SAIL kako bi ponudio daljnju funkcionalnost poput predmemoriranja nedavnih rezultata upita. Važan aspekt pristupa podacima je optimizacija upita. To može biti prepušteno sustavu baza podataka, uzimajući u obzir sofisticirane mehanizme procjene i optimizacije suvremenih RDBMS-a. Dakle, upit mora biti preveden na SQL, što potpunije.[13]

[12] Usp. Handbook on ontologies, 2010. str. 502
[13] Usp. Isto str. 496

2.4 RDFS

RDF Schema (RDFS) je univerzalni jezik koji dopušta korisniku opisivanje izvora/resursa putem vlastitog rječnika. RDFS ima nekoliko formata pohrane. Najčešći formati pohrane su Turtle i RDF/XML. RDFS je gradivni blok semantičkog weba (Semantic Web). RDFS udomljava tzv. terminološko znanje ili *'schema znanje'*. RDFS rječnik ne odnosi se na kontrolirane rječnike termina nego na semantički strukturirane rječnike specificirane u samom dokumentu.

RDFS je najjednostavniji ontologijski rječnik, ili *lightweight* ontologija. Temeljni koncepti RDFS-a su klase i svojstva. Klasa je skup izvora koji određuju vrstu izvora. Svojstvo je predikat (glagol) povezan s klasom ili njenom instancom. URI/IRI se tretiraju kao nazivi klasa i nazivi svojstava. RDFS opisuje izvore pomoću svojstava, njihovih vrijednosti i klasa. RDFS predstavlja okvir (*framework*) za opis specifičnih klasa i njihovih svojstava. Primjer klasa su: ljudi, građani, ljubimci, hrana, knjige itd. Klase se mogu sastojati od potklasa (engl. subclasses). Primjeri potklasa za klasu ljudi su: profesori, web dizajneri i sl. U tom slučaju, ljudi su superklasa za klasu profesora. Instanca klase - konkretni entitet koji se javlja kao pripadnik odgovarajuće klase/potklase (npr. Boris Badurina je pripadnik 'potklase' web dizajneri i potklase profesori koja je pak potklasa klase ljudi).

Specijalna svojstva RDF-a i RDFS-a su: rdf:type - povezuje instancu klase i klasu. Piše se i kao malo 'a'. Primjer: oziz:Badurina a oziz:Profesori . rdfs:subClassOf omogućuje hijerarhiju klasa i nasljeđivanje – potklase i superklase. Specijalno svojstvo rdfs:subPropertyOf - omogućuje hijerarhiju svojstava i nasljeđivanje.

2.5 OWL

OWL (*Web Ontology Language*) je ekstenzija RDF *Scheme*. OWL je ontologijski jezik za semantički web s formalno definiranim značenjem. Prvi put objavljen je 2004. Ključni termini OWL-a su: klase, svojstva, pojedinci (*individuals*, u RDF-u: instance), podatkovne vrijednosti (*data values*). Kada je stvoren 2004. godine, dijelio se na: OWL Lite, OWL DL i OWL Full.

2012. je stvoren OWL2 koji se dijelio na OWL DL koji je imao manju izražajnost, ali veću moć zaključivanja te se temeljio na opisnoj logici i OWL Full koji ima veću izražajnost, a manju moć zaključivanja te je kompatibilan s RDF-om. OWL2 dopušta deklaraciju klasa isključivo za neslovne resurse; OWL2 DL klase su instance owl:Class (prije nego rdfs:Class). OWL2 DL svojstva su instance klase owl:ObjectProperty ili klase owl:DatatypeProperty i ne

mogu biti istovremeno u obje klase. U OWL2 DL-u resurs ne može biti klasa, svojstvo i instanca u isto vrijeme. Posljedica toga je da se izgubila kompatibilnost s RDF-om (nije svaki RDF dokument i OWL2 DL dokument) ali se ostvarilo bolje zaključivanje zbog ograničavanja načina izražavanja. Unutar OWL2 DL klase su instance owl:Class umjesto rdfs:Class. U OWL2 članovi klase se nazivaju *individuals* (pojedinci) radije nego instance. OWL2 je u osnovi jezik za opis setova stvari. Ti setovi se nazivaju klase.

OWL2 ima nekoliko sintaksi od *Functional Style Syntax*, preko *OWL/XML* do *Manchester* sintakse. Svaka od navedenih sintaksi ima svoje prednosti i nedostatke. *Functional Style Syntax* je kompaktna i čitljiva sintaksa. OWL/XML je dobra sintaksa za interakciju s XML alatima. *Manchester* sintaksa je stvorena da bude što čitljivija za ljude i zato se koristi u alatima za uređivanje ontologija poput Protégéa.

3. Prikazi alata za izradu ontologija

Alati za izradu ontologija omogućuju inspekciju, pregledavanje, kodificiranje i modifikaciju ontologija, podržavajući na taj način razvoj i održavanje ontologije.[14] Ovo je područje koje se brzo razvija. Alati koji će se uspoređivati u ovom radu su OWLGrEd, Protégé, WebProtégé, i Fluent Editor. U potrazi za alatima koji su se koristili i koji se koriste za izradu ontologija otkriveno je kako su neki od pregledanih alata ili nestabilni ili više nisu dostupni.[15]

3.1. OWLGrEd

3.1.1. Općenito o OWLGrEd-u

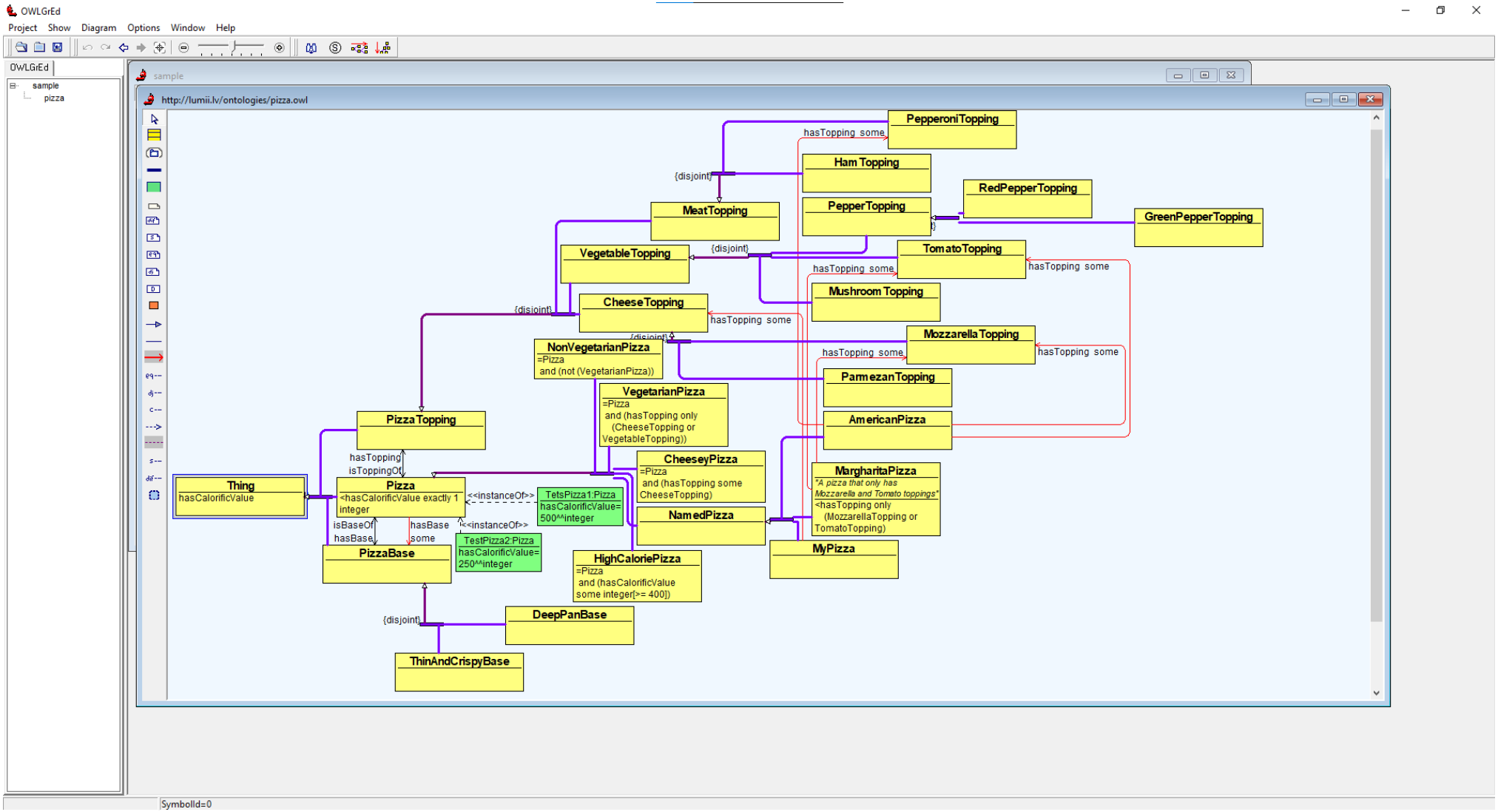
U ovom poglavlju opisuje se implementacija *OWLGrEd Ontology Visualizer* alata za izradu ontologija. OWLGrEd je aplikacija klijent-poslužitelj u kojem klijentska strana prihvaća korisničke zahtjeve i sadrži komponente vizualizacije, dok se većina poslova obavlja na strani poslužitelja. Poslužiteljska strana implementirana je kao kanal za transformaciju podataka i sastoji se od usluge prijenosa datoteka, upravitelja 'kanala' i individualnih modula 'kanala' detaljno opisanih kasnije u radu. Ova modularna arhitektura dopušta lakše ažuriranje aplikacije omogućujući skalabilnost rješenja koja paralelno mogu izvoditi transformacije. Poslužiteljska strana aplikacije radi unutar JVM-a (*Java Virtual Machine*) i napisana je u programskim jezicima Javi i Clojure. Komunikacija između klijenta i poslužitelja koristi se asinkronim HTTP (*Hypertext Transfer Protocol*) zahtjevima s JSON (*JavaScript Object Notation*) formatom. Korisnici mogu, ili učitati i vizualizirati svoju ontologiju; ili učitati primjere za vizualizaciju. Kako bi se započeo postupak, korisnik odabire datoteku ontologije koja će biti vizualizirana. Aplikacija sprema datoteku i omogućuje izvođenje sljedećih koraka - raščlanjivanje ontologije i generiranje grafova. Aplikacija analizira ontologiju pomoću OWL API-ja. Korištenjem uspostavljenog API-ja za raščlanjivanje OWL-a može se izbjeći više posla s više načina za predstavljanje OWL ontologija. Kao rezultat toga korisnici mogu učitati ontologije u bilo kojem formatu sve dok ga prepoznaje OWL API.[16]

