

Ontologija pojmova vezanih za COVID-19

Jeličić, Luka

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:195:801039>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-07-12**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)



Sveučilište u Rijeci – Odjel za informatiku

Informacijski i komunikacijski sustavi

Luka Jeličić

Ontologija pojmova vezanih uz COVID-19

Diplomski rad

Mentor: izv.prof.dr.sc Ana Meštrović

Rijeka, srpanj 2021.

Contents

1. Uvod	3
2. Ontologije	5
2.1. Definicija Ontologije	5
2.2. Vrste ontologije	6
2.3. Primjeri ontologije.....	10
2.4. Primjene ontologije	11
3. Alat Protégé.....	13
3.1. Arhitektura i mogućnosti Protégé-a	13
3.2. Aplikacija i programsko sučelje	13
3.3. Paralelna obrada više verzija projekta	16
3.4. Vizualizacija	16
4. Razvoj ontologije vezane uz COVID-19.....	18
4.1. Procesiranje i priprema podataka za izradu ontologije.....	18
4.2. Konceptualni prikaz ontologije i objašnjenje prikaza.....	18
4.3. Prikaz ontologije u alatu Protégé	24
5. Zaključak.....	30
Bibliography.....	32

1. Uvod

Cilj ovog diplomskog rada biti će pregled i opis odnosa (relacija) i pojmova podataka vezanih uz pandemiju SARS-CoV-2 virusa. Podaci su prikupljeni sa različitih internetskih izvora na dnevnoj bazi te će biti grupirani po pojmovima koji su bili spominjani u medijima vezano za COVID-19. Prikupljeni podaci biti će prikazani u obliku ontologije te će biti objašnjeno što su ontologije. Kako bi se ontologije mogle dobro definirati potrebno je imati sustav koji omogućuje jednostavnu implementaciju ontologija. U prirodnim znanostima ontologije su jedan od načina prikaza znanja. U medicini svaka nova činjenica dobivena istraživanjem mora biti dobro dokumentirana i stavljena u kontekst i povezana s ostalim činjenicama. Problem bi nastao kada ne bi postojali uređeni sustavi prema kojima je svaki lijek ili tretman definiran za točno određenu bolest. Svako novo medicinsko istraživanje može dati nove podatke o bolestima ili lijekovima koje treba povezati sa već postojećim podacima u smislenu cijelinu. Velika većina epidemija i pandemija bolesti u 21. stoljeću imaju svoje ontologije. Većina ontologija napravljenih za različite pandemije se sprema u bazu Nacionalnog Centra za Biomedicinske Ontologije. [1] Sustav koji se koristi za definiranje ontologija je protégé. Protégé sustav služi za hijerarhijski prikaz pojmova te konceptualnih modela različitih sustava i ontologija. Ukratko, ontologije su formalni eksplicitni prikaz znanja u različitim domenama i poveznica između tih domena. Modeliranje znanja je proces kreiranja modela znanja kojeg računalo može iskoristiti za izgradnju različitih objekata ili proizvoda poput proizvoda baziranih na umjetnoj inteligenciji. Kod podataka su definirane klase koje sam koristio za izgradnju konceptualnog modela i ontologije COVID pandemije. Nadalje, događa se problem da različiti znanstvenici sa različitih strana svijeta imaju različite terminologije te se događa neusklađenost terminologija u znanstvenoj zajednici. Primjerice, terminologija u SAD-u se razlikuje od terminologije u Europi ili Aziji jer su aspekti pandemije u različitim dijelovima svijeta različiti i znanstvena zajednica u različitim dijelovima svijeta na probleme oko pandemije gleda različito. U današnjem svijetu pojavljuje se problem infodemije koji polako postaje veći problem i od same pandemije zbog konstantnog priljeva velike količine podataka koje je nemoguće procesirati te odrediti koje su informacije točne, a koje ne. Određena skupina ljudi se pokušavaju boriti protiv infodemije na način da traže informacije iz više izvora te pokušavaju pronaći neku provjerenu informaciju. Glavni pokretač infodemije je senzacionalizam u masovnim medijima i neusklađenost u informiranju putem medija. No, senzacionalizam nije jedini način na koji se događa infodemija. Dovoljno je prenjeti vijest na krivi način da bi se vijest proširila i stvorila efekt infodemije. Kako bi pokušali smanjiti efekt infodemije pojavljuju se posebne grupe ljudi(eng. Fact checkers) koji šire dokazane informacije koje se u većini slučajeva skrivaju. Potraga za valjanim informacijama, koja je u današnje vrijeme sveprisutnosti interneta ne predstavlja veliki izazov, važan je dio razumijevanja i učenja o okolini. Kako se pandemija sve više širila pokazala se potreba za izradom ontologije vezane za COVID-19. Najpoznatiji primjer takve ontologije je COVID-19 ontology koja je nastala na sveučilištu Oxford te sadrži 2270 klasa i 38987 aksioma od kojih su 2622 logička aksioma i 2434 deklarativna aksioma. Služi za prikazivanje uloga molekula i ostalih entiteta u virusu, interakcija u domaćinu i životnog vijeka virusa kao i prikazivanje širokog spektra medicinskih i epidemioloških koncepata povezanih sa pandemijom. Performanse ontologije su testirane na Medeline korpusu i COVID-19 korpusu koji su izrađeni na Allen institutu. Znanstveni pristup koji se koristi u ovom radu baziran je nekoliko različitih disciplina poput epidemiološkog modeliranja, molekularne dinamike na superkompjutorima i visokopropusno označavanje preko biblioteka za prenamjenu lijekova [2]. Druga

ontologija koju ću spomenuti je Coronavirus Infectious Disease Ontology (CIDO) koja je izvedena iz Infectious Disease Ontology (IDO) te joj je glavna zadaća prikazati podatke na smislen način koji potiče prikupljanje podataka i njihovu standardizaciju, integraciju, dijeljenje i analizu. CIDO ontologija pokriva nekoliko područja u domeni koronavirusa poput etiologija, transmisije, epidemiologije, patogena, dijagnoza, prevencija i tretmana. CIDO se koristi kao primjer baze znanja za standardizaciju i logičku reprezentaciju heterogenog znanja o koronavirusu [3]. Osim što postoje različite ontologije postoje i različiti principi stvaranja ontologija. Jedan od principa je uporaba deskriptivne logike koja se kombinira sa semantikom za razvoj ontologija. Cilj ovog principa je izgraditi semantički bogato znanje za praćenje COVID-19 pandemije te smanjiti nejasnoća oko nje. Svrha spajanja semantike i deskriptivne logike je povezivanje riječi i značenja da bi se dobila točna tumačenja pandemije [4]. U nastavku ću opisati kako će rad biti strukturiran te što će u kojem djelu biti pokriveno. Drugo poglavlje je posvećeno ontologijama, njihovoj definiciji, izradi, vrstama, primjerima, primjenama te alatu Protégé i njegovoj strukturi i mogućnostima. Svaki spomenuti dio vezan uz ontologije uključujući i alat Protégé biti će detaljnije razrađen u zasebnom poglavlju. Odabrao sam takav pristup radii količine informacija vezanih uz pojmove koje obrađujem. U trećem poglavlju će biti izneseni te detaljnije objašnjeni pojmovi vezani za COVID-19 pandemiju uključujući opis načina prikupljanja podataka te će biti prikazan konceptualni model tih pojmova sa pripadnim objašnjenjem tog modela. Velika većina materijala je prikupljena sa stranice Nacionalnog Centra za Biotehnoške Informacije koji služi kao repozitorij znanstvenih radova sa područja biologije, biotehnologije, genetike te radova vezanih uz COVID-19. Osim toga na stranici postoji aplikacija koja omogućava samostalno istraživanje genoma te otkrivanje dodatnih podataka. Nakon opisa konceptualnog modela podataka slijedi podpoglavljje u kojem će biti detaljnije objašnjen svaki pojam u ontologiji. Također će biti opisane relacije između pojmova kako bi se prikazao smisao podataka te njihova primjena. Zbog same količine dostupnih informacija koje se može naći na internetu modeli koji ću prezentirati u radu neće imati najsvježije informacije jer bi modele trebalo ažurirati jednom dnevno. Posebice to vrijedi za modele lijekova i simptoma koji se brzo mijenjaju zbog novih otkrića u ta dva područja. Nakon toga ću objasniti kako sam prikazao podatke u alatu Protégé te kako sam došao do takve hijerahije te ću kod određenih pojmova koji se mogu klasificirati različito objasniti zašto sam ih klasificirao u odabrane klase. U zaključku ću dati kratki pregled svih podataka iznesenih u radu te dati daljnje smjernice kako razvijati rad.

2. Ontologije

2.1. Definicija Ontologije

Formalni eksplicitni prikaz znanja baziran na podacima koji pomažu znanstvenicima u definiranju zajedničkog vokabulara za djeljenje informacija. Kada je riječ o ontologiji važno je znati kako se u znanosti ona opisuje. Za početak ću napisati kako je opisuje američki informatičar Tom Gruber. On ju opisuje riječima: "An ontology is an explicit specification of a conceptualization. The term is borrowed from philosophy, where an ontology is a systematic account of Existence. For knowledge-based systems, what "exists" is exactly that which can be represented." [5] Prema američkoj profesorici Catherine Legg formalna ontologija je: "Strojno čitljiva teorija najtemeljnijih pojmova potrebnih da bi se shvatila pripadnost pojedine informacije određenoj domeni znanja" te je također „teorija o svim vrstama stvari koje postoje uključujući i njihove odnose [6]. Treći opis ontologije dao je Robert T. Neches koji kaže da „Ontologija određuje osnovne pojmove i odnose koji obuhvaćaju vokabular tematskog područja kao i pravila kombiniranja pojmova i odnosa za definiranje ekstenzija vokabulara“ [7]. Neki od razloga definiranja takvog vokabulara su:

- djeljenje unaprijed poznate strukture informacija između ljudi,
- mogućnosti ponovnog korištenja znanja iz određene domene,
- iskorištavanje domene znanja kao pokriće za eksplicitne pretpostavke o nečemu,
- za ograđivanje domene znanja od operativnog znanja,
- za analiziranje domene znanja.

Jedan od glavnih ciljeva ontologije je stvoriti unaprijed poznatu strukturu informacija koja se može koristiti kod agregiranja informacija u zajedničku domenu. Omogućavanje ponovnog iskorištavanja znanja jedan je glavnih pokretača razvoja ontologija u današnje vrijeme. Na primjer, modeli u određenoj domeni moraju biti reprezentativni. Reprezentativnost modela mora sadržavati podatke o promjeni kroz vrijeme, važna stanja u vremenu i relativna mjerenja u vremenu. Također, ako se treba izgraditi veći ontološki model može se napraviti više manjih modela koji opisuju određen dio problema te se njihovom kombinacijom može dobiti potpuni model sustava. Moguće je koristiti generalne ontologije te ih proširiti kako bi ih prilagodili novoj domeni. Koristeći eksplicitne domene znanja daje mogućnost promjene i prilagodbe implementacije novim pretpostavkama ako se domena znanja promjeni. Dodatno, eksplicitne domene znanja su korisne za nove korisnike koji žele naučiti značenje pojmova u domeni. Odvajanje domene znanja od operacijske domene još je jedan od razloga korištenja ontologija. Analiza domene znanja je moguća samo u slučaju kada je specificiran svaki dio te domene te je takva analiza značajna kada se želi koristiti i proširiti ta domena. Razvoj ontologija pojedinih domena u većini slučajeva nije taj da bi se domene koristile same za sebe, već da bi predstavljale set podataka kojeg drugi programi mogu koristiti. Različite metode za rješavanje problema i softverski agenti koriste ontologije i baze znanja kao ulazne podatke za svoje procese preko kojih donose odluke. Jedan od primjera gdje se ontologija može koristiti je u restoranima gdje se preko ontologije svakom gostu koji je naručio jelo preporučuje vino koje najbolje paše uz to jelo. U načelu ontologije su veoma slične objektno orijentiranom programiranju, ali je bitna razlika ta što se objektno orijentirano programiranje koncentrira na metode, dok se ontologija koncentrira na strukturu samih podataka koje prikazuje. Zbog prikaza strukture klasa rješenja zadataka koja daje ontologija ne moraju se nužno poklapati sa rješenjima koja su dobivena preko objektno orijentiranog programiranja. Kako ontologija zavisi od podataka tako ne postoji standardizirana metodologija njezina razvoja već se preko prikupljenih podataka definira njihov model. Kada se priča o definiciji ontologije različita područja na

različite načine definiraju ontologiju. Primjerice kod umjetne inteligencije definicija je potpuno različita te kaže da je ontologija formalni eksplicitni opis u domeni klase čije uloge opisuju atribute te klase te imaju posebna ograničenja. Ontologija zajedno sa setom individualnih instanci služi za konstrukciju baze znanja. Zbog opisa koncepata u pojedinoj domeni dobro oblikovanje klasa je u fokusu prilikom dobrog definiranja ontologije. Svaka klasa može imati potklase te se pomoću potklase definira veza općeg i pojedinačnog. Klasa može imati i uloge koje opisuju svojstva pojedinih klasa i instanci. U praksi razvoj ontologije podrazumjeva: definiranje klasa u ontologiji, pozicioniranje klasa u taksonomijskoj hjerarhiji, definiranje uloga i opisa za te uloge, popunjavanje uloga vrijednostima. Nakon definiranja ontologije može se kreirati baza znanja definiranjem instanci klase i popunjavanjem tih instanci vrijednostima i ograničavanjem uloga. Kako ne postoji posebna metodologija može se doći do određenih pravila koji opisuju kako se koristi ontologija. Pravila su sljedeća:

- ne postoji jedinstven način definiranja ontologije, već mogu postojati brojne alternative,
- proces razvoj ontologije je iterativan,
- koncepti u ontologiji trebali bi što bolje predstaviti objekte i relacije između objekata u domeni.