3.1.2. Analiza OWLGrEd

Korisničko sučelje je rudimentarno, dok mogućnost uređivanja ontologije unutar vizualnog editora prednost je koja se mora izdvojeno vrednovati. OWLGrEd ima solidnu vizualizaciju ali

[14] Usp. Stojanovic, Liljana; Motik, Boris. Ontology Evolution within Ontology Editors. 2002. str. 1.
[15] Usp. Clunis, Julaine Sashanie. Comparative Survey of Ontology Editors for the Semantic web. 2019. str. 2.
[16] Usp. Liepins Renars; Grasmanis Mikus; Bojars Uldis. OWLGrED Ontology Vizualizer. 2014. str. 39.

za izradu ontologija nedostaju funkcionalnosti kao što su dovršavanje riječi. Zbog toga se preporučuje korištenje OWLGrEd-a samo u slučajevima u kojima postoji potreba za vizualizacijom ontologije.



Slika 2. Primjer uređivanja ontologije u OWLGrEd, verzija 1.6

Slika 2 prikazuje pizza.owl ontologiju unutar OWLGrEd verzije 1.6. U lijevom prozoru vidi se ime datoteke u kojem se nalazi ontologija i ime ontologije koja se trenutno vizualizira. Žuti pravokutnici predstavljaju klase. Klase su prikazane u horizontalnoj hijerarhiji. Promjena orijentacije se nalazi u gornjoj alatnoj traci na desnom kraju trake. Zeleni pravokutnici predstavljaju objekte unutar ontologije. Ljubičaste linije predstavljaju hijerarhijsku poveznicu s drugim klasama. Strelica na kraju ljubičaste linije upućuje na superklasu u odnosu klasa koje su povezane (npr. klasa *Pizza* je superklasa klasi *NamedPizza*). Crvena linija označava svojstva klase. Crvena strelica na kraju linije označava ili upućuje na superklasu u odnosu klasa koje su povezane.

3.2. Fluent Editor

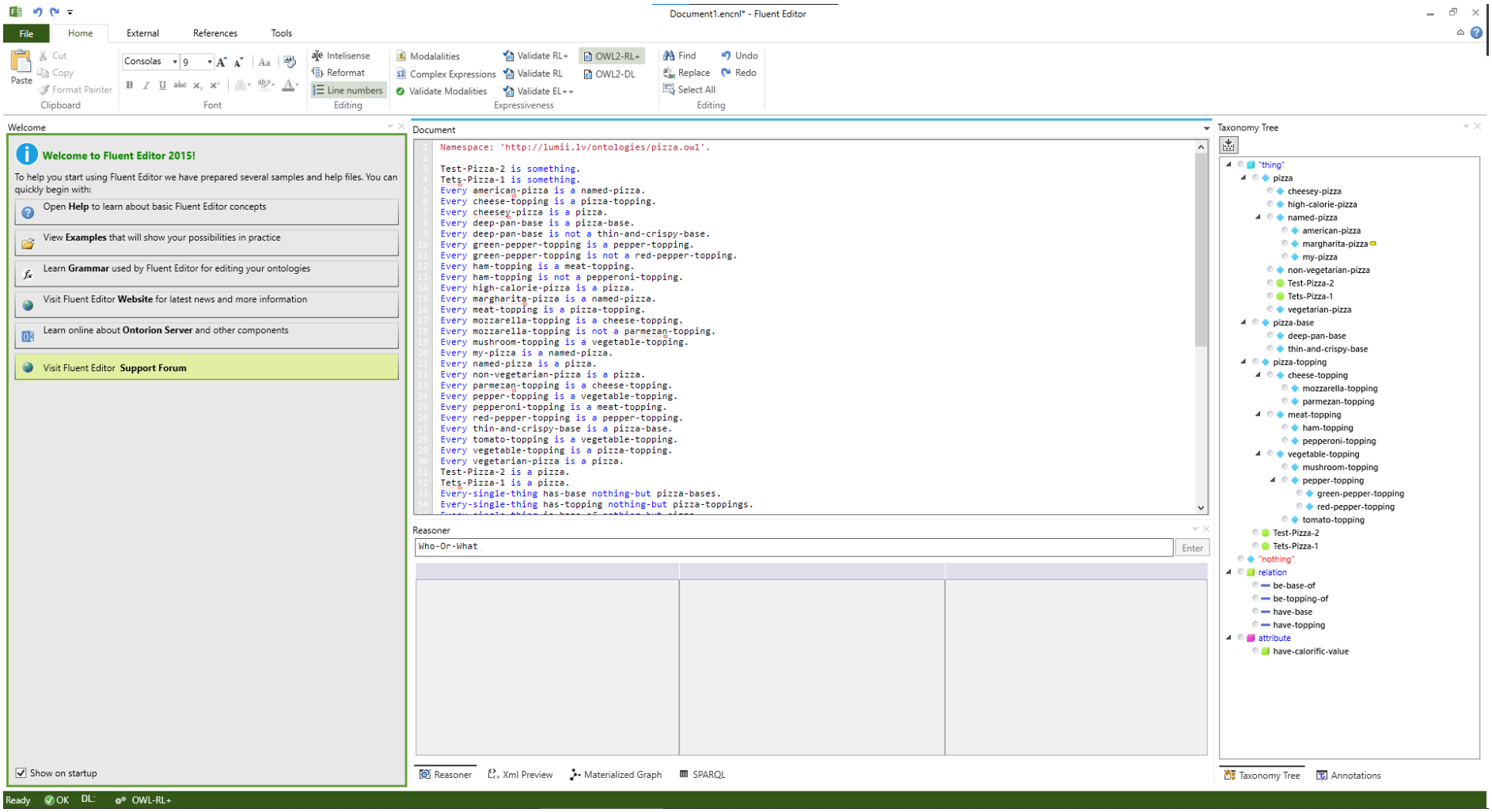
3.2.1. Općenito o Fluent Editoru

Fluent Editor je alat za uređivanje ontologija koji koristi *Ontorion Controlled Natural Language* (OCNL) za dizajn sučelja. Glavne značajke Fluent Editora su: automatsko dovršavanje teksta koje pomaže korisniku u pisanju ispravnih rečenica, mnogo alata za interakciju s komponentama treće strane (dodaci), OCNL sučelje, logičko zaključivanje i materijalizacija, složeni CNL upiti za trenutnu ontologiju, cjelovito upravljanje referencama (uvoz / izvoz / referenciranje OWL / RDF ontologija), grafički prikaz ontologija i interakcija s Ontorion poslužiteljem. S više od 2000 korisnika, Fluent Editor ubrzano postaje alternativa OWL editorima poput Protégéa.[17]

3.2.2. Analiza Fluent Editora

Sučelje kojim se koristi Fluent Editor je pristupačno i lagano za korištenje (Slika 3). Nedostaje funkcionalnost vizualizacije ontologije. Omogućuje interoperabilnost sa Protégé ontologijama. Fluent Editor je sličan po funkcionalnostima Protégéu, ali ih ima znatno manje od Protégéa.

[17] Usp. Seganti A.; Kaplanski Pawel; Zarzycki Pawel. Collaborative editing of Ontologies using Fluent Editor and Ontorion. 2015. str. 1.



Slika 3. Primjer uređivanja ontologije u Fluent Editoru, verzija 3.6.

Na slici 3 je prikazan Fluent Editor u verziji 3.6. U lijevom prozoru je prikazan prozor dobrodošlice s poveznicama koje upućuju korisnika na to kako koristiti aplikaciju, od primjera jednostavnih ontologija do poveznica koje vode do službene stranice Fluent Editora. U središnjem prozoru s nazivom *Document* prikazan je alat za logičko zaključivanje u kojem su u tekstualnom obliku na engleskom jeziku ispisane veze svih klasa u ontologiji (npr., *Every-single-thing that has-topping is a pizza*.).

Slika 4. Fluent 3.6

Na slici 4 prikazano je kako funkcionira prozor za logičko razlučivanje. On se nalazi u ispod prozora *Document*. U prostor za unos teksta se upiše *Who-Or-What has-base ?*. *Who-Or-What* je unaprijed zadani tekst kojim se uvijek započinje upit. *has-base ?* će iz ontologije izvući sve instance potklase i superklase koje imaju svojstvo *has-base.* Prikaz rezultata podijeljen je u tri stupca. U prvom stupcu se nalaze instance, u drugom potklase, a u trećem superklase.

3.3. Protégé

3.3.1. Desktop Protégé

3.3.1.1. Općenito o Desktop Protégéu

Protégé je besplatna platforma otvorenog kôda s rastućom zajednicom korisnika i paketom alata za izgradnju modela domena i aplikacija baza znanja u obliku ontologija. Protégé je editor ontologije i baze znanja koji je stvoren na Sveučilištu Stanford. To je alat koji omogućuje izgradnju ontologija domene znanja određenog područja s prilagođenim obrascima za unos podataka.[18]

Posjeduje čitav niz modula za modeliranje, stvaranje, vizualizaciju i uređivanje ontologija. Također, može se proširiti dodatnom arhitekturom i sučeljem za programiranje (API) temeljenim na Javi za izgradnju alata i aplikacija za baze znanja. Protégé dopušta definiciju klasa, hijerarhije klasa, ograničenja vrijednosti varijabli i različite odnose između klasa kao i svojstava koja se uspostavljaju između instanci klasa. Protégé je besplatno dostupan za preuzimanje s adrese https://protege.stanford.edu/. Sveučilište Stanford objavilo je udžbenik koji pokriva osnove korištenja Protégé-a s OWL dodatkom. Protégé-OWL pruža API za logičko zaključivanje koji može pristupiti vanjskom DIG-kompatibilnom uređaju za logično zaključivanje, omogućujući zaključke o klasama i instancama u ontologiji. Protégé uključuje sučelje za SWRL (*Semantic Web Rule Language*), koji se u modelu odnosno hijerarhiji tehnologija semantičkog weba nalazi na višoj razini od OWL-a, te se SWRL rabi za rješavanje matematičko-logičkih problema, vremensko zaključivanje i dodavanje tzv., 'prolog-tip pravila rasuđivanja' (radi se o programskom jeziku specijaliziranom za logičko programiranje). Značajna prednost Protégé-a u odnosu na ostale alate za izradu ontologija je njegova skalabilnost i proširivost. Protégé omogućuje izgradnju i obradu velikih ontologija na učinkovit način. Zahvaljujući svojoj proširivosti Protégé se može prilagoditi potrebama korisnika.

Najpopularnija vrsta dodataka u Protégéu su dodaci u obliku zasebnih modula koji se u sučelju prikazuju u formi kartica; trenutno dostupni moduli odnosno kartice pružaju korisniku mogućnost za naprednu vizualizaciju, spajanje ontologija, upravljanje verzijama jedne ontologije, logičko zaključivanje, itd. Na primjer, kartice OntoViz i Jambalaya odnose se na različite grafičke prikaze baze znanja. S karticom Jambalaya se omogućuje interaktivna

[18] Usp. Alatrish, Salem, Emhimed. Comparison of Ontology Editors. 2012. str. 5

navigacija, zumiranje pojedinih elemenata u strukturi ontologije i pogled na različit raspored čvorova u grafikonu kako bi se istakle veze između klastera podataka. Protégé također podržava suradničko uređivanje ontologija.[19]

Stanford Sveučilište radi na kontinuiranom poboljšanju Protégéa. Snaga Protégé-a je u tome što istovremeno daje kontinuiranu i sinkroniziranu podršku graditeljima alata, inženjerima znanja i specijalistima domena znanja. To je glavna razlika u odnosu na postojeće alate koji su obično usmjereni na inženjere znanja i zbog toga imaju nedostatak fleksibilnosti za meta-modeliranje. Ova posljednja značajka olakšava prilagodbu Protégéa novim zahtjevima i promjenama u strukturi modela. Prilikom započinjanja projekta izrade ontologije, prvi korak je pronaći odgovarajući alat za izradu ontologije. Ovi alati mogu pomoći korisniku u stjecanju, organizaciji i vizualizaciji znanja iz domene prije i tijekom izgradnje formalne ontologije.[20]

Prije dvadeset godina većina ljudi u području umjetne inteligencije (engl. *artificial intelligence*- AI) osjećala se nelagodno koristeći riječ ontologija u svom svakodnevnom razgovoru. Međutim, svijet se znatno promijenio od tog doba. Sada se ontologije prepoznaju kao važni elementi većine AI tehnologija koje spominju mnogi autori. Dok je na Stanfordu Protégéov tim nastavio izgrađivati ontologije na osnovi strukturiranih podataka iz baza podataka, drugi programeri stvarali su ontologije za širok spektar aplikacija. Ideja semantičkog weba puštala je korijenje, a ontologije su smatrane ključnim za e-trgovinu i e-znanost.

Protégé-II, koji je razvijen u okviru platforme NeXTSTEP, zamijenjen je verzijom sustava koja je radila pod Windowsima. U roku od nekoliko godina Java programski jezik učvrstio se u zajednici softverskog inženjerstva. Nova verzija sustava utemeljenog na Javi, Protégé-2000, oslobodila se od oslanjanja na određenu hardversku platformu, a istovremeno je omogućila stvoriti vrlo fleksibilnu arhitekturu s mogućnostima implementacije različitih dodataka osnovnom softverskom rješenju. S inačicom Protégéa programiranom u Java programskom jeziku ostvarena je implementacija podrške za razvoj ontologija s RDF-om, a ubrzo nakon toga, i s OWL-om. S većim promjenama u dizajnu, Protégé-2000 je uskoro postao Protégé 3, zatim Protégé 4, a danas Protégé 5.

U međuvremenu, sa zanimanjem za ontologije koje je raslo u AI krugovima i šire, i sâm Protégé se ubrzano razvijao. Softver koji se uglavnom smatrao kao infrastruktura koja je omogućavala

[19] Usp. Alatrish, Salem, Emhimed. Comparison of Ontology Editors. 2012. str. 20-21
[20] Usp. Kapoor, Bhaskar; Sharma, Savita. Comparative Study Ontology Building Tools for Semantic Web Aplications. 2010. str. 6.

istraživanje inteligentnih sustava uspostavljenih na Sveučilištu Stanford - dosezao je fazu u kojoj će uskoro postati "*proizvodom*".

Autor „*The Protégé Project*“ Mark A. Musen u nastavku opisuje osobno iskustvo i percepciju Protégé tima u javnosti kako se popularnost njihovog projekta širila i kakav je utjecaj imala na tim. Troje njegovih kolega i on 1996. godine prisustvovali su jednoj konferenciji u Europi, a njihov su skup u šali nazvali "Prvi sastanak grupe korisnika Protégé-a". No, ubrzo je zajednica korisnika Protégéa zahtijevala održavanje Protégé radionice. Tako je 1998. na Sveučilištu Stanford održana prva Protégé radionica koju je pohađalo oko 50 ljudi. Oko 100 ljudi 2000. godine sudjelovalo je na radionici Protégé-a na Sveučilištu u Manchesteru. Više od 200 ljudi pojavilo se na sličnom skupu u Nacionalnom institutu za zdravstvo Sjedinjenih Država 2002. godine. Ove su Protégé radionice ujedno bile i uzbudljivi sastanci za cijeli njihov tim: dobili su izuzetno vrijedne povratne informacije o *softveru*[21], upoznali su ljude koji su se redovito dopisivali s njima na *mailing* listama povezanim sa Protege-om koje su u međuvremenu otvorili te se informirali o lokalnim aplikacijama koje su članovi zajednice stvarali pomoću njihova programa.