Proces razvoja ontologije je iterativan i traje kroz cijeli životni vijek postojanja ontologije. Kod razvoja ontologije bitno je odgovoriti na nekoliko pitanja kako bi znali u kojem smjeru razvijati ontologiju. Pitanja su sljedeća:

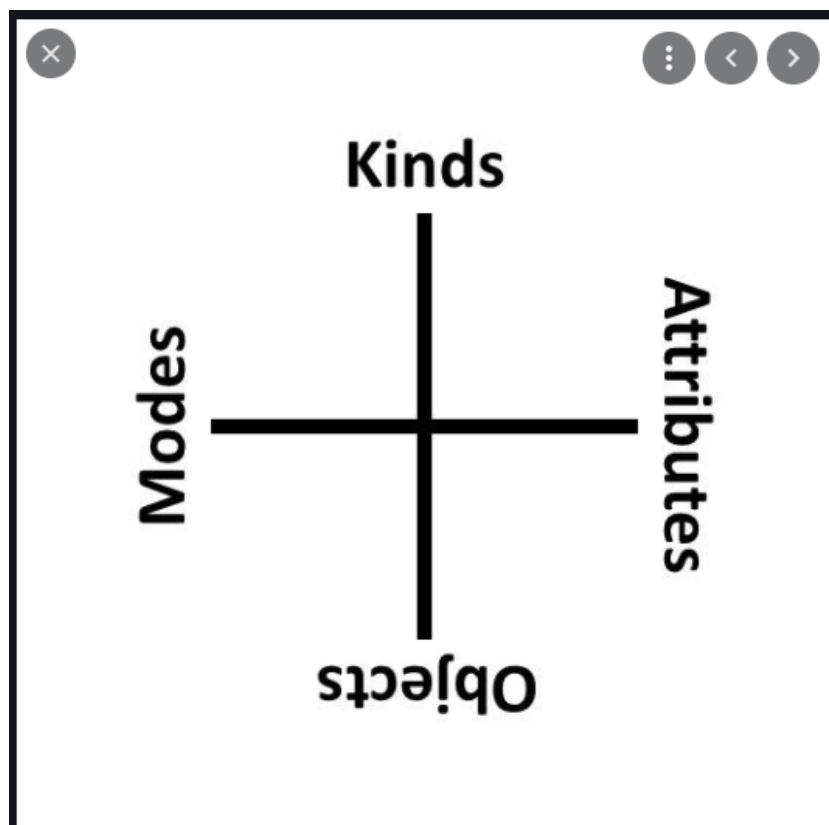
- što se novom ontologijom želi postići,
- koju domenu će nova ontologija pokrivati,
- na koja pitanja će nova ontologija moći dati odgovore,
- tko će ontologiju koristiti te tko će ontologiju koristiti.

Ako će se ontologija koristiti za procesiranje prirodnog jezika u ontologiji bi trebalo biti što više sinonima kako bi se mogao određeni tekst isparsirati na više načina. Jedan od načina kako možemo saznati domenu koju određena ontologija pokriva je da se napravi lista kompetentnih pitanja na koju baza znanja može odgovoriti. Nakon toga treba se vidjeti sadržava li ontologija dovoljno informacija za odgovoriti na pitanja te jesu li odgovori dovoljno detaljni [8].

2.2. Vrste ontologije

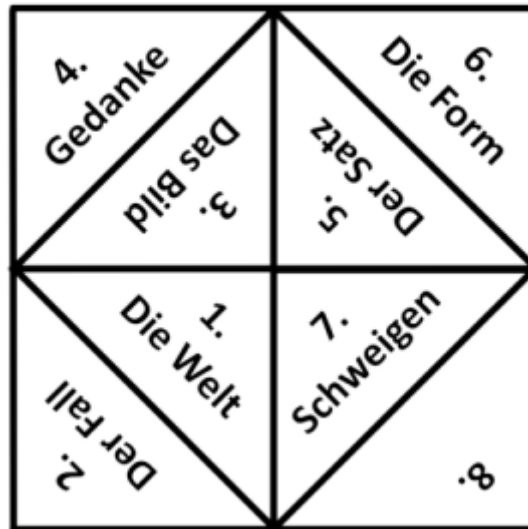
Kada se govori o ontologiji postoji mnogo različitih vrsta koje ovise o načinu kreiranja ontologije, području primjene, domeni koju pokrivaju, poziciji i broju entiteta. U nastavku ću dati primjere i opise svake pojedine vrste. Najjednostavniji primjer ontologije je ravna ontologija koja kaže da svaki objekt ima istu razinu subjektivnosti kao svaki drugi objekt te samim time ne postoji koncept viših i nižih objekata u takvoj ontologiji. U ravnoj ontologiji niti jedan objekt nije subjektivniji od drugih i svi su objekti jednostavni. Ključna stvar u prepoznavanju ontologije je djelovanje jednog objekta na drugi objekt. Jedan od zagovornika je američki filozof William van Orman Quine te se ovakav pristup ontologiji naziva Quinov pristup. Polikategorična ontologija sastoji se od većeg broja kategorija koje su nezavisne te svaki entitet ima točno jednu kategoriju. Najpoznatiji primjer takve ontologije je ontologija četiri kategorije Edwarda Jonathana Lova čiji su koncepti sam objekt, tip objekta, način funkcioniranja objekta i atributi objekta. Ontologija četiri kategorije temelji se na percipiranju strukture bazirane na dva odnosa i razlikama u kategorijama koje sudjeluju u ta dva odnosa. Prvi je odnos između bitnih kategorija što su objekt i tip i manje bitnih kategorija što su način funkcioniranja i atributi objekta.

Nadalje, postoji hijerahijska ontologija koji opisuje stupanj fundamentalnosti entiteta koje posjeduje. Sam cilj takve ontologije je prikazati koji su entiteti fundamentalni i kako ne fundamentalni entiteti ovise o njima. Fundamentalni entiteti se razlikuju od ne fundamentalnih entiteta u tome što ne sadrže puno drugih entiteta. Spomenuo sam ranije ontologiju četiri kategorije te se temeljem njenih saznanja opisuju ontologija činjenica i ontologija stvari. Ontologija činjenica i ontologija stvari su u načelu slične, no razlikuju se u shvaćanju pojma kategorije. Ontologija činjenica shvaća kategoriju kao kategoriju činjenica, dok ontologija stvari shvaća kategoriju kao kategoriju stvari. Slogan za ontologiju stvari Austro-Britanskog filozofa Ludwiga Wittgensteina glasi: „Svijet se sastoji od činjenica, ne od stvari“. Postoje još konstitutivna ontologija i blob teorija koje pripadaju relacijskim ontologijama koje se bave internim struktirama objekta. Konstitutivna ontologija smatra da je interna struktura objekta važna za njegovu konstituciju, dok blob teorija govori da objekti nemaju solidnu strukturu te da se ona mijenja ovisno o svrsi objekta. U informacijskoj tehnologiji ontologija se klasificira na različite načine korištenjem stupnja abstrakcije i područja aplikacije te se razlikuju: Viša ontologija koja podržava razvoj ontologije te se još naziva meta-ontologija, domenska ontologija koju sam spominjao i koja opisuje određeno područje u informatici, ontologija sučelja je koncept koji opisuje međusobnu interakciju dvije discipline i procesna ontologija koja opisuje ulaze, izlaze, ograničenja, sekvenciju informacija i procese u IT industriji. U nastavku ću prikazati primjere ranije spomenutih ontologija. Za početak ću prikazati kako izgleda ontologija četiri kategorije:



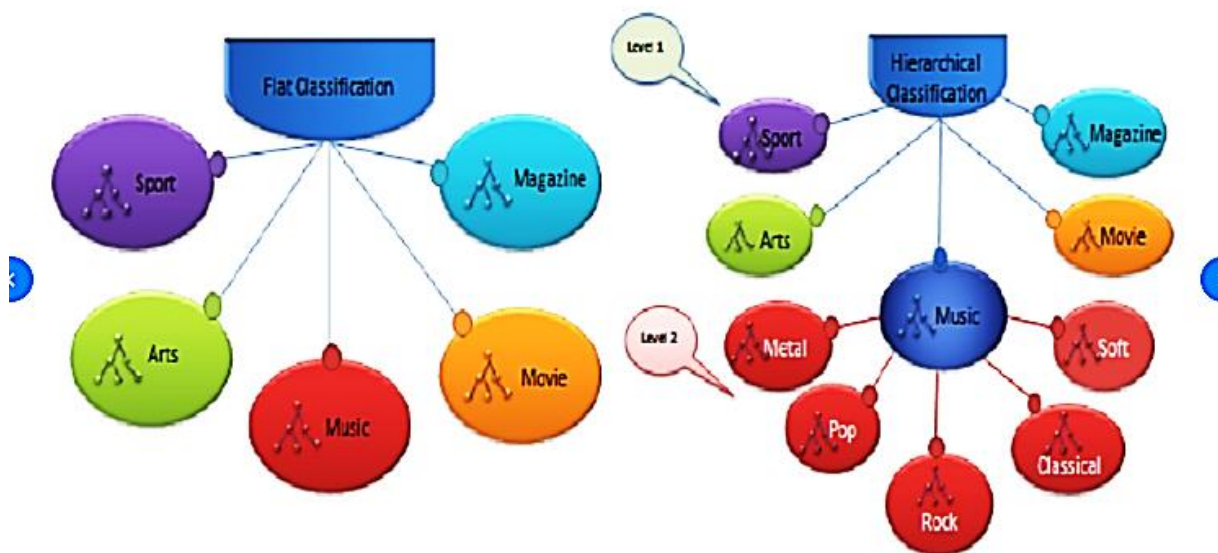
Slika 1 E.J.Lowe Ontologija četiri kategorije [1]

Potom sam spomenuo austrijsko-britanskog filozofa Ludwiga Wittgesteina čija je ontologija nazvana Logika Traktatusa podupire njegovu izjavu da se svijet sastoji od činjenica ne od stvari te će primjer njegove ontologije biti prikazani u nastavku:



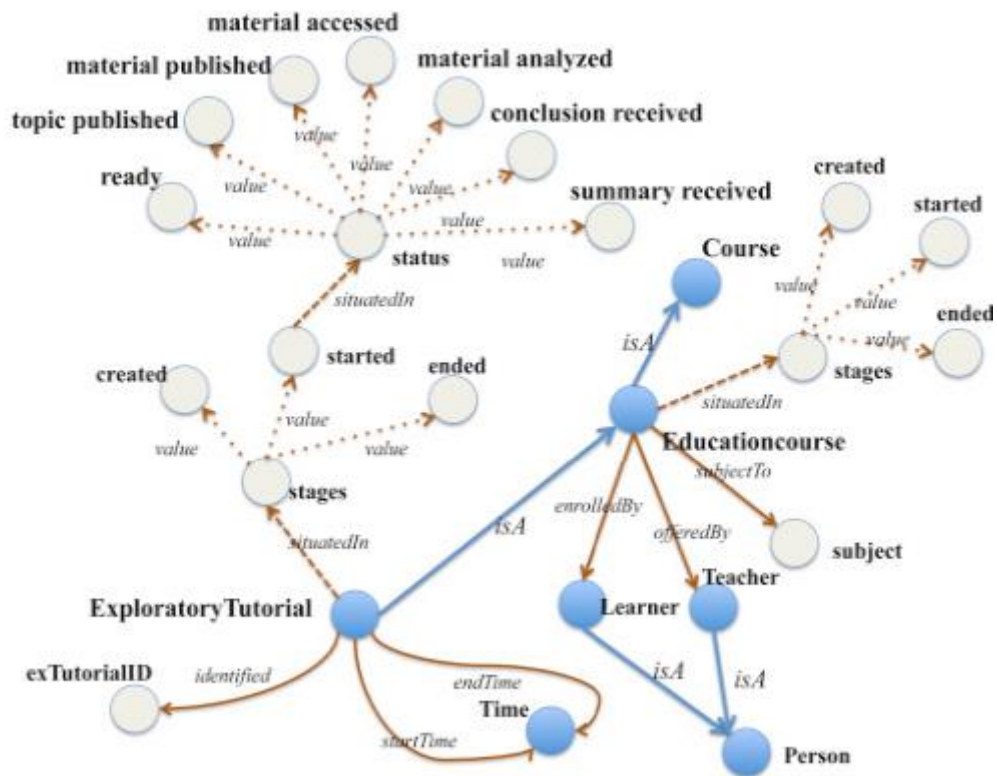
Slika 2 Ludwig Wittgstein Logika Traktatusa [2]

Također postoji nekoliko načina klasifikacije ontologije od kojih su najpoznatije ravna ontologija, hijerarhijska ontologija, domenska ontologija i Guarinova ontologija. Ravna ontologija je ontologija koja uvijek ima uz glavni pojam jednu razinu pojmova koji su povezani s njom za razliku od hijerarhijske ontologije kod koje podpojmovi glavnog pojma mogu imati svoje podpojmove. U nastavku ću prikazati kako izgledaju ravna i hijerarhijska ontologija:



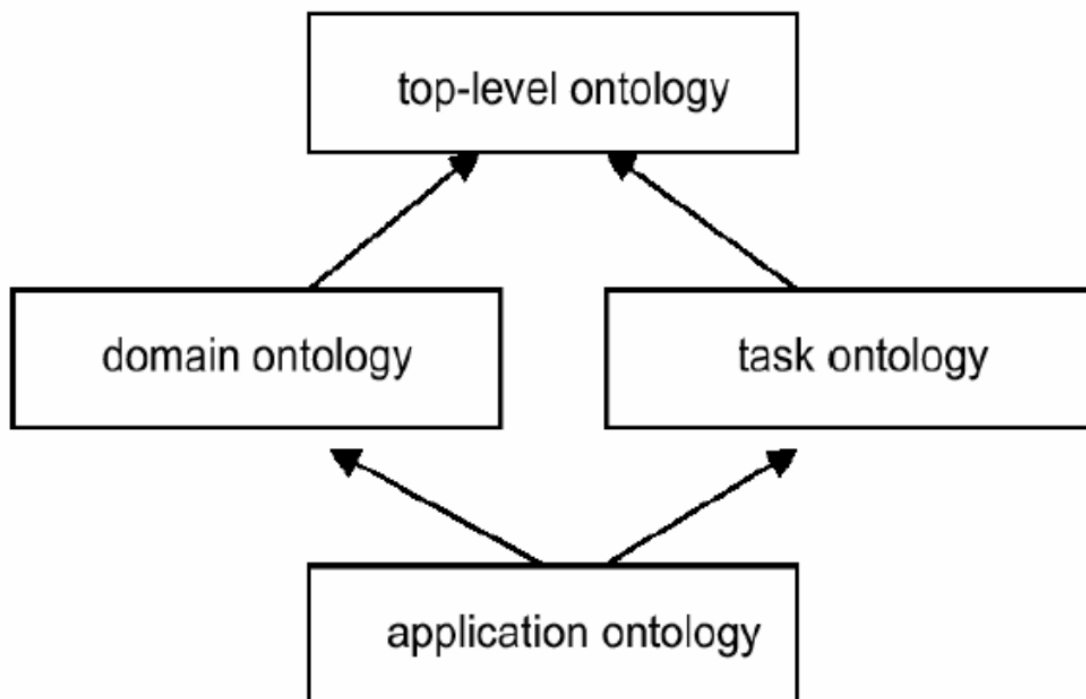
Slika 3 Prikaz ravne i hijerarhijske ontologije[3]

Domenska ontologija je ontologija koja se bazira na prikazu pojmova u određenoj domeni znanja te njihovih relacija. Domenska ontologija ima poseban značaj u informacijskoj tehnologiji za klasificiranje pojmova. Ona je u načelu slična hijerarhijskoj ontologiji samo što umjesto pojmova prikazuje hijerarhiju čitavih područja u određenoj domeni:



Slika 4 Prikaz Domenske ontologije [4]

Nicola Guarino je 1997. godine definirao jednostavnu ontologiju za prikaz procesa u informacijskoj tehnologiji. U nastavku ću prikazati sliku Guarinove ontologije:



Ontology classification according to [Guarino, 1998].

Slika 5 Ontologija prema Nicoli Guarinu[5]

2.3. Primjeri ontologije

U nastavku ću navesti nekoliko primjera ontologija te ću se koncentrirati na medicinske ontologije. Prva ontologija je Common Anatomy Reference Ontology (CARO) što je ontologija više razine koja služi za prikaz interoperabilnosti između različitih vrsta, Disease ontology (DO) je ontologija namjenjena prikazu različitih bolesti i medicinskog vokabulara vezanih uz tretmane tih bolesti, Fundamental model of anatomy (FMO) je domenska ontologija za eksplicitno deklarativno znanje prikaza ljudske anatomije, Gene ontology (GO) je ontologija koja je nastala kao napor znanstvenika da se objedine podaci o genetici pohranjeni u različitim bazama podataka diljem svijeta, Human Phenotype ontology (HPO) je ontologija koja prikazuje fenotipske abnormalnosti u zarazama, Infectious Disease Ontology (IDO) je set interoperabilnih ontologija koje pokrivaju domenu zaraznih bolesti te je u svojoj srži dio generalne ontologije zaraznih bolesti (IDO-Core) čiji su entiteti relevantni za najzaraznije bolesti, Microarray Gene Expression Data Ontology (MGED) su koncepti, definicije i resursi koji služe za standardizaciju opisa kod eksperimenata sa mikropoljima, Mammalian Phenotype Ontology je ontologija koja je nastala kroz napore zajednice da se naprave standardizirani termini za anotiranje fenotipskih podataka sisavaca, Ontology for Biomedical Investigation(OBI) je integrirana ontologija za klinička istraživanja, Ontology for Clinical Investigation (OCI) je ontologija za anotiranje rezultata kliničkih istraživanja, Ontology for General Medical Science (OGMS) je formalna ontologija za kliničke fenotipe, Phenotypic Quality Ontology (PTO) je ontologija koja u suradnji s drugim ontologijama bolje opisuje fenotipove, Vaccine Ontology (VO) je ontologija za reprezentaciju znanja o cijeplivima [9].

2.4. Primjene ontologije

U nastavku ću opisati gdje se sve i na koji način primjenjuju ontologije. Obzirom da su ontologije formalizirani eksplicitni način prikaza nekog sustava tako je razvoj ontologije prisutan u većini poput biologije, umjetne inteligencije, procesiranja prirodnog jezika i podatkovnoj znanosti. U podatkovnoj znanosti, zbog trenda rasta podataka, korištenje ontologije je bitno zbog toga što je potrebno podatkovna jezera pretvoriti u pametna podatkovna jezera. To se odvija preko podatkovnih tvornica koje procesiraju podatke iz podatkovnih jezera i spremaju ih u pametna podatkovna jezera. Ovakav način rada je tranzicija iz tradicionalnih bazi podataka u grafičke baze podataka temeljenih na semantici. Na takav način se dobivaju podaci prilagođeni čovjeku sa posebni identifikatorima i povezani sa čestim poslovnim terminima. Principe bazirani na ontologiji puno je jednostavnije koristiti i nadograđivati te će vremenom ontologije postati osnova za izgradnju sustava. Veliki broj ontologija je prisutan u biologiji te posebice u genetici. Razlog za to je potreba da se napravi hijerarhijski opis podataka vezanih za različita biološka istraživanja. U biologiji je znanje kompleksno te ga nije moguće na jednostavan način spremati u bazu podataka te se rade posebne biološke ontologije koje definiraju na koji način se pojedini koncept može iskoristiti kako bi se prikupljeno znanje moglo spremati u molekularnu bazu podataka. Prilikom procesiranja prirodnog jezika važno je kreirati domensku ontologiju za pojedini jezik. Na takav način se mogu definirati pravila koja se mogu koristiti u različitim zadacima vezanim uz procesiranje prirodnog jezika poput sažimanja teksta, ekstrakcije informacija, izradi sustava za pitanja i odgovore te izradi sustava za računalno prevođenje. Problem kod ontologija u području procesiranja jezika je taj što se određeni algoritmi poput algoritma za sažimanje teksta krenu implementirati prije no što se napravi ontologija. Problem implementacije prije definiranja ontologije pa je predloženo da se prilikom procesiranja teksta automatski definira ontologija za taj tekst. To se radi na način da se u početku mala ontologija koja se prilikom procesiranja i prepoznavanja teksta nadograđuje novim znanjem. Takva ontologija se koristi u projektima za upravljanje i prezentiranje informacija. Prisutnost sve većeg broja ontologija za različite jezike vodi prema interoperabilnosti problema. Nadalje to vodi do definiranja nove metodologije i novog okvira za spajanje heterogenih ontologija baziranih na WordNet-u. Kako bi se to ostvarilo koriste se sučelja prirodnog jezika koja se u većini slučajeva odnose kao sustavi pitanja i odgovora smanjene domene. Iako se već pokušavalo na takav način automatski definirati domenu prirodnog jezika to i dalje predstavlja problem posebice kad je tekst pisan na jeziku koji nije engleski te je pridružen nekoj posebnoj domeni. U tom slučaju struktura ontologije ovisi o načinu kako tu strukturu percipira korisnik ontologije. Kako bi korisnik što lakše mogao shvatiti i izgraditi ontologiju prirodnog jezika potrebno je definirati i implementirati polu-automatske softverske alate za izgradnju ontologije prirodnog jezika. Takvi alati postoje već na internetu. Jedan od alata koji se spominje je DOODLE-OWL što je interaktivna razvojna okolina za razvoj ontologije za engleski japanski jezik. Iduće veće područje gdje se ontologija koristi kao sredstvo za modeliranje i dohvaćanje podataka bazirani na semantici putem domenske ontologije. Ono zbog čega se koristi dohvaćanje podataka bazirano na ontologiji je stvaranje sučelja između pojmova koji se traže i podataka kako bi se rezultati što više približili zahtjevima korisnika prilikom njihovih istraživanja. Jedna od većih prednosti domenskih ontologija je kombinacija domenskog znanja i semantičkog modela podataka. Ontologije se mogu iskoristiti kako bi se definirale poveznice između tipova semantičkog znanja. Da bi mogli koristiti određenu ontologiju za naš zadatak moramo prvo pregledati sve ontologije dostupne u našoj domeni rada i vidjeti kako one zaključuju na našim podacima. Dva najpoznatija okvira za definiranje ontologije su Resource Description

Framework(RDF) i Ontology Web Language(OWL). Najveći razvoj RDF okvira baziran je na pravilima i strukturi RDF/RDFS-a, dok se OWL ontologija bazira na tri podjezične strukture što daje dodatnu ekspresivnost u odnosu na RDF okvir. Te tri podjezične strukture su OWL-Lite, OWL-DL i OWL-Full i razvijene su na temelju RDF i DAML-OIL okvira. Kratica DAML-OIL označava DARPA Agent Markup Language i Ontology Inference Layer. OWL-DL se upotpunjuje pravilima semantičkog jezika za web(Semantic Web Rule Language, SWRL). Ključna karakteristika SWRL-a je njegova jednostavnost i uska integracija sa postojećim OWL jezikom [10].

3. Alat Protégé

Alat Protégé je alat otvorenog koda namjenjen za izradu ontologija, izgradnju inteligentnih sustava i sustava za upravljanje znanjem. Obzirom da je alat otvorenog koda za njegov razvoj važna je zajednica koja, kako se Protégé sve više razvija, konstantno raste i unapređuje softver. Softver je nastao na Sveučilištu Stanford koje je prve verzije softvera radio u suradnji sa Sveučilištem u Manchesteru te je pisan u Javi i koristi Swing za implementaciju sučelje. Osnovni razlog za razvoj Protégé-a je omogućiti znanstvenicima koji se bave biomedicinom da mogu što lakše prikazati svoja otkrića. Svjetska Zdravstvena Organizacija ga koristi kako bi kreirala razne ontologije poput Svjetske Klasifikacije Zaraza i Svjetske Klasifikacije Tradicionalnih Medicina. Kako se podaci sve više povećavaju tako je potrebno razviti metode i alate za prevođenje biomedicinskih podataka u akcije. U Protégé-u model znanja kao i razne terminologije i ontologije reprezentirane su preko okvira koji se u samom alatu dijele na klase, *slotove* i aspekte. Klase koje se ponekad zovu i koncepti reprezentiraju entitete koji služe za definiranje domene u kojoj se čitava ontologija razvija. *Slotovi* opisuju pojedinosti klase i njezinih atributa te se definiraju putem uređivača slotova. Uređivač *slotova* može se koristiti za definiranje i mijenjanje aspekata pojedinih slotova. Primjerice, ako imamo unikatni identifikator koji je tipa string te njegova vrijednost nema tip potrebno je napraviti *slot* za njega sa pripadni aspektom. Aspekti pojedinih *slotova* objašnjavaju karakteristike tih *slotova* te omogućuju da se izraze činjenice da određena klasa mora postojati. Instanca je okvir koji je izgrađen od barem jedne klase sa posebnom vrijednosti za *slotove*. Baza znanja se sastoji od ontologije i instance pojedine klase sa specifičnim *slotovima*. Svaka ontologija se sastoji od klasa, *slotova*, aspekata i aksioma. Aksiomi su posebna dodatna ograničenja definirani da bi se ontologija bila što bolja reprezentacija podataka. Klase, *slotovi* i aspekti su osnovne gradivne jedinice svake terminologije [11].