Percepcija Protégéa radikalno se mijenjala. Softver se više nije smatrao samo alatom za izgradnju sustava temeljenih na znanju, nego alatom za izgradnju upravo računalnih ontologija. Istodobni rast Protégé zajednice i nagla pojava mnoštva dodataka programu kojima su doprinijeli korisnici izvan Sveučilišta Stanford nastavili su mijenjati prirodu istraživačke aktivnosti unutar Protégé zajednice. Svaka promjena Protégé baze kôda mogla bi iznenada utjecati na rad tisuća drugih ljudi - a tisuće drugih ljudi sada su imale vlastite ideje u kojem smjeru bi rad na projektu Protégé trebao krenuti. Sveukupno, projekt je imao ogromnu korist od širenja korisničke baze, a priljev novih ideja i novih dodataka iz dana u dan potvrđivao je korisnost aplikacije.

Iako su se komercijalni alati za izradu ontologija s podrškom za OWL počeli pojavljivati na tržištu krajem prošlog desetljeća, nije zabilježen pad entuzijazma za Protégé i njegove dodatke. Kao akademska skupina, Protégé zajednica se ne vidi kao konkurencija takvim komercijalnim alatima, a „tržišni udio“ je nebitan pojam za Stanford tim i njihove donatore. Jasno je, međutim, da korisnici cijene što je Protégé objavljen i dostupan pod licencom otvorenog kôda, kao i predanost Stanford tima potpunoj provedbi preporuka W3 konzorcija. Pored toga, korisnici ističu i ogromnu knjižnicu Protégé dodataka koju su sami razvili i održavaju, kao i mogućnost

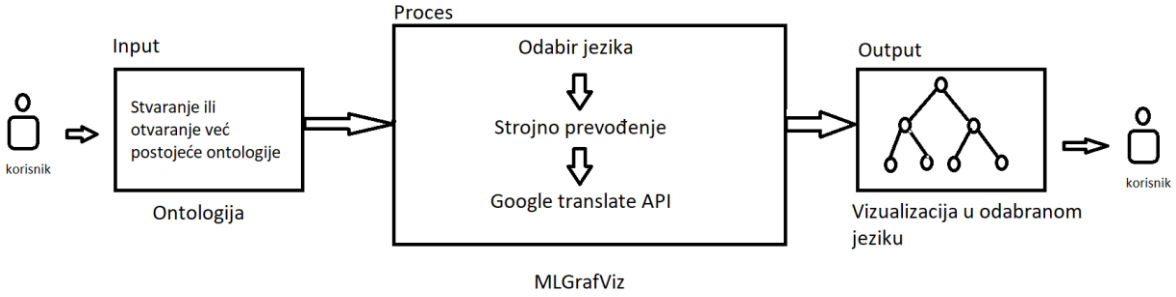
[21] Usp. Musen, A., Mark. The Protégé Project: A Look Back and a Look Forward. 2015.

podrške koju međusobno pružaju članovi zajednice putem internetskih medija pomažući jedni drugima u rješavanju određenih problema, bilo u korištenju softvera ili u modeliranju određenih entiteta. Stanford tim tako nije stvorio zajednicu Protégéa. Socijalni inženjering nikada nije bio „rezultat“ njihovog tima. Korisnička zajednica se pojavila i izrasla temeljem korištenja alata. Projekt Protégé danas broji tisuće korisnika koji su se okupili na mreži kako bi učili jedni od drugih i kako bi pomogli da Protégé učine boljim sustavom.[22]

3.3.1.2. Analiza Desktop Protégéa

Prvi korak u korištenju desktop Protégéa je njegovo preuzimanje s mrežne stranice protege.stanford.edu i instalacija. Protégé za svoj rad zahtjeva *Java Runtime Environment* (JRE). Nakon toga se aplikaciju može pokrenuti (vidi *Sliku 5*). Korisničko sučelje je potpuno izmjenjivo, od prozora do tabova te promjena boje sučelja po operacijskom sustavu. Također je moguće preuzeti dodatke koji proširuju mogućnosti za izradu, održavanje i vizualizaciju ontologija. Za većinu korisnika, Protégé pruža grafičko sučelje koje je prema korisnicima 'prijateljski' usmjereno (engl. *user-friendly),* što znači da je jednostavno za korištenje.[23]

Dodatak MLGrafViz omogućuje korisniku da ontologiju vizualizira na sedam načina: abecedna mreža (*grid alphabetical*), vodoravno stablo (*tree horizontal*), okomito stablo (*tree vertical*), opruga (*spring*), radijaln o(*radial*), usmjereno vodoravno (*directed horizontal*) i usmjereno okomito(*directed vertical*). Korisnik može kreirati jezgru ontologije na bilo kojem prirodnom jeziku koji će biti preveden i vizualiziran na 135 različitih prirodnih jezika.[24]

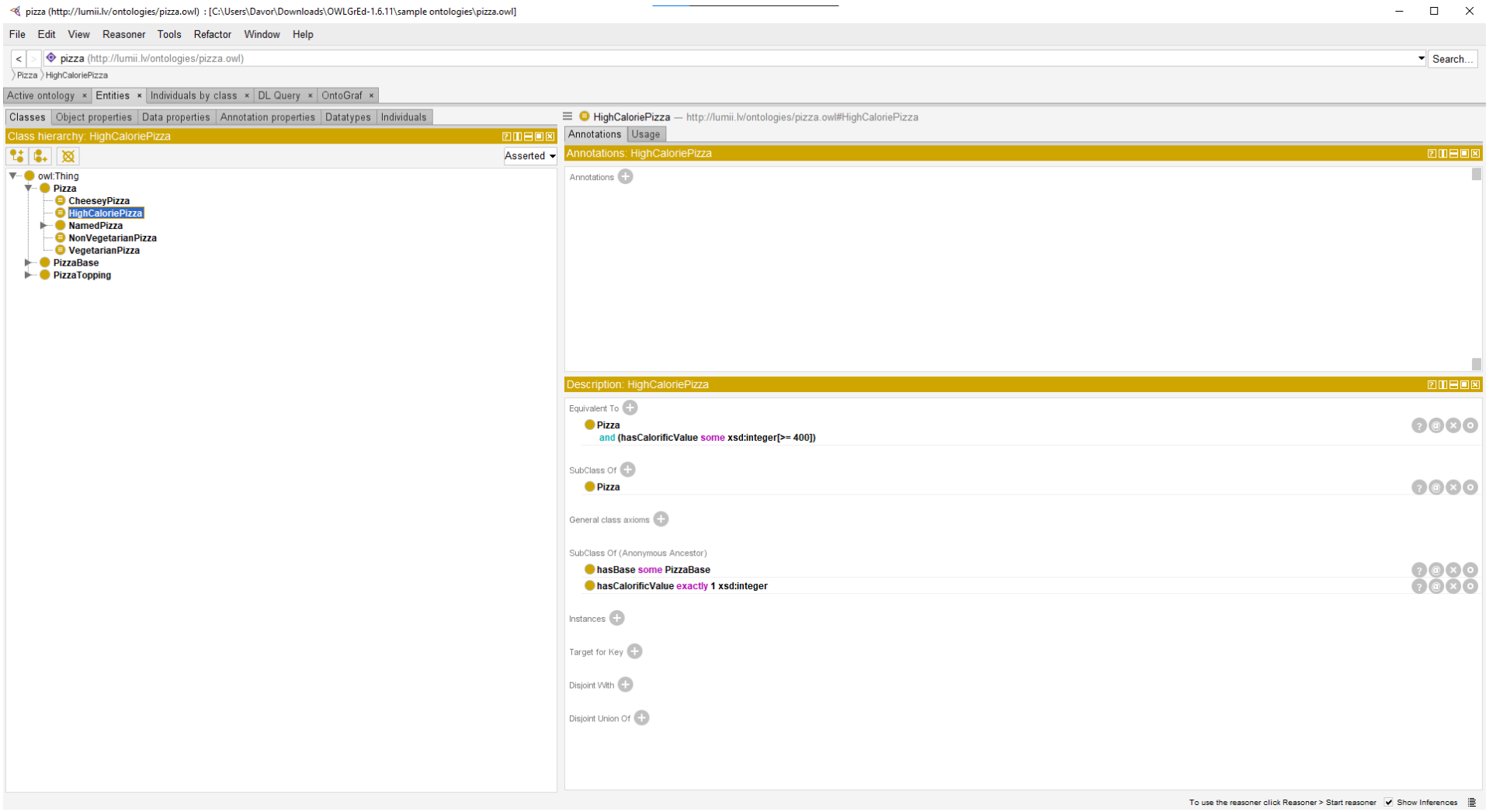


Slika 4. MLGrafViz funkcionalnost

Na slici 4. prikazan je proces kojim MLGrafViz prevodi ontologije. Proces počinje s korisnikom koji stvara ili otvara već postojeću ontologiju. Nakon toga se otvara *tab* MLGrafViza te se vrši odabir jezika trenutne ontologije i jezika u koji se želi prevesti

[22] Usp. Musen, A., Mark. The Protégé Project: A Look Back and a Look Forward. 2015.
[23] Usp. Alatrish, Salem, Emhimed. Comparison of Ontology Editors. 2012. str. 23.
[24] Usp. Florrence, Merlin. MLGrafViz: multilingual ontology visualization plug-in for Protégé. 2021. str. 45

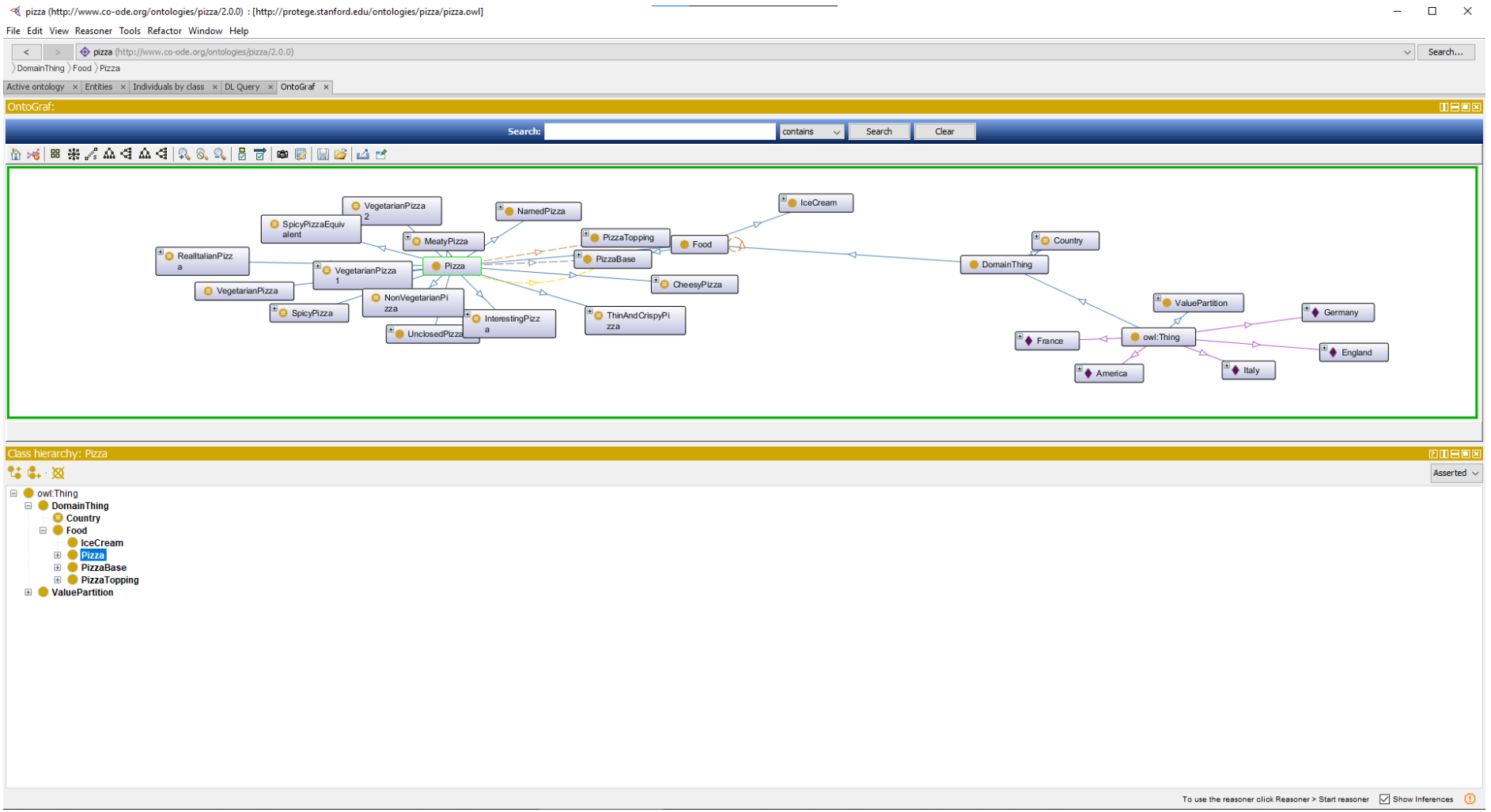
ontologija. Nakon toga ontologija se strojno prevodi putem Google *translate* API-ja. Zadnji korak je vizualizacija ontologije u odabranom jeziku.



Slika 5. Protégé 5.5

Na slici 5. prikazano je sučelje programa Protégé verzija 5.5. u koji je učitana pizza.owl[25] ontologija. Otvorena kartica klasa (*Classes*) omogućuje prikazivanje prozora hijerarhije klasa i opisa klasa. Osim navedenih prozora prikazuje se i prozor anotacija. U prozoru hijerarhije klasa može se pregledati taksonomija ontologije a odabirom određene klase mogu se vidjeti detaljnije informacije vezane uz tu klasu u prozoru na desnoj strani sučelja.

[25] Datotteka pizza.owl nalazi se na adresi: https://protege.stanford.edu/ontologies/pizza/pizza.owl.



Slika 6. OntoGraf modul/kartica programa Protégé 5.5

Slika 6. prikazuje OntoGraf modul/karticu za vizualizaciju ontologije. Na slici se vizualizira ontologija pizza.owl. Klikom na pravokutnik s imenom klase postepeno se vizualizira ontologija te se mogu uočiti veze između klasa. Ponovnim klikom na već otvorenu klasu moguće je istu zatvoriti. Linije određuju veze između klasa a smjer strijelice određuje potklasu.

3.3.2. WebProtégé

3.3.2.1. Općenito o WebProtégé

WebProtégé se može usporediti u osnovi s Google Docs alatom za uređivanje ontologija. Korisnici pristupaju "projektima" koji predstavljaju zbirke OWL ontologija s dodanom vrijednošću - poviješću izmjena i komentara korisnika. WebProtégé pruža unaprijed zadano korisničko sučelje koje podržava jednostavno uređivanje ontologija. No, WebProtégé može se podesiti i za uređivanje ontologija koje zahtijevaju potpuno izražene aksiome OWL 2 (OWL 2 *axioms*) i izraze klase (engl. *class expressions*). WebProtégé omogućuje uređivanje ontologije tako da svi korisnici vide trenutno stanje ontologije. Moguće je postaviti projekt tako da korisnici mogu i samo pogledati ili komentirati projekt, u slučajevima u kojima postoji skupina urednika ontologije i veća skupina komentatora i recenzenata. Korisnici WebProtégéa mogu lako povezati entitete s popularnim 'bazama znanja' kao što su Schema.org, Wikidata i DBpedia. Na primjer, povezati se s entitetom Skup podataka u Schema.org, korisnik može učiniti tako da upiše schema:Dataset pri stvaranju nove klase i WebProtégé će stvoriti entitet s njegovim IRI-jem https://schema.org/Dataset; slično se može napraviti i za entitete Wikidata i DBpedije koji koriste nazive prefiksa wikidata i dbpedia.[26] WebProtégé je jednostavan editor ontologija osmišljen da se rabi u mrežnom okruženju. Cilj u izradi ovog alata nije bio ponuditi još jedan novi editor ontologije, nego zadovoljavanje potrebe za online editorom ontologija. Cilj je bilo pružiti alat za ontologiju kojeg mogu koristiti korisnici iz šireg spektra, u rasponu od stručnjaka za ontologiju do eksperata područja znanja. Dakle, mogućnost prilagodbe korisničkog sučelje za korisnike s različitim razinama stručnosti bilo je od najveće važnosti kod odluke o dizajnu. Nadalje, većina projekata koristi pristupe zasnovane na zajednici za razvijanje ontologija. Ponovna upotreba ontologije također je česta u naprednijim domenama znanja, poput biomedicine. WebProtégé podržava ponovnu upotrebu i međusobno povezivanje biomedicinskih ontologija. Stvaranje otvorene, proširive i fleksibilne infrastrukture koja može

[26] Usp. Horridge, Matthew; Goncalves, S., Rafael; Nyulas, I., Csongor. WebProtégé: A Cloud-Based Ontology Editor.2019. str. 686

biti lako prilagodljiva potrebama različitih projekata bio je također glavni prioritet u dizajniranju WebProtégéa.[27]

Uz funkcionalnost zajedničkog uređivanja ontologije, WebProtégé također omogućuje korisnicima ostavljanje poruka u projektu i prijavu problema u obliku „*threaded*“ komentara. Komentari se nalaze izvan ontologije, ali se odnose na određene entitete u ontologiji. Formatirani su u sintaksi *Markdown* i mogu sadržavati veze s bilo kojim entitetom u ontologiji, a mogu i spominjati korisnike koji su prijavljeni na neku drugu platformu poput GitHub-a. Broj niti (engl. *threads*) se također može prikazati pored imena entiteta. Kada se objave komentari, sudionici projekta primaju obavijest o komentaru putem e-pošte. E-pošta sadrži tijelo komentara i poveznicu koja korisnike vodi izravno na komentar u prozoru preglednika.