3.1. Arhitektura i mogućnosti Protégé-a

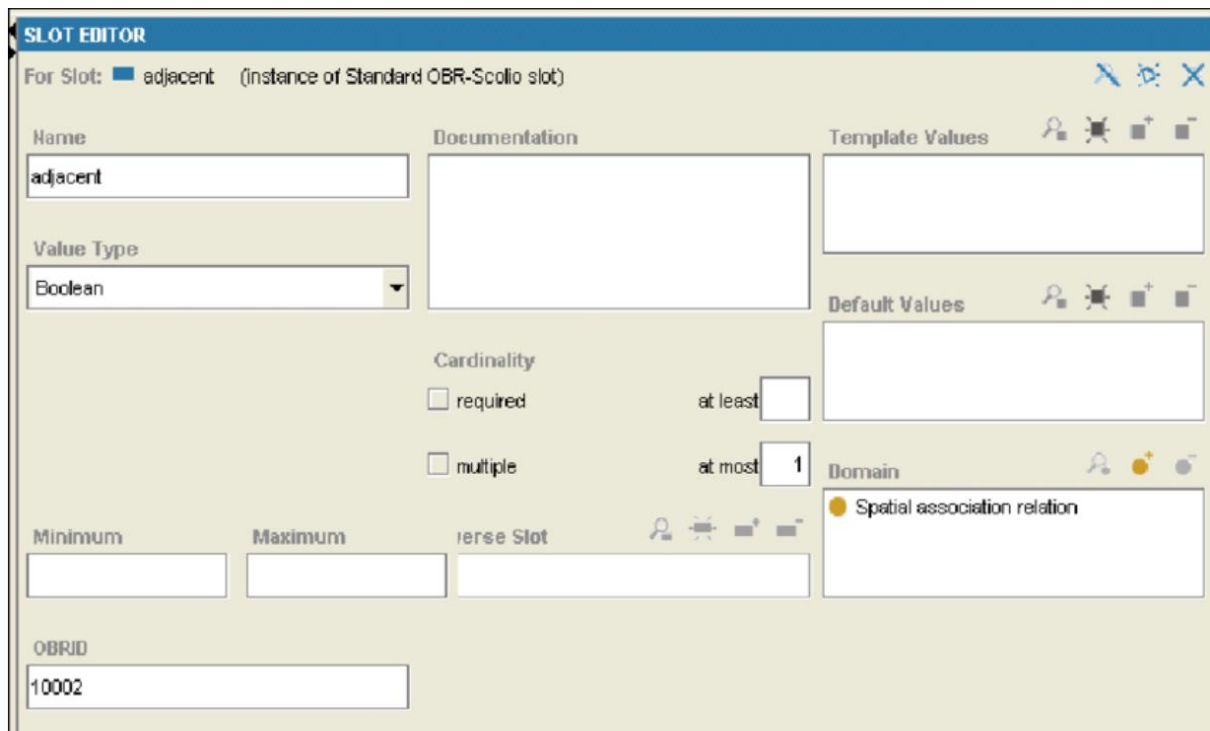
U nastavku ću nešto više reći o arhitekturi i mogućnostima alata Protégé. Protégé je implementiran u javi te je dostupan na većini operacijskih sustava današnjice te ima multistupanjsku arhitekturu. Zahvaljujući multistupanjskoj arhitekturi Protégé omogućuje nekoliko funkcionalnosti. Funkcionalnosti su slijedeće: razina za skadištenje ontologije, razina modela znanja, grafičko korisničko sučelje (GUI) i aplikacijsko programsko sučelje. Zahvaljujući aplikacijskom programskom sučelju korisnik može pristupiti grafičkom sučelju, dodacima i ontologiji bez problema što čini Protégé modularnom i fleksibilnom aplikacijom. U Protégé je implemerana arhitektura koja omogućuje programerima razvoj podataka koji mogu proširiti mogućnosti same aplikacije. Neke od naprednih mogućnosti su napredno uvođenje i izvođenje ontologija, vizualizacija ontologija i validaciju ontologija. Kako je uvođenje i izvođenje ontologija ključno kod svake aplikacije zbog toga što postoji puno različitih formata u koje se može spremi ontologije i terminologije tako postoji nekoliko Protégé podataka koji omogućuju izvođenje podataka u različite formate uključujući Protégé format, RDF, OWL I XML. U Protégé-u postoji relacijska baza u pozadini koja se može koristiti u slučaju da imamo veće modele. Programeri mogu kreirati vlastite metode uvođenja i izvođenja u slučaju da rade sa posebnim formatima [12].

3.2. Aplikacija i programsko sučelje

Protégé se može koristiti kao samostalna aplikacija ili preko klijenta koji je spojen sa udaljenim serverom. Kad je aplikacija pokrene korisnik ima mogućnosti kreirati novu ontologiju, otvoriti postojeću ili uvesti ontologiju različitih formata. Prilikom kreiranja nove datoteke dodatno se kreira pomoćna datoteka koja sadržava metapodatke o samom projektu. Jednom kad je ontologija otvorena korisnik koristi sučelje koje je organizirano u posebne ekrane od kojih svaki predstavlja dio procesa

razvoja ontologija i daje različiti pogled na ontologiju. Prvi od ekrana je ekran s klasama u kojem korisnik provede najviše vremena. U njemu se definira čitava ontologija sa svim informacijama o pojedinoj klasi uključujući njezine atribute, stablo klasa i veze između klasa. U stablu pojmovi se nalaze na različitim razinama što označava odnose između nadklase i podklase. U terminologiji se za nadklase i podklase koriste izrazi roditelji i djeca. Također se u tom ekranu mogu prevlačiti klase kako bi se reorganizirao njihov odnos u hijerarhiji te kreirale i preimenovala klase. Nakon definicije klase potrebno je definirati *slotove* za pojedine klase uz pomoć uređivača *slotova*. Uređivač *slotova* može prikazati sve pojedinosti vezane uz *slotove* poput tipa, kardinalnosti i tekstualne dokumentacije. Svaki *slot* ima sustavni opis koji se može mijenjati i specijalizirati za svaku pojedinu klasu. Način na koji se mogu vidjeti *slotovi* ovisi o načinu pristupa *slotovima*. Ako se pristupa *slotovima* preko gumba za sustavni opis onda se može pregledavati i mijenjati sustavni opis te će se novi sustavni opis primjeniti na sve buduće definirane klase, no ako se pristupa *slotovima* preko gumba za prikaz promjena mogu se pregledati i promijeniti svojstva trenutno selektirane klase. Modifikacije na razini klase mogu djelovati na *slotove* trenutne selektirane klase i njezinih podklasa. Uređivač *slotova* može prikazati različite informacije o *slotovima* poput:

- mogućnosti dodavanja bilješki,
- ime *slot*,
- tip vrijednosti,
- kardinalnost *slot*,
- opcionalni minimalni ili maksimalni broj za svaki *slot* koji može biti cijeli ili decimalni,
- opcionalna dokumentacija za svaki *slot*,
- opcionalni inverzni *slot* za svaki *slot* koji može biti tipa klase ili instance,
- opcionalna vrijednost predložka koji može biti definiran za svaki *slot*,
- opcionalna difoltna vrijednost koja može biti definirana za svaki *slot*,
- domena *slot*.



Slika 6 Slot editor [6]

Na svaku klasu u ontologiji moguće je dodati bilješke koje nisu dio ontologije, nego pomažu korisniku oko shvaćanja same ontologije. Sve bilješke su prikazane na ekranu za *slotove*. Na ekranu za *slotove* postoji polje za unos imena *slot*a. Polje za unos imena je osjetljivo na velika i mala slova te se preporuča korištenje malih slova i odvajanje riječi doljnjom crtom. Izbornik tipa vrijednosti omogućuje da se svakom *slotu* odredi koji tip vrijednosti može sadržavati te kada se kreira instanca pojedinog *slot*a tip vrijednosti također određuje kako će se prikazivati instanca u uređivaču instanci. Tipovi vrijednosti mogu biti sljedeći:

- logički tip koji može sadržava vrijednosti točno i netočno,
- klasu što znači da pojedini slot može sadržavati jednu ili više klasa kao vrijednosti,
- decimalni broj,
- instancu što znači da svaki slot može sadržavati jednu ili više instanci kao vrijednosti,
- cijeli broj,
- riječi koje su napisane sa alfanumeričkim znakovima i koje mogu sadržavati razmake,
- simboli koji su definirani kao lista riječi.

Prilikom selektiranja vrijednosti za klasu, instancu ili simbol pojavljuje se dodatni ekran koji omogućava povezivanje vrijednosti i postojećeg *slot*a ili kreiranje novog *slot*a. Nadalje postoji polje kardinalnost u kojem se može definirati broj vrijednosti koje *slot* može ili treba primiti. Kardinalnosti je uvijek zadano da može imati najviše jednu vrijednost. No, vrijednost kardinalnosti se može mijenjati tako da se unesu minimalna i maksimalna vrijednost koju *slot* može primiti. Osim unošenja minimalne i maksimalne vrijednosti zahtjev da se u *slot* unese više vrijednosti u *slot* se može definirati opcijom „multiple“ koja kad je uključena označava da *slot* ima više od jedne vrijednosti. Ako u *slot* unesemo broj veći od jedan tada je opcija „multiple“ automatski uključena, a kad je minimalni i maksimalni broj jednak jedan to znači da je u tom *slotu* dozvoljena točno jedna vrijednost. Važno je reći da kada je opcija „multiple“ uključena u polje *slot*a se mogu unositi duplicirane vrijednosti no njihova uporaba nije česta.

Kada je *slot* sa minimalnom vrijednosti postavljen to znači da taj *slot* može imati vrijednost veću ili jednaku postavljenoj vrijednosti. Ako je vrijednost *slot*a postavljena na nulu to znači da instanca ili klasa sa tim *slot*om ne smije imati negativne vrijednosti. Polje sa dokumentacijom je opcionalno polje u kojem se može opisati značenje i važnost pojedinog *slot*a. No, iako je opcionalno preporučljivo je jer omogućava lakše snalaženje u ontologiji i lakše upravljanje čitavom bazom znanja. Inverzni slot je opcionalan te je dostupan samo za klase i instance. Omogućava kreiranje recipročne relacije između dva *slot*a. Ako je inverzni *slot* napravljen kako treba svaka vrijednost definirana za *slot* automatski definira vrijednost za inverzni *slot*. Ako *slot* ima inverzni *slot* ime inverznog *slot*a se prikazuje nakon imena *slot*a na ekranu koji prikazuje informacije za *slot*ove. Predložak za vrijednosti omogućava da se specificira vrijednost za *slot* na razini klase koji je obavezan te se vrijednost automatski popunjava u svakoj klasi i instanci koja nasljeđuje taj *slot*. Vrijednosti zadane putem predloška ne mogu biti promjenjene niti prepisane na razini instance te nebi trebale premašivati maksimalan broj zapisan u polju kardinalnosti te je preporuka za vrijednosti koje se trebaju mijenjati da se koristi polje za difoltne vrijednosti. Polje za difoltnu vrijednost omogućuje da se specificira difoltna vrijednost za svaki *slot*. To znači da će svaka kreirana klasa koja ima taj *slot* imati u tom *slot*u zapisanu vrijednost koju je korisnik specificirao u difolt polju. Za razliku od predložene vrijednosti difoltna vrijednost se može mijenjati, no također ne smije prelaziti maksimalan broj specificiran u kardinalnosti. Domena pojedinog *slot*a je skup klasa koje imaju definiran taj *slot*.

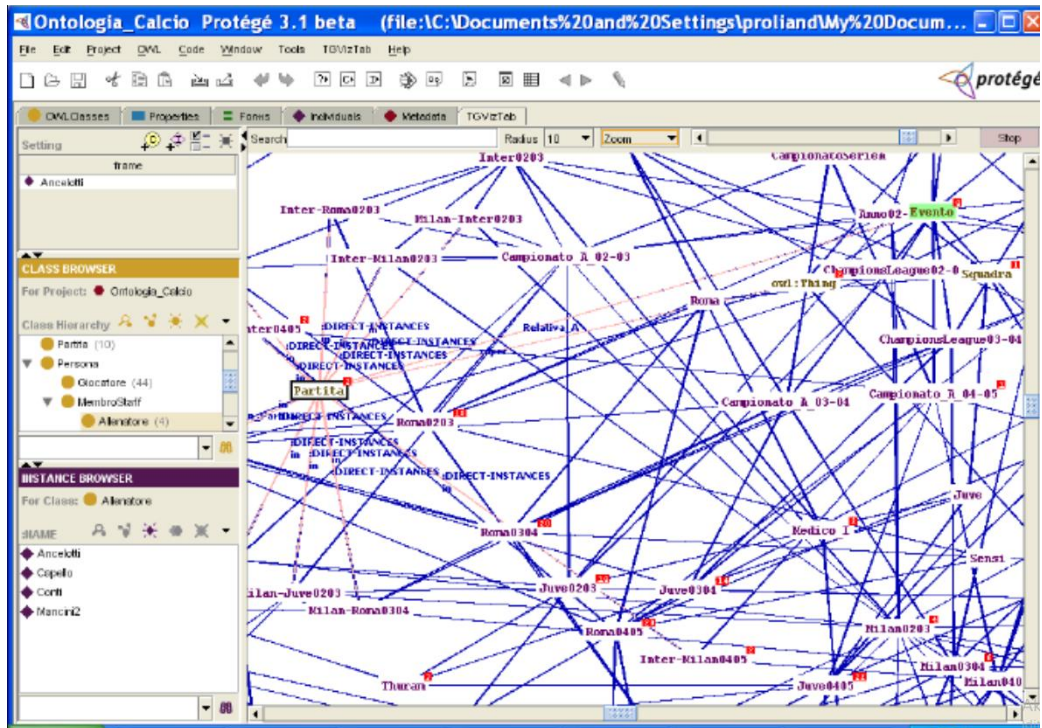
3.3. Paralelna obrada više verzija projekta

Kreiranje ontologije nije deterministički proces u smislu da ćemo početnu ideju o tome kako bi ontologija trebala izgledati dovesti do kraja. Radi priljeva informacija takav način rada čini se manje vjerojatan te je potrebno imati mogućnost spremanja više verzija projekta. Zato u Protégé-u postoji sustav za kontrolu verzija koji osim spremanja verzija omogućuje usporbu dviju verzije. Za razliku od verziranja dokumenata kod verziranja ontologija gledaju se razlike u strukturi umjesto razlika u sadržaju. To znači da korisnik može usporediti dvije verzije ontologije i vidjeti informacije o tome koje su klase dodane, oduzete i promjene te prihvatiti ili odbaciti promjene. Uz kontrolu verzije Protégé ima mogućnost prikaza dvije ontologije jedne do druge što omogućava lakše pronalaženje sinonima.

3.4. Vizualizacija

Iako Protégé prikazuje informacije u obliku stabla mana je takvog prikaza što se ne mogu jednostavno utvrditi koji pojmovi su sinonimi. Stabla obično prikazuju samo pojmove u jednoj relaciji najčešće u „is-a“ relaciji. Od dodataka koji se dosta često koriste za vizualizaciju popularan je Onoviz koji je baziran na Graphvizu koji omogućuje izradu visoko konfigurabilnih vizualizacija ontologije ili nekih njenih djelova selektiranjem klase i instance, prikazom *slot*ova i njihovih veza, bojanjem vrhova i veza između vrhova i vizualizacijom susjednih klasa za posebnu klasu ili instancu koristeći različite operacije. Za vizualizaciju se može koristiti TGvizTab koji koristi TouchGraph biblioteku za vizualizaciju ontologija te je zbog svog kompaktrnog i grafičkog načina vizualizacije jako dobra za ontologije sa jako puno klasa. U nastavku ću navesti i kratko objasniti još nekoliko dodataka koji omogućuju različite načine vizualizacije u Protégé-u. Prvi dodatak koji valja spomenuti je ChangeAnalysis Tab koji daje drugačiji pogled na promjene i anotacije u projektu koji su napravljeni uz pomoć Change Management dodatka. Promjene i anotacije mogu se istraživati preko autora i pojmova u ontologiji te biti vizualizirane s obzirom na životni vijek projekta. Cloud View je dodatak koji omogućuje vizualizaciju ontologije putem prikazivanja relacija između tagova te njihovih pripadajućih težina. DroolsTab omogućuje da se u Protégé-u prikazuju geografske informacije. DroolsTab koristi OpenMap Java biblioteku otvorenog

koda koja služi za prikaz geografskih informacija. Groovy i Clojure programski jezici se koriste za dodatno poboljšavanje dodatka. EditorPane je dodatak koji omogućava korištenje HTML-a i dinamičkih hiperlinkova za sve linkove unutar aplikacije i van nje. Expert System Shell r4f-pro je dodatak koji implementira integrirano razvojno okruženje koje olakšava razvoj ontologije te je bazirano na Protégé 3.5 editoru sa podrškom za vizualnu kreaciju algoritma. HTML Chat je dodatak koji omogućuje da korisnici razgovaraju dok rade na istoj ontologiji. IDESSP je dodatak koji implementira integrirano razvojno okruženje sa vizualnom kreacijom algoritma. Koristi koncept scenarija za opisivanje procesa te programski jezik Clojure za pokretanje procesa. Također implementira integrirano razvojno okruženje za programski jezik Clojure.



Slika 7 Vizualizacija u Protégé-u [7]

4. Razvoj ontologije vezane uz COVID-19

U nastavku ću opisati načine dolaženje do podataka te kako sam predprocesirao podatke u alatu Notion u kojem napravio najviše posla te si olakšao sam razvoj ontologije. Notion je aplikacija za pisanje natuknice koja svojom strukturom podsjeća na WordPress te ima različite mogućnosti skladištenja i prikazivanja informacija tako da se svaka informacija može tagirati posebnim tagom i dodati joj opsežan opis. Također se može razraditi čitava ontologija putem natuknica i dodati natuknicama opis u kojem su definirani atributi koji će biti implementirani prilikom implementacije ontologije. Osim toga alat omogućava korištenje informacija kao poveznica na dodatni opis pojmova te tako jednostavno izgrađivati bazu znanja. U alatu se sve informacije mogu lako mijenjati te je bilo kakvo novo znanje lako uvrstiti u postojeće znanje. Zahvaljujući konceptu poveznica moguće je generalizirati podatke u slučaju potrebe.

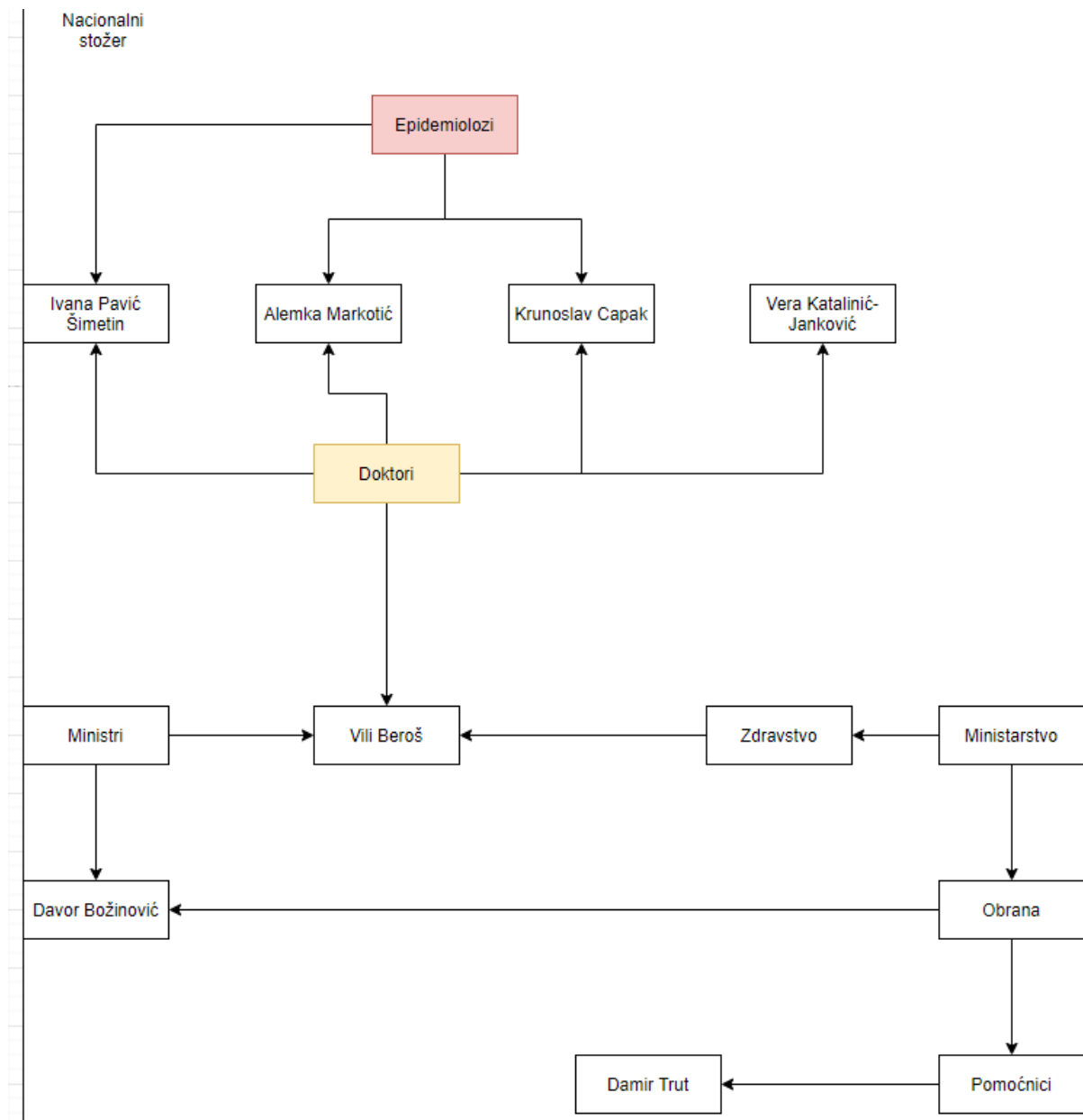
4.1. Procesiranje i priprema podataka za izradu ontologije

Podatke sam prikupljao preko objava novinskih članaka koje bih pronašao na društvenim mrežama i spremio ih u posebnu tablicu koja je bila sortirana od novijih prema starijim člancima. Potom sam svaki članak tagirao ovisno o klasi informacije koja se obrađuje u članku. Klase informacija koje sam definirao su mjere, lijekovi, simptomi i stožer. Nakon toga sam napravio tablicu za svaku klasu informacija kako bi mogao svakoj informaciji dodati attribute koje ću kasnije dodati u ontologiju. Kada sam napravio sve tablice krenuo sam vaditi informacije te ih svrstavati u predviđene tablice te im pridruživati attribute koji ih dodatno opisuju. Pobrinuo sam se da svaka informacija ima barem jedan atribut. Za one informacije koje nisu bile potpuno jasne dodao sam dodatni opis kako bi bile jasnije. U daljnjem tekstu ću raspisati kako su tablice bile izrađene. Svaka tablice u alatu Notion ima stupac Ime koji služi kao primarni ključ tablice te ga nije moguće obrisati nego samo promijeniti naslov. Nadalje, uz ime mjere sam stavio tip mjere koji može imati vrijednosti slabe mjere, srednje mjere i stroge mjere. Uz tip mjere sam koristio atribut područje uspostave koje ima vrijednosti RH za mjere koje su uspostavljene na području čitave Republike Hrvatske te PGŽ za mjere koje su uspostavljene na području Primorsko Goranske županije. Nakon toga sam napravio tablicu simptoma koja kao attribute sadrži težinu simptoma i učestalost simptoma. Težina simptoma može biti blaga, srednja i teška te učestalost može biti česta ili rijetka. Kada sam definirao tablicu mjera i tablicu simptoma definirao sam tablicu lijekova koja ima uz ime samo informaciju o tome jeli lijek potvrđen. Nakon toga sam definirao tablicu stožer koja je najkompleksnija jer sadrži podatke u ljudima koji su članovi stožera, njihovim titulama i funkcijama unutar stožera te sam radio dosta na filtriranju podataka kako mogao prikazati tko su stalni, tko pridruženi, a tko privremeni članovi stožera. Problem kod popisivanja privremenih članova je što se izmjenjuju te je teško zapisati podatke u njima. Također postoji problem da netko za koga se misli da je stalni član može biti pridruženi te takve informacije mogu dovesti do njihove nekonzistentnosti što se dalje može propagirati u ontologiju te tako dovesti do njezine loše implementacije. Zbog toga je proces prikupljanja informacija i izgradnje ontologije iterativan te se informacije trebaju redovito ažurirati.

4.2. Konceptualni prikaz ontologije i objašnjenje prikaza

U nastavku ću prikazati konceptualni prikaz ontologije te objasniti kako sam struktuirao prikaz ontologije. Jedna od prednosti konceptualnog prikaza modela je razumljivost prikaza podataka u odnosu na ontologiju zbog toga što se većina ljudi nije u tolikoj mjeri susretala sa ontologijama te im

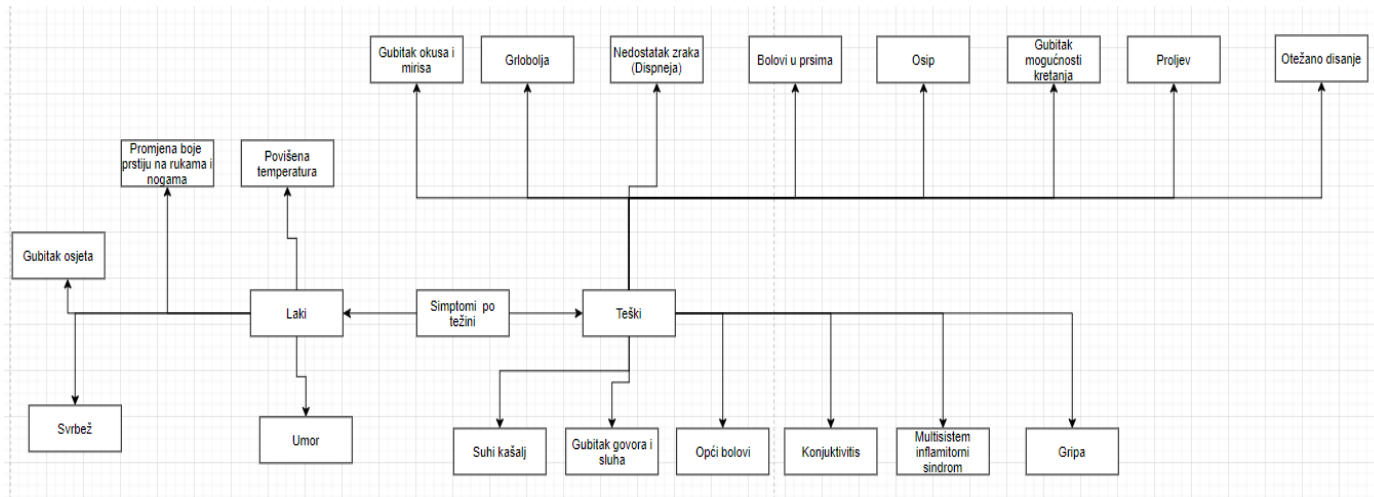
je preko konceptualnog prikaza puno lakše shvatiti hijerarhiju podataka. Druga prednost je ta što je konceptualni prikaz podataka u velikoj većini slučajeva način prikaza podataka te ga ljudi češće koriste u odnosu na ontologiju. No, iako je konceptualni prikaz jako dobar za prikaz podataka on nema definiranu domenu znanja koja bi se koristila za validaciju podataka. U znanosti je kod razvoja hijerarhije važna domena znanja koja nam može osigurati točnost podataka utoliko da ne dopusti prikupljanje podataka koji nisu dio te domene. Konceptualni prikaz sam izradio preko alata Diagram.net u kojem sam napravio četiri dijagrama za svaku od četiri ontologije koje sam implementirao u Protégé alatu. Prvi dijagram koji sam napravio je dijagram Nacionalnog stožera Civilne Zaštite u kojeg sam stavio najvažnije članove koji se pojavljuju u medijima i na sjednicama Nacionalnog stožera. U dijagram sam uvrstio titule članova i funkcije članova koji su članovi vlade. Dijagram je prikazan u nastavku:



Slika 8 Prikaz dijagrama Nacionalnog stožera

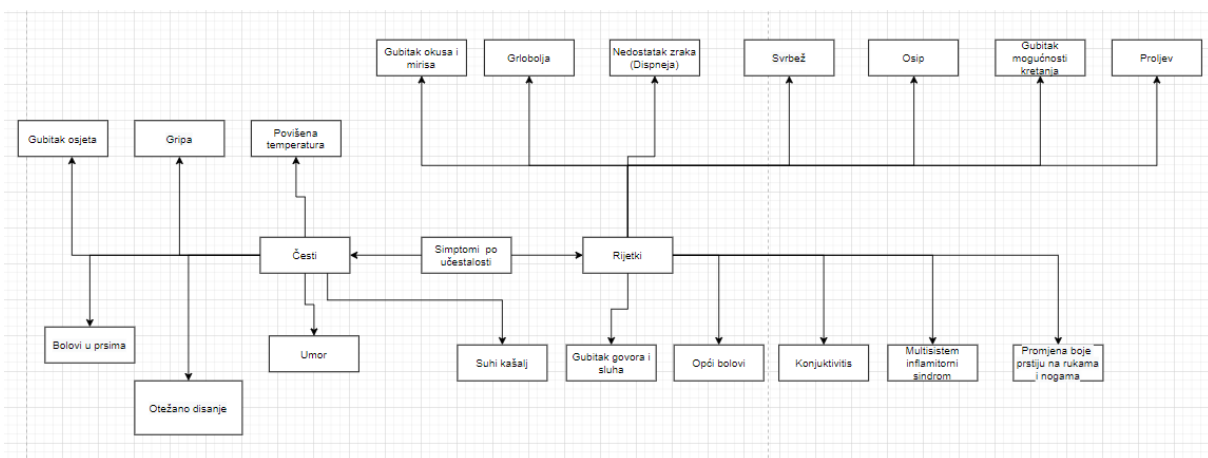
Nacionalni stožer ukupno broji 30 članova od kojih je 7 stalnih koji su prikazani na slici i 23 pomoćna člana. Grupi od 23 pomoćna člana pripadaju predstavnici svakog ministarstva, pripadnici hitnih službi i zdravstvene zaštite.