WebProtégé također omogućuje integraciju sa *Slackom* (platformom za komunikaciju) i omogućuje povezivanje *Slack webhooka* (način slanja poruka u stvarnom vremenu) s projektom tako da se komentari objavljuju na *Slack* kanalu za vanjske obavijesti. Korisnici *Slack*a mogu kliknuti na veze u svojoj *Slack* aplikaciji kako bi pogledali komentar u WebProtégéu. Osim komentara, WebProtégé također omogućuje entitetima da budu "označeni" oznakama koje se mogu definirati za projekt. Oznake i komentari obično se grupiraju zajedno. Entiteti s određenim oznakama mogu se također pretraživati. Entiteti se mogu automatski označiti prema pravilima koje korisnici izrade koristeći iste kriterije koji su im dostupni za pretraživanje. Na primjer, entiteti koji nemaju vrijednost za svojstvo oznake rdfs: label će biti automatski označeni oznakom „*nedostaje oznaka label*“.[28] Jedna od ključnih značajki WebProtégé-a jest da se sve promjene u ontologiji prate i grupiraju u revizije.

Revizije se temelje na atomskim operacijama unutar korisničkog sučelja, i moguće je sustav vratiti u stanje prije revidiranja. Promjene mogu biti pregledane ili filtrirane po entitetima tako da obuhvaćaju cijelu povijest entiteta. Oznake za revizije automatski se generiraju na temelju promjene koje su se dogodile. U OWL 2 uređivačkom sučelju moguće je grupirati skup promjena u tzv. *'commit'*, i s time mu se onda može dodijeliti tzv. *commit* poruka. WebProtégé dopušta preuzimanje skupa ontologija u danoj reviziji.[29] WebProtégé pruža mrežni *frontend* (za svakog korisnika) koji se povezuje s WebProtégé *backendom*. *Backend* pokreće Tomcat

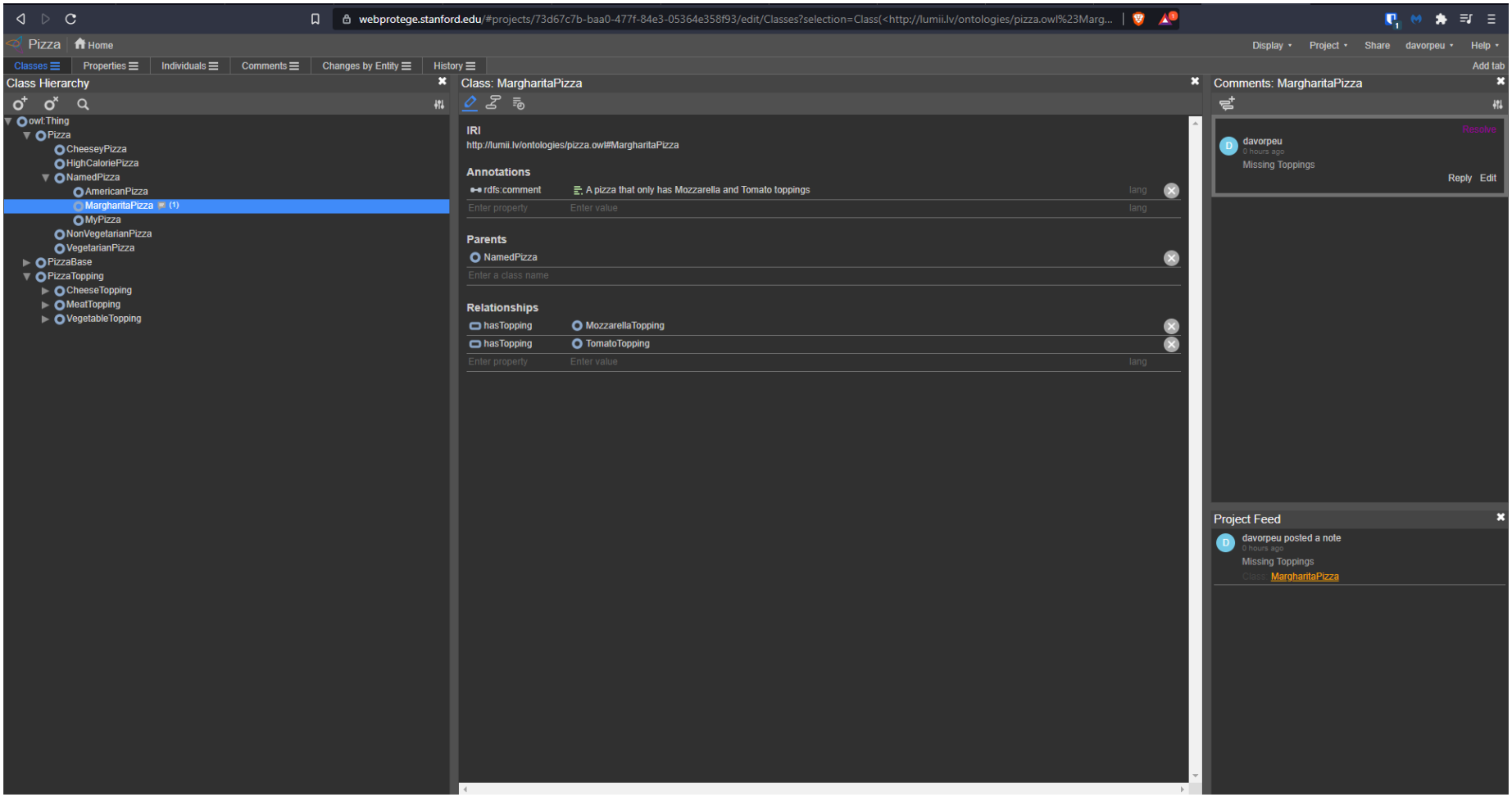
[27] Usp. Tudorache, Tania; Nyulas, Csongor; Noy, Natalya. WebProtégé: A Collaborative Ontology Editor and Knowledge Acquisition Tool for the Web. 2011. str. 2-3
[28] Usp. Horridge, Matthew; Goncalves, S., Rafael; Nyulas, I., Csongor. WebProtégé: A Cloud-Based Ontology Editor.2019. str. 686-688.
[29] Usp. Horridge, Matthew; Goncalves, S., Rafael; Nyulas, I., Csongor. WebProtégé: A Cloud-Based Ontology Editor.2019. str. 686.

mrežni poslužitelj i odgovoran je za rukovanje korisničkim zahtjevima. WebProtégé podržava zajedničke značajke kao što je istodobno i višestruko uređivanje ontologije. Klijenti na internetu prate promjene u ontologiji. WebProtégé *backend* se povezuje s Protégé serverom koji implementira pristup središnjoj verziji ontologije u kojoj se pohranjuju sve funkcionalnosti ontologije (npr. uređivanje, logično zaključivanje itd.). Međutim, upravljanje ulogama je nedovoljno razvijeno (tj. svi korisnici imaju potpuno ista prava po pitanju uređivanja ontologije; izmjene koje je napravio jedan korisnik, odmah su vidljive i svim ostalim korisnicima). Logičko zaključivanje je podržano, ali samo na središnjoj verziji ontologije koja je pohranjena na Protégé poslužitelju. Konfiguracija i prilagodba korisničkog sučelja po ontologiji, a i po korisniku je poželjna značajka WebProtégé-a koju drugi sustavi ne podržavaju.[30]