Nakon toga sam izradio dva dijagrama za simptome. Prvi dijagram prikazuje simptome po težini koji kao podklase imaju slabe i teške simptome. U nastavku ću prikazati simptome po težini:



Slika 9 Dijagrama simptoma po težini

Drugi dijagram je dijagram simptoma po učestalosti koji za podklase ima česte i rijetke simptome.

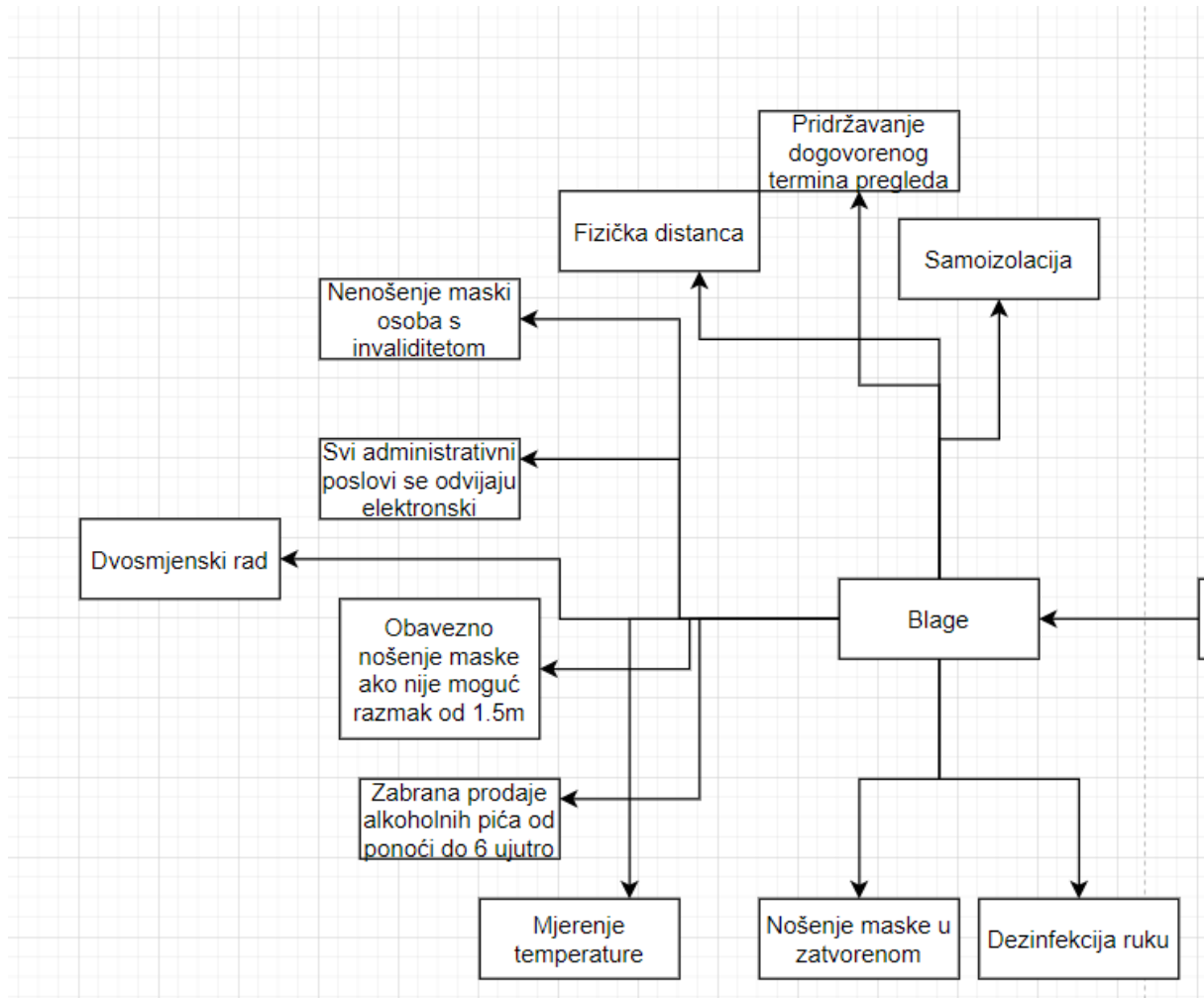


Slika 10 Dijagram simptoma po učestalosti

Oba dijagrama se mogu mijenjat zbog novih otkrića koja mogu izmjeniti shvaćanje određenih simptoma koji mogu najčešće promijeniti učestalost te neki simptomi mogu se početi pojavljivati češće od drugih te se podaci u dijagramima moraju reorganizirati kako bi se mogle prikazati točne informacije. U nastavku ću prikazati dijagrame mjera koje sam podijelio na skupine mjera koje su donesena za područje Republike Hrvatske i za područje Primorsko-Goranske županije te skupinu slabih, srednjih i strogih mjera. Obzirom da je mjera dosta bilo mi je potrebno u diagram aplikaciji proširiti klasu srednjih mjera dodatni poveznica kako bi mogao prikazati sve srednje mjere. U nastavku ću prikazati svaku skupinu mjere na zasebnoj slici. Prva skupina je skupina mjera po području uspostave. Mjere u Primorsko-Goranskoj županiji mogu biti različite jer županijski stožer u većini slučajeva samostalno odlučuje o mjerama koje treba provesti. Dijagram mjera za područje Republike Hrvatske i

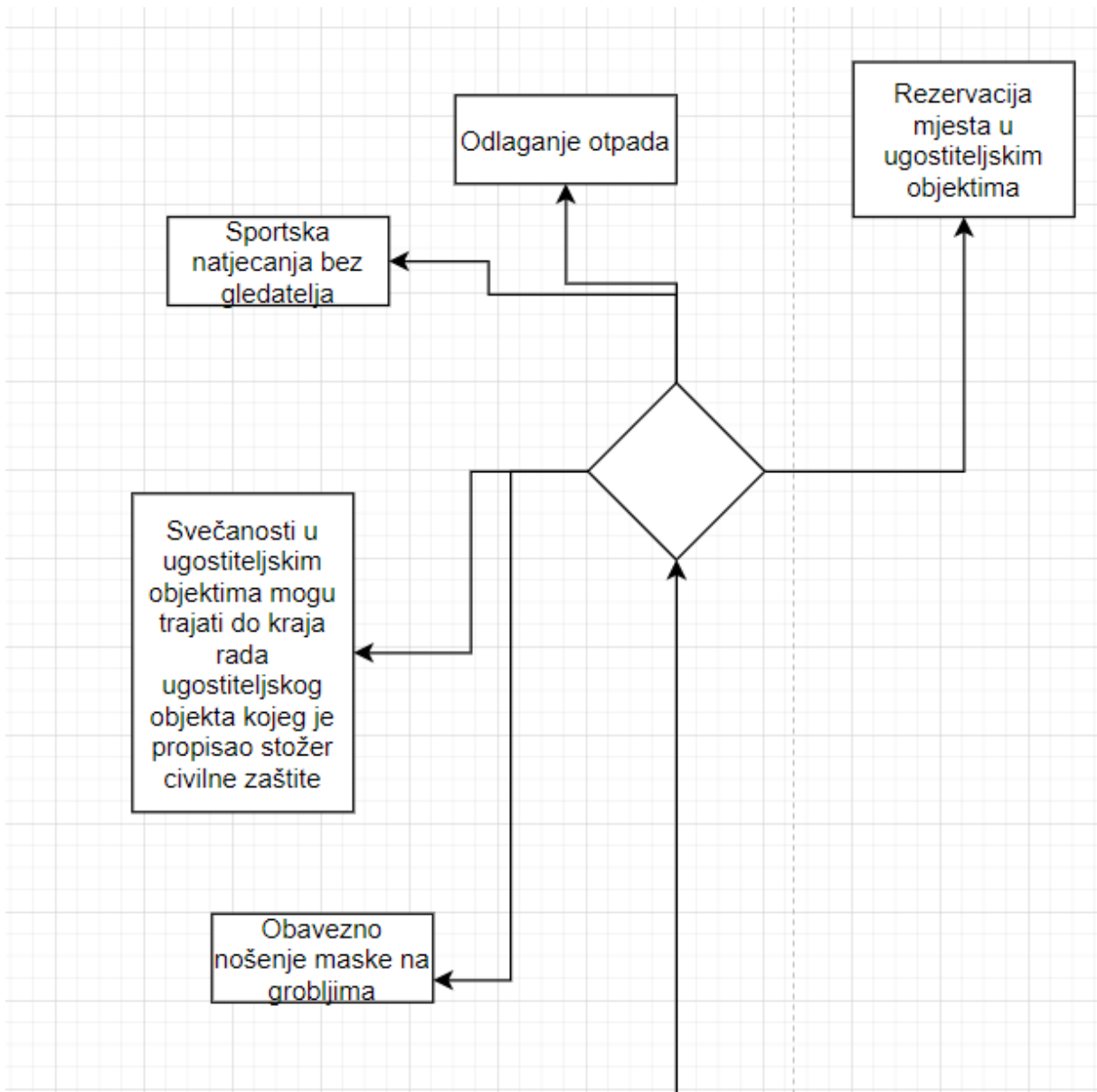
Primorsko-Goranske županije definiraju se hijerarhijski slično kao mjere po tipu gdje su klase na vrhu područja na koja se mjere odnose dok podklase reprezentiraju pojedinu mjeru koja je donesena za to područje.

Nakon prikaza mjera po području prikazati ću mjere po tipu u tri dijagrama od kojih svaki dijagram prikazuje jedan tip mjere. Neke mjere se mogu svrstati u različite klase ovisno o stupnju pandemije, dok su određene mjere poput držanja distance i lockdowna uvijek istog tipa. Prvi prikaz su blage mjere kojih je najmanji broj i njih je bilo lako prikazati:

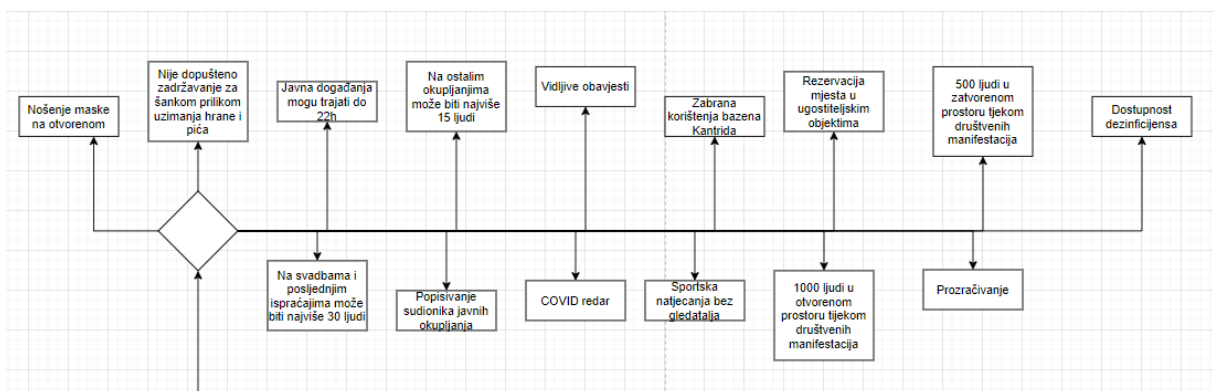


Slika 11 Dijagram blagih mjera u Republici Hrvatskoj

Najveći broj mjera klasificirao sam kao srednje jer su implementirane da bi se pandemija bolje kontrolirala, a ipak nemaju toliko strogoće u sebi da bi se mogle svrstati u stroge mjere. Kako zbog količine mjera nije moguće prikazati jedan dijagram mjere će biti prikazane na iduće dvije slike:

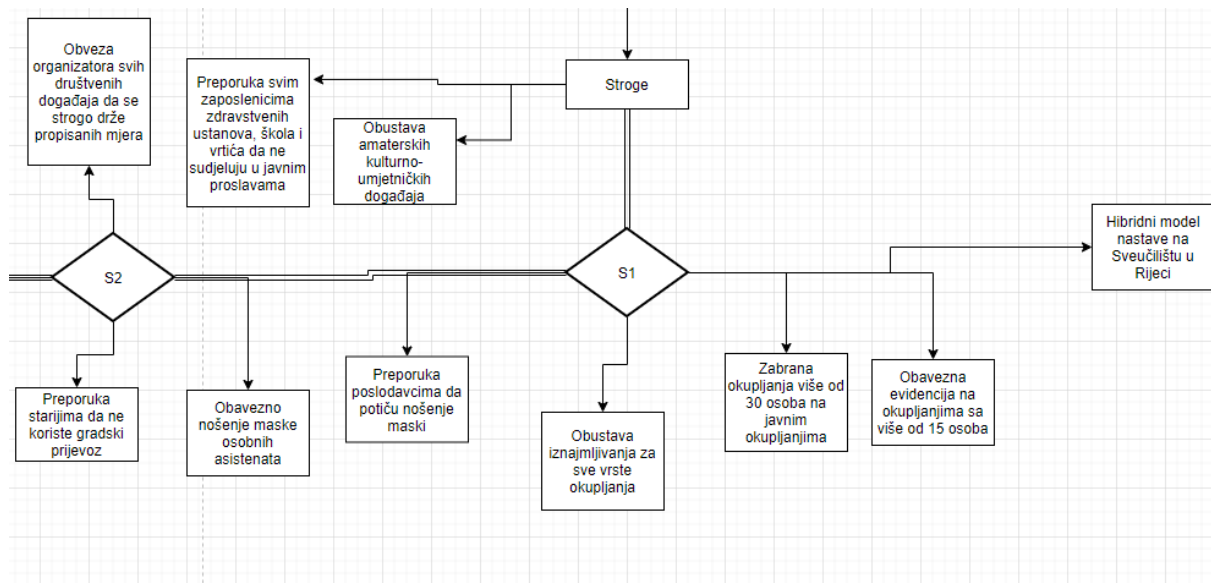


Slika 12 Prvi dio dijagrama Srednjih mjera

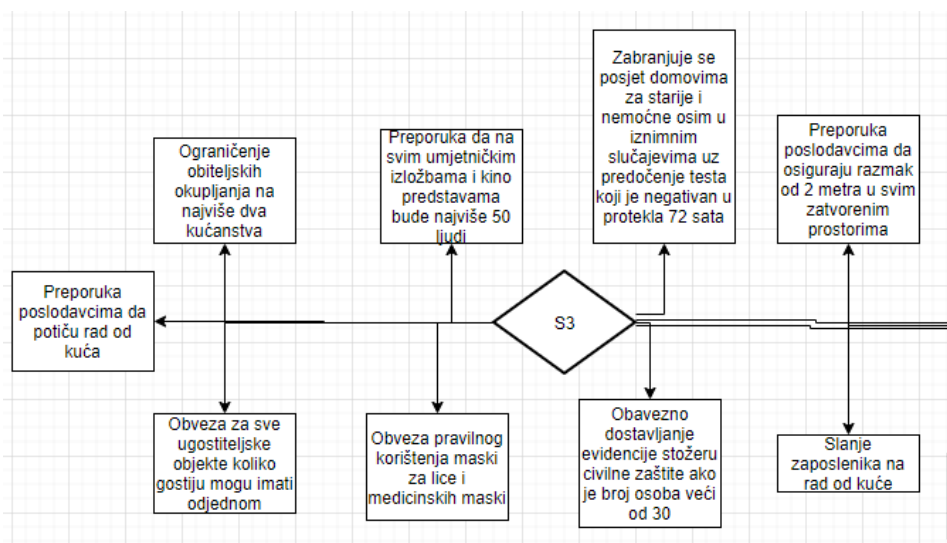


Slika 13 Drugi dio dijagrama Srednjih mjera

Veliki broj mjera je klasificiran u stroge mjere koje karakterizira zabrane i obveze. Cilj takvih mjera je spriječiti širenje virusa pod svaku cijenu. Takvih mjera ima dosta zbog toga što se donose u trenutku rasplamsavanja pandemije. U idućih nekoliko slika ću prikazati stroge mjere:

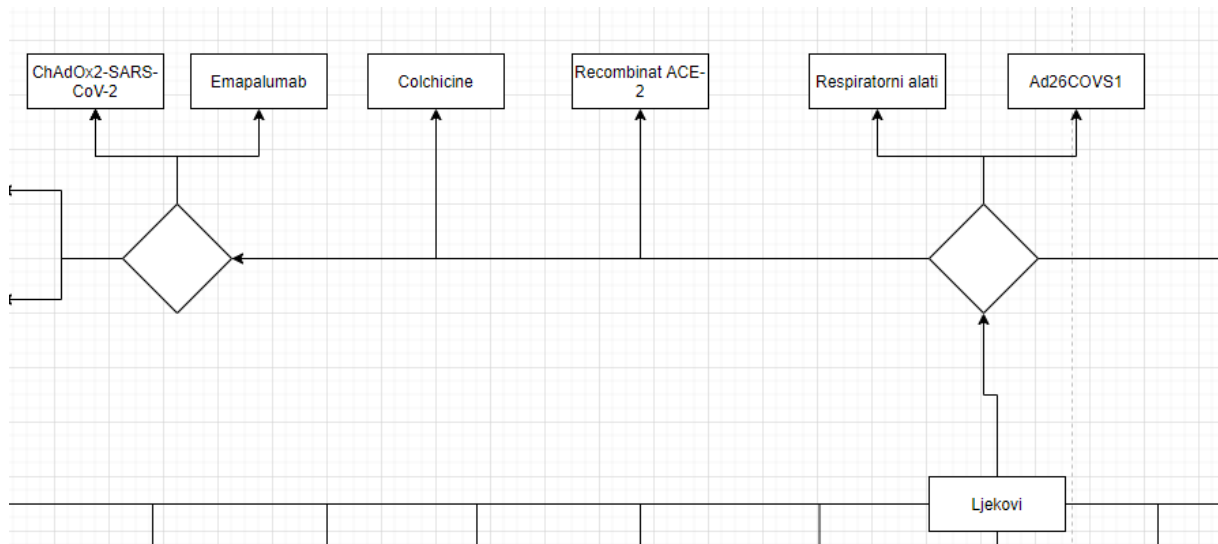


Slika 14 Prvi dio dijagrama Strogih mjera



Slika 15 Drugi dio dijagrama Strogih mjera

Nadalje ću prikazati dijagram lijekova. Obzirom da je broj lijekova iznimno velik nisam želio u dijagram stavljati dodatne attribute zbog jasnoće samog dijagrama. Sam dijagram se može napraviti na više načina ovisno o izvoru iz kojeg se dobijaju podaci jer su podaci različitih regija poput Europe, Amerike i Azije različiti:

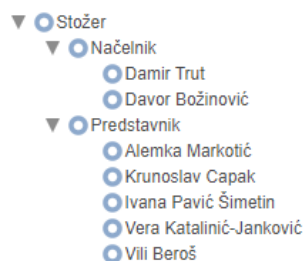


Slika 16 Dijagrama lijekova

4.3. Prikaz ontologije u alatu Protégé

Nakon što sam napravio konceptualni model krenuo sam u implementaciju elemenata u alatu Protégé. Zbog njegovog načina funkcioniranja puno je lakše prikazati veću količinu podataka u odnosu na konceptualni model. Zbog toga sam u Protégé stavio dodatne podatke koji opisuju pojmove. Za razliku od konceptualnog modela gdje su titule dio modela stožera u Protégé-u su izdvojene u zasebne klase. Također kao što sam spomenuo u objašnjenju konceptualnog modela lijekova u Protégé-u sam stavio sve attribute, koji opisuju lijekove poput potvrđenosti lijeka i drugog imena lijeka.

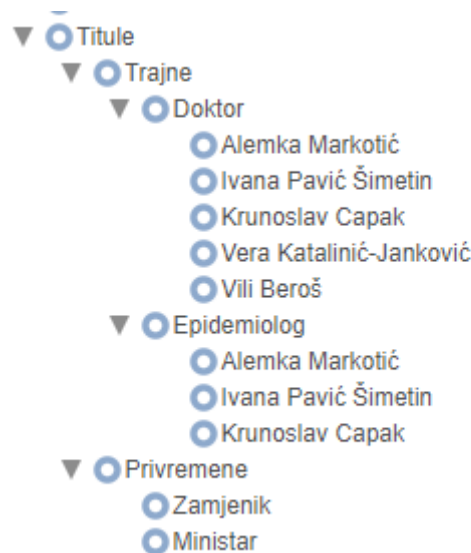
Za početak ću prikazati strukturu stožera koja se sastoji od načelnika i predstavnika:



Slika 17 Prikaz ontologije stožera

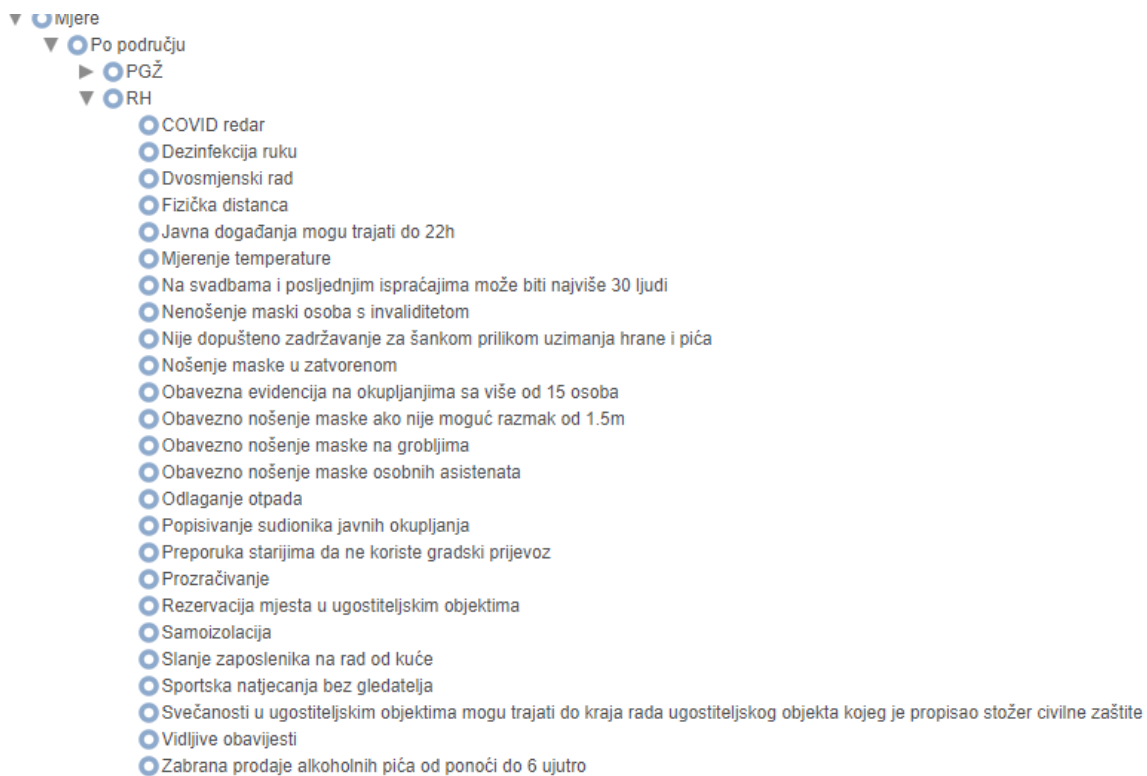
Nadalje ću prikazati ontologiju titula članova stožera u koje sam dodao dodatno klase privremenih i trajnih titula. Privremene titule su titule koje su vremenski određene poput ministra i zamjenika

ministra te mogu biti izgubljene u slučaju da se ministra ili zamjenika ministra ponovo ne izabere. Za razliku od privremenih trajne titule nemaju vremenskih tijek te su steknute akademskim zvanjem koje traje doživotno poput titule doktora i epidemiologa. Da bi dodatno naglasio razliku tih dviju vrsta titula odlučio sam trajne titule implementirati kao nadklase pojedinih članova dok su privremene klase implementirane kao relacije povezane s članom kojem pripadaju:



Slika 18 Prikaz ontologije titula

U nastavku ću prikazati ontologiju mjera koju sam podjelio na mjere po području i mjere po vrsti. Potom sam mjere po području podjelio na mjere na području Republike Hrvatske i mjere na području Primorsko-Goranske županije, dok se mjere po vrsti dijele na blage, srednje i stroge:



Slika 19 Mjere na području Republike Hrvatske

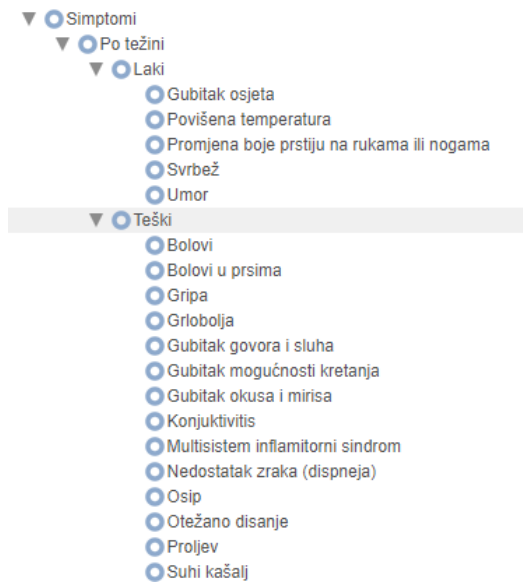
- ▼ Mjere
 - ▼ Po području
 - ▼ PGŽ
 - 1000 ljudi u otvorenom prostoru tijekom društvenih manifestacija
 - 500 ljudi u zatvorenom prostoru tijekom društvenih manifestacija
 - Hibridni model nastave na Sveučilištu u Rijeci
 - Obustava amaterskih kulturno-umjetničkih događaja
 - Obustava iznajmljivanja za sve vrste okupljanja
 - Obveza organizatora svih društvenih događaja da se strogo drže propisanih mjera
 - Obveza pravilnog korištenja maski za lice i medicinskih maski
 - Ograničenje obiteljskih okupljanja na najviše dva kućanstva
 - Preporuka da na svim umjetničkim izložbama i kino predstavama bude najviše 50 ljudi
 - Preporuka poslodavcima da osiguraju razmak od 2 metra u svim zatvorenim prostorima
 - Preporuka poslodavcima da potiču nošenje maski
 - Preporuka poslodavcima da potiču rad od kuća
 - Preporuka svim zaposlenicima zdravstvenih ustanova, škola i vrtića da ne sudjeluju u javnim proslavama
 - Pridržavanje dogovorenog termina pregleda
 - Svi administrativni poslovi se odvijaju elektronski
 - Zabrana korištenja bazena Kantrida
 - Zabrana okupljanja više od 30 osoba na javnim okupljanjima
 - Zabranjuje se posjet domovima za starije i nemoćne osim u iznimnim slučajevima uz predočenje testa koji je negativan u protekla 72 sata

Slika 20 Mjere na području Primorsko-Goranske županije

Nakon ontolgije mjera napravio sam ontologiju simptoma koje sam podjelio na simptome po učestalosti i simptome po težini pa sam potom podjelio simptome po učestalosti na česte i rijetke, dok sam simptome po težini podjelio na lake i teške. U nastavku ću prikazati prvo simptome po učestalosti potom simptome po težini:

- ▼ Po učestalosti
 - ▼ Rijetki
 - Bolovi
 - Grlobolja
 - Gubitak govora i sluha
 - Gubitak mogućnosti kretanja
 - Gubitak okusa i mirisa
 - Konjuktivitis
 - Multisistem inflamitorni sindrom
 - Osip
 - Proljevanje**
 - Promjena boje prstiju na rukama ili nogama
 - Svrbež
 - ▼ Česti
 - Bolovi u prsima
 - Gripa
 - Gubitak osjeta
 - Otežano disanje
 - Povišena temperatura
 - Suhu kašalj
 - Umor

Slika 21 Simptomi po učestalosti



Slika 22 Simptomi po težini

Nakon toga sam oblikovao ontologiju lijekova koju sam podjelio na četiri podkategorije, a to su ime lijeka, drugo ime lijeka, potvrđeni lijek i nepotvrđeni lijek. Pojedini lijekovi mogu i ne moraju imati drugo ime jer se drugo ime traži najviše u slučaju da je originalno ime koje dolazi od strane znanstvenika koji su pronašli lijek prekomplikirano za svakodnevnu uporabu. S druge strane jedan te isti lijek u različitim dijelovima svijeta može se imenovati različito što dovodi do više imena lijeka. Što se tiče potvrđenog ili nepotvrđenog lijeka stvar je kompleksnija jer se neki lijek može koristiti, a da ne bude službeno potvrđen od strane Svjetske Zdravstvene Organizacije. Svaka zemlja samostalno odlučuje o popisu lijekova koje će koristiti te pritom ne mora slijediti popis potvrđenih lijekova Svjetske Zdravstvene Organizacije. U nastavku ću ontologiju lijekove:

- AZD7442
- Acalabrutinib
- Ad26COVS1
- Ad5-nCOV
- Adjuvanted whole viron inactivated vaccine
- Anakinra
- Anti-SARS-COV-2 polyclonal hyperimmun immunoglobulin
- Antikolagulatori
- Apremilast
- BNT162b2
- Bakrov klorid
- Baricitinib
- Bemcentinib
- Bevacizumab
- Blood filtration system
- CD24Fc
- CHAG-COVID19-CA
- CT-P59
- CVnCov
- ChAdOx1-SARS-CoV-2
- ChAdOx2-SARS-CoV-2
- Colchicine
- Convalescent Plasma
- Coronavirus-like particle
- Cytokine Inhibitors
- Danoprevir
- Dexamethasone
- EIDD-2801
- Eculizumab
- Emapalumab
- FAV00B
- Favipiravir
- Fingolimod
- Gimsilumab
- Glukokortikosteroid
- Hydroxychloroquine
- Ibuprofen

Slika 23 Ontologija lijekova



Slika 24 Opis lijekova i sinonima

Što se tiče samih ontologija koje sam implementirao moglo bi ih se na drugi način implementirati. Primjerice, napraviti ontologiju sojeva koronavirusa te klase iz ontologije sojeva koristiti kao nadklase u ontologiji simptoma. Na takav način bi dobili jasan uvid u razvoj i nestanak određenih simptoma u određenom soju. Na isti način bi se moglo napraviti ontologiju lijekova koji bi pripadali određenom soju te preko zajedničkih nadklasa pokušati suziti korištenje određenih lijekova za određene simptome. Kod mjera mogla bi se postići dodatna granularnost kada bi se za svaku regiju posebno u Republici Hrvatskoj moglo popisati mjere i klasificirati ih u blage, srednje i stroge te bi se moglo bolje vidjeti kako mjere dijeluju na pandemiju. Na takav način bi se moglo vidjeti koje su mjere na području Republike Hrvatske najučinkovitije te stvoriti prioritete za njihovo provođenje. Na temelju liste najučinkovitijih mjera bi se moglo izraditi okvir prema kojem bi se mogle donositi učinkovitije mjere u slučaju da se pandemija pogorša. Da bi se navedene promjene mogle implementirati u sustav bilo bi potrebno automatizirati prikupljanje i spremanje velike količine podataka koja bi se potom koristila za implementaciju i modifikaciju ontologija. Način na koji bi se mogao implementirati sustav koji automatski prikuplja informacije o mjerama, simptomima, lijekovima i stožeru je uz pomoć Notion API-ja. Princip prema kojem bi se to moglo postići je definirati pojmove koji su potrebni za ontologiju i preko njih pretraživati Twitter ili neku drugu društvenu mrežu. Kada bi definirali pojmove mogli bi iskoristiti sustav za automatsko pokretanje zadataka tako da bi na dnevnoj bazi imali svježije podatke. Jedan od takvih sustava je Zapier preko kojeg se može povezati velik broj aplikacija u jedan sustav. Zapier je postao popularan kada je izašao Notion API te se preko njega odvijaju prikupljanja podataka. Nakon prikupljanja podataka bilo bi potrebno nove podatke procesirati i prilagoditi prikazivanju u ontologiji. Nadalje bi se mogla, za svaki novi prikupljeni podatak, definirati uloga tog podatka u ontologiji te bi se na takav način puno lakše mogla ažurirati ontologija s novim podacima.

5. Zaključak

U ovom diplomskom radu prezentirao sam što su ontologije bazirajući se na medicinskim ontologijama kao najznačajnijim varijantama ontologija. Opisujući ontologije prezentirao sam mogućnosti i važnost njihove primjene u prezentiranju znanstvenih činjenica. Dobro implementirane ontologije dovode do razumnijeg prikaza znanja i njihova najveća prednost je proširivost što znači da bilo koje novo znanje može biti lako prezentirano. Smatram da, ako se treba prezentirati novo znanje formalno, treba se koristiti ontologiju. Važno je reći da se za implementiranje kvalitetne ontologije prvo treba dobro definirati domena znanja te tako ograničiti koji podaci mogu biti u ontologijama. Nakon definicije same ontologije dao sam primjere pojedinih ontologija kao i područja njihove primjene te sam neke od njih pobliže opisao. Također sam dao prikaze i objašnjenja različitih vrsta ontologija kao i prikaze pojedinih povjesnih prikaza ontologija. Nadalje sam spomenuo alat Protégé i njegovu važnost u implementaciji ontologija. Njegova važnost konstantno raste zbog njegove arhitekture i mogućnosti prilikom implementacije ontologije. Zbog svoje otvorenosti u razvoju te konstantno porastu broja programera koji rade na njemu broj dodataka je u konstantnom porastu. Zahvaljujući tome eventualni problemi i nedostaci u aplikaciji se mogu puno brže riješiti. Protégé je u početku bio aplikacija za implementaciju medicinskih ontologija međutim sa porastom broja dodataka postaje sve prihvaćenije rješenje u mnogim granama prirodnih znanosti. Spomenuo sam umjetnu inteligenciju i procesiranje prirodnog jezika kao dva područja u sve jačem razvoju u kojim se sve više koriste ontologije i alat Protégé u kreiranju samog plana za izradu i implementaciju sustava. Dobar primjer su ekspertni sustavi za razgovor čija se implementacija kontrolira preko ontologije implementirane u Protégé-u. Nakon opisa alata sam samostalno izgenerirao potrebne klase za projekt. Izgenerirao sam šest klasa od kojih sam klasu titula dodatno izgenerirao kako bi prikazao odnos između člana stožera i njegove akademske titule ili pomoćne funkcije u stožeru. Nakon šest klasa za svaku klasu sam izgenerirao po dvije podklase za većinu klasa dok sam za klasu lijekova izgenerirao četiri klase. Na takav način sam dobio čvrstu strukturu u koju se vrlo lako može unijeti novo znanje. Primjerice bilo koju novu mjeru kojoj se odredi težina može se unijeti u sustav bez da se sama struktura mijenja. Na isti način sam definirao sve ostale klase tako da se samo unose podaci na najnižoj razini dok čitava struktura na višoj razini ostaje očuvana. Takav način rada može biti koristan u praćenju pandemije jer omogućuje efikasno i brzo prikupljane informacija o mjerama, simptomima, stožeru i lijekovima uključujući njihove sinonime. Važno je zapisivati sinonime jer se na takav način dobivaju potpunije informacije koje vrijede u bilo kojem trenutku i bilo gdje jer se uvijek zna u kojem lijeku se radi te se ljudi u slučaju potrebe za nekim lijekom mogu snaći koristeći njegovo ime ili njegov sinonim. Ontologija koju sam izradio mogla bi se lako proširivati tako da se pretvori u aplikaciju preko koje bi bilo puno lakše upisivati i čitati podatke te bi se povećala njezina dostupnost i olakšala njezina primjena. Osim toga moglo bi se provjeriti postoji li već definirana ontologija koja implementira slične podatke kao moja te vidjeti kako ta ontologija može poboljšati trenutno prikazane podatke. Kod same primjene korisnici ontologije bi mogli pretraživati ontologiju da bi dobili korisne informacije i upisivati vlastite informacije koje bi dodatno proširile njezin sadržaj te tako pomagati drugim korisnicima u informiranju. Te bi se ontologija mogla primjenjivati za brzi prijenos informacija između korisnika koji bi imali pristup svježim informacijama o COVID-19 pandemiji.

Slika 1 E.J.Lowe Ontologija četiri kategorije [1]	7
Slika 2 Ludwig Wittgstein Logika Traktatusa [2].....	8
Slika 3 Prikaz ravne i hijerarhijske ontologije[3].....	8
Slika 4 Prikaz Domenske ontologije [4]	9
Slika 5 Ontologija prema Nicoli Guarinu[5]	10
Slika 6 Slot editor [6]	15
Slika 7 Vizualizacija u Protégé-u [7].....	17
Slika 8 Prikaz dijagrama Nacionalnog stožera	19
Slika 9 Dijagrama simptoma po težini	20
Slika 10 Dijagram simptoma po učestalosti	20
Slika 11 Dijagram blagih mjera u Republici Hrvatskoj	21
Slika 12 Prvi dio dijagrama Srednjih mjera.....	22
Slika 13 Drugi dio dijagrama Srednjih mjera	22
Slika 14 Prvi dio dijagrama Strogih mjera.....	23
Slika 15 Drugi dio dijagrama Strogih mjera	23
Slika 16 Dijagrama lijekova	24
Slika 17 Prikaz ontologije stožera	24
Slika 18 Prikaz ontologije titula	25
Slika 19 Mjere na području Republike Hrvatske	25
Slika 20 Mjere na području Primorsko-Goranske županije	26
Slika 21 Simptomi po učestalosti.....	26
Slika 22 Simptomi po težini	27
Slika 23 Ontologija lijekova.....	28
Slika 24 Opis lijekova i sinonima.....	28

Bibliography

- Astghik Sargsyan, Alpha Tom Kodamullil, Shounak Bakshi, Johannes Darms, Sumit Madan, Stephan Gebel, Oliver Keminer, Geena Mariya Jose, Helena Balabin, Lauren Nicole DeLong, Manfred Kohler, Marc Jacobs, Martin Hofmann-Apitius. 2020. »The COVID-19 Ontology.« *Bioinformatics* 36 (24): 5703-5705.
- BioNCBI, Bioportal Nacionalnog Centra za Biotehnoške Informacije. n.d. *Human Disease Ontology*. <https://bioportal.bioontology.org/ontologies/DOID?p=classes&conceptid=DOID%3A0050211#visualization>.
- Gruber, Tom. 1992. *A Translation Approach to Portable Ontology Specifications*. Stanford, Kalifornija: Sveučilište Stanford.
- Legg, Catherine. 2007. »Ontologies on the Semantic Web.« *Phil Papers* 41: 407-451.
- Michigan, University of. n.d. *Biomedical Ontologies and Controlled Vocabularies*. <https://guides.lib.umich.edu/ontology/ontologies>.
- Munir, Kamran, Arjum, Sheraz. n.d. *The use of ontologies for effective knowledge modelling and information retrieval*. <https://www.sciencedirect.com/science/article/pii/S2210832717300649>.
- Nataly F. Noy, Deborah L. McGuinness. n.d. *Ontology Development 101: A Guid to Creating Your First Ontology*. Stanford University.
- NCBO, Nacionalni Centar za Biomedicinske Ontologije. n.d. *Nacionalni Centar za Biomedicinske Ontologije*.
- Rita Rawal, Kartik Goel, Charu Gupta. 2020. *COVID-19: Disease Pattern Study based on Semantic-Web Approach using Description Logic*. Bangluru, India: IEEE.
- Robert T. Neches, Angel R. Puerta, Henrik Eriksen, Pedro A. Szekely. 1996. *Toward Ontology-Based Frameworks for Knowledge-Acquisition Tools*. ResearchGate.
- Rubin. Daniel L., Noy, Natalya F., Musen, Mark A. n.d. *Protégé: A Tool for Managing and Using Terminology in Radiology Applications*. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2039856/>.
- Shane Babcock, Lindsay G. Cowell, John Beverly, Barry Smith. 2021. *The Infectious Disease Ontology in the Age of COVID-19*. Phil Papers. Pokušaj pristupa 17. July 2021. <https://philpapers.org/rec/BABTID>.
- Tošić Dušan, Milenković Nikola. n.d. *Building ontologies for different natural languages*. https://www.researchgate.net/figure/Relation-between-natural-language-NLP-and-ontology_fig1_270471292.