3.3.2.2 Analiza WebProtégé

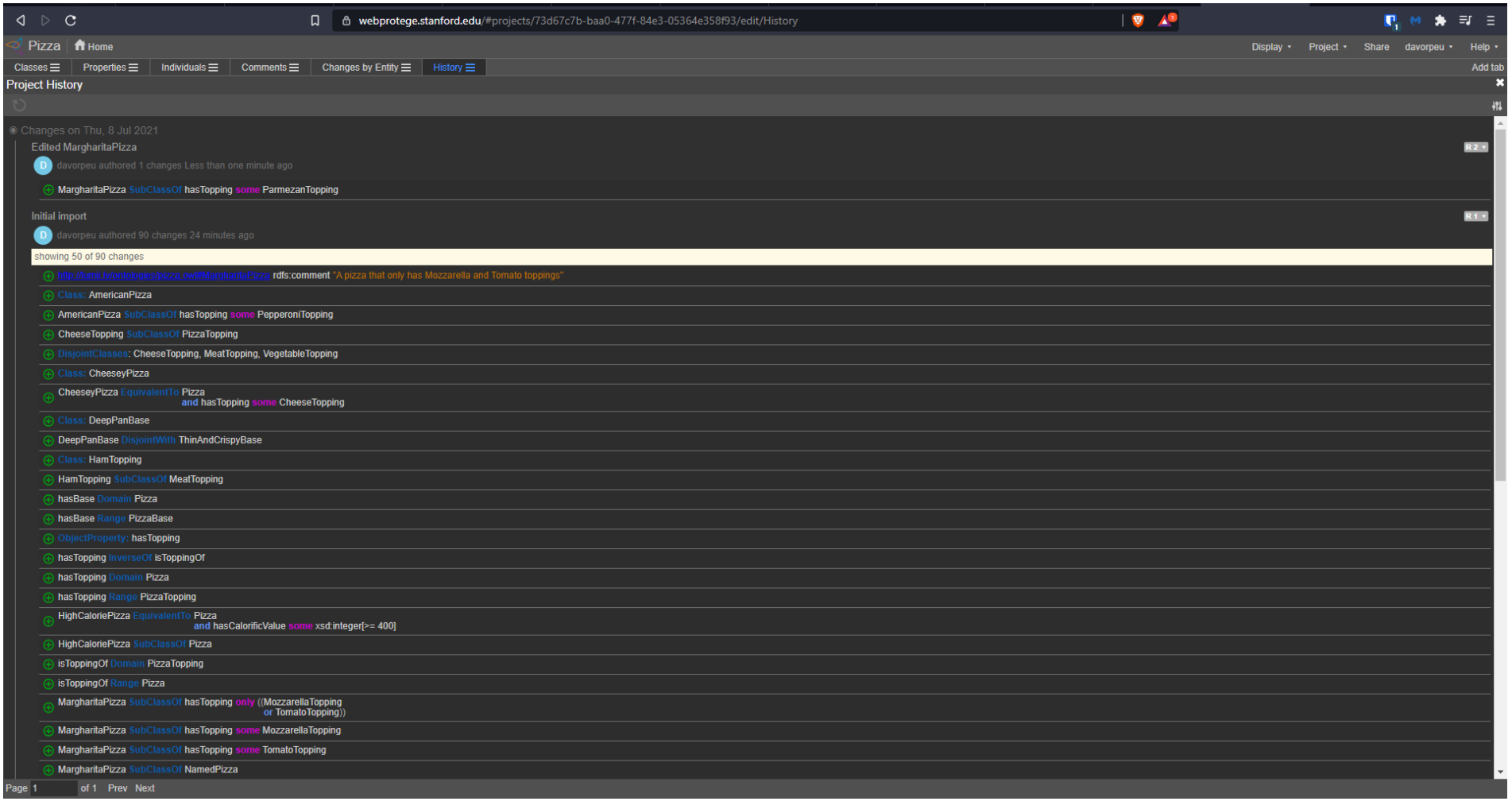
Da bi započeli koristiti WebProtégé korisnici se najprije moraju prijaviti sa svojim korisničkim računom na mrežnoj stranici https://webprotege.stanford.edu/. Nakon toga, mogu odabrati novi ili postojeći projekt te pristupiti jednostavnom i pristupačnom sučelju unutar preglednika. Korisničko sučelje je prilagodljivo, a zbog mogućnosti zajedničkog uređivanja ontologije posjeduje mogućnost prikazivanja vijesti s najnovijim izmjenama i vremenom kada su te izmjene napravljene. Također u sučelju se nalazi prozor u kojem se prikazuje povijest svih izmjena ontologije, od samog nastanka. Sa lakoćom se dodaju, mijenjaju i brišu ovlasti uređivanja i pregledavanja ontologije u postavkama za dijeljenje. Pored toga, sučelje omogućuje automatsko dovršavanje teksta što znatno olakšava izradu ontologije.

[30] Usp. Preventis, Alexandros; Petrakis, Euripides. CLONE: Collaborative Ontology Editor as a Service in the Cloud. 2021. str. 3.



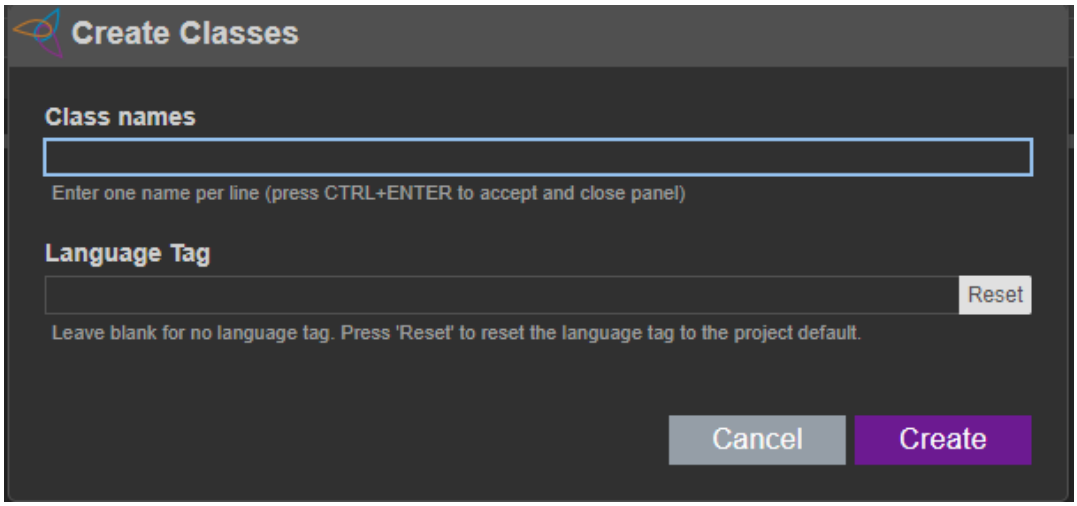
Slika 7. WebProtégé

Slika 7. prikazuje sučelje WebProtégéa s učitanom ontologijom pizza.owl. U desnom prozoru sučelja prikazuje se hijerarhija klasa kao i u sučelju *desktop* Protégéa. U sredini sučelja su prikazani detalji odabrane klase *MargharitaPizza.* U desnom prozoru sučelja mogu se vidjeti komentari vezani uz odabranu klasu. Na komentare se može odgovoriti i tako nastaviti raspravu u vezi određene klase ili se komentar može razriješiti klikom na riječ *resolve* iznad komentara.



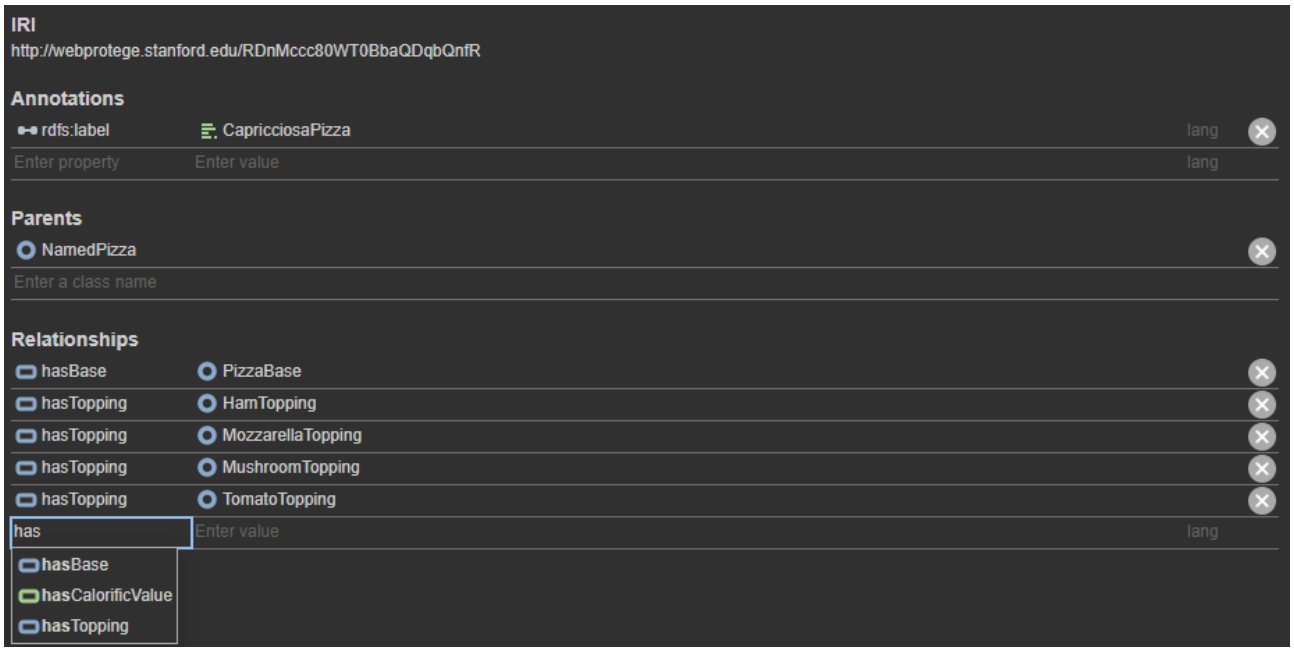
Slika 8. WebProtégé - povijest izmjena u razvoju ontologije.

Na slici 8. prikazan je prozor u kojem se nalazi povijest izmjena u razvoju ontologije. Na slici se može uočiti da je izvršen inicijalni uvoz ontologije a iznad toga uređivanje klase *MargharitaPizza* dodavanjem dodatnog svojstva ParmezanTopping.



Slika 9. WebProtégé – kreiranje klasa.

Na slici 9. prikazano je kako izgleda sučelje kada se stvara klasa. Nakon što se klikne ikona u obliku slova O sa znakom + pojavi se prozor koji je prikazan na slici. Klikom na gumb *create* stvara se klasa u ontologiji.



Slika 10. WebProtégé – informacije o novostvorenoj klasi.

Na slici 10 prikazano je sučelje s detaljima novostvorene klase. IRI je automatski stvoren i dodijeljen klasi. Ispod *Annotations* mogu se dodati anotacije pisanjem teksta unutar polja za

unos teksta '*Enter property' i 'Enter value'*. Ispod *Parents* mogu se dodati nadklase kao npr. Za 'CapricciosaPizza' nadklasu *'NamedPizza'*. U prostoru za unos teksta ispod *Relationships* mogu se upisati svojstva. Pritom, WebProtégé nudi funkcionalnost automatskog dopunjavanja teksta. Na ovaj način vrlo brzo se mogu stvoriti i uređivati klase.

5. Zaključak

Svrha ovog rada bila je prikaz alata za izradu ontologija koji su trenutno dostupni za korištenje. Tijekom godina, kako su se razvijali novi alati za izradu ontologija, samo je Protégé ostao dulje od 20 godina na tržištu. Ovaj rad trebao je olakšati izbor odgovarajućeg alata kako za početnike tako i za profesionalne kreatore i urednike ontologija.

U radu je bio dan prikaz sljedećih alata: OWLGrEd, Fluent Editor, Protégé i WebProtégé. Unutar alata OWLGrEd ontologije se izrađuju *drag and drop* funkcionalnošću putem koje stvaramo, povezujemo i opisujemo entitete ontologije. Sučelje ovog alata je rudimentarno, a vizualizacija je dobro osmišljena tako da ovaj alat mogu koristiti i početnici. Za razliku od OWLGrEda, Fluent Editor koristi iznimno čitljiv način zapisivanja klasa. Također, nudi i interoperabilnost s Protégéom preko funkcionalnosti izvoza (*export*) ontologije u Protégé. Od svih nabrojanih alata, Protégé editor za ontologije ima najdužu povijest i najduži period aktivnog korištenja. To ga čini vrlo stabilnim i sigurnim izborom kao alata za izradu ontologija. Uz to, posjeduje velik broj dodataka u formi modula koji se naknadno mogu priključiti unutar Protégé-a koji isti dodatno čine korisnijim i pristupačnijim za korištenje. Od početnika do profesionalnog urednika ontologija Protégé i dalje uživa veliku popularnost kao alat za izradu ontologija.

Od predstavljenih alata za početnike u izradi ontologije najprihvatljivije rješenje je WebProtégé. Za stručnjake ontologija i eksperata područja znanja, već sam Protégé je odličan izbor za alat za izradu ontologija, no za zajednicu korisnika koja radi na zajedničkoj ontologiji - WebProtégé je svakako prvi izbor, zbog svoje pristupačnosti i *user-friendly* sučelja. Protégé ima dugi razvojni period koji mu je omogućio da se vremenom poboljšava i mijenja u skladu s novim tehnološkim smjerovima. K tomu, WebProtégé i Fluent su međusobno interoperabilni odnosno posjeduju mogućnost zajedničkog uređivanja ontologija. WebProtégé ipak nudi jednostavniji pristup korisniku koji samo treba dobiti ovlasti i poveznicu od glavnog urednika programa.

Područje daljnjeg istraživanja ogleda se u polju mogućnosti zajedničkog uređivanja ontologija. Zajedničko uređivanje je relativno nova mogućnost u svijetu IT-a i za dobrobit znanstvene zajednice trebalo bi se dodatno istražiti i implementirati kroz nove projekte.

Literatura

1. Alatrish, Salem, Emhimed. Comparison of Ontology Editors. 2012. URL: http://joc.raf.edu.rs/images/vol4/poredjenje-ontologijskih-editora.pdf (2021-5-22)
2. Berners-Lee, T., Hendler, J., & Lassila, O. The semantic web. // *Scientific American*, 2001, *284*(5), 34-43.
3. Clunis, Julaine Sashanie. Comparative Survey of Ontology Editors for the Semantic web. 2019. URL: https://www.ideals.illinois.edu/handle/2142/103300 (2021-5-22)
4. Dolzhenkov, Valeriy Nikolaevich;Maltzagov, Daudovich Isa;Makarova, Igorevna, Aleksandra. Software Tools for Ontology Development.2020. URL: http://www.warse.org/IJATCSE/static/pdf/file/ijatcse05922020.pdf (2021-5-22)
5. Florrence, Merlin. MLGrafViz: multilingual ontology visualization plug-in for Protégé. 2021. URL: http://www.iaesprime.com/index.php/csit/article/view/87 (2021-5-22)
6. Giri, Kaushal. Role of Ontology in Semantic Web. 2011. URL: https://www.semanticscholar.org/paper/Role-of-Ontology-in-Semantic-web-Giri/507510e87374837d25cad7ad56242c7da2632818?p2df (2021-5-22)
7. Gruber, T.R. A Translation Approach to Portable Ontology Specifications. 2007. URL: http://tomgruber.org/writing/ontolingua-kaj-1993.pdf (2021-6-7)
8. Handbook on ontologies, 2010. URL: https://kask.eti.pg.gda.pl/redmine/projects/sova/repository/revisions/5378040326bc499e118636a1d25ad667285e005c/entry/Praca_dyplomowa/materialy/handbook%20on%20ontologies%202nd%20edition.pdf (2021-5-22)
9. Horridge, Matthew; Goncalves, S., Rafael; Nyulas, I., Csongor. WebProtégé: A Cloud-Based Ontology Editor.2019. URL: https://arxiv.org/ftp/arxiv/papers/1902/1902.08251.pdf (2021-5-22)
10. Kapoor, Bhaskar; Sharma, Savita. Comparative Study Ontology Building Tools for Semantic Web Aplications. 2010. URL: https://core.ac.uk/download/pdf/205969664.pdf (2021-5-22)
11. Liepins Renars; Grasmanis Mikus; Bojars Uldis. OWLGrED Ontology Vizualizer. 2014. URL: http://owlgred.lumii.lv/downloads/publications/OWLGrEd_Ontology_Visualizer.pdf (2021-5-22)
12. Musen, A., Mark. The Protégé Project: A Look Back and a Look Forward. 2015. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4883684/ (2021-5-22)

13. Preventis, Alexandros; Petrakis, Euripides. CLONE: Collaborative Ontology Editor as a Service in the Cloud. 2021. URL: https://www.sciencedirect.com/science/article/pii/S1877050921007791 (2021-5-22)
14. Sari, Fitri, Riri. Implementation of web Ontology and Semantic Application for Electronic Journal Citation System. 2010. URL: http://www.jetwi.us/uploadfile/2014/1226/20141226042002516.pdf (2021-5-22)
15. Seganti A.; Kaplanski Pawel; Zarzycki Pawel. Collaborative editing of Ontologies using Fluent Editor and Ontorion. 2015. URL: https://cgi.csc.liv.ac.uk/~valli/OWLED2015/OWLED_2015_paper_6.pdf (2021-5-22)
16. Stojanovic, Liljana; Motik, Boris. Ontology Evolution within Ontology Editors. 2002. URL: http://ceur-ws.org/Vol-62/EON2002_Stojanovic.pdf (2021-5-22)
17. Tudorache, Tania; Nyulas, Csongor; Noy, Natalya. WebProtégé: A Collaborative Ontology Editor and Knowledge Acquisition Tool for the Web. 2011. URL: http://www.semantic-web-journal.net/sites/default/files/swj210_1.pdf (2021-5-22)