

Baza znanja temeljena na ontologijama

Petljak, Denis

Undergraduate thesis / Završni rad

2015

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:235:013922>

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

Download date / Datum preuzimanja: **2025-01-25**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Denis Petljak

Zagreb, 2015.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentori:

Prof. dr. sc. Bojan Jerbić, dipl. ing.
Dr. sc. Tomislav Stipančić, dipl. ing.

Student:

Denis Petljak

Zagreb, 2015.

Izjavljujem da sam ovaj rad izradio samostalno koristeći stečena znanja tijekom studija i navedenu literaturu.

Zahvaljujem mentorima prof.dr.sc. Bojanu Jerbiću i dr.sc. Tomislavu Stipančiću na stručnoj pomoći i razumijevanju tijekom izrade ovog rada.

Također zahvaljujem roditeljima Franju i Jadranki te sestri Gabrieli na pruženoj podršci i razumijevanju tijekom izrade ovog rada.

Denis Petljak



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za završne ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo
materijala i mehatronika i robotika

Sveučilište u Zagrebu	
Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

ZAVRŠNI ZADATAK

Student: **Denis Petljak** Mat. br.: 0035189610

Naslov rada na hrvatskom jeziku: **Baza znanja temeljena na ontologijama**

Naslov rada na engleskom jeziku: **Ontology based knowledge base**

Opis zadatka:

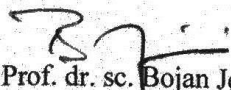
Ontologije se koriste najčešće u sklopu semantičkog weba za predstavljanje znanja o promatranoj domeni. Zasnivaju se na identifikaciji koncepata te na formiranju pripadajućih međurelacija. U ovisnosti o domeni interesa, ulaznim stanjima te implementiranim logičkim izrazima, ontologije omogućuju generiranje odgovora u skladu s očekivanjima eksperta koji je razvio sustav.

U radu je potrebno razviti ontologiju koja opisuje operacijsku salu kao domenu interesa te u sklopu toga identificirati pripadajuće koncepte koje je potom potrebno povezati odgovarajućim relacijama. Kreirana ontologija treba poslužiti kao temelj za integraciju robotskog sustava unutar operacijske sale.

Rad treba biti proveden i ispitan na opremi dostupnoj u Laboratoriju za projektiranje izradbenih i montažnih sustava, Fakulteta strojarstva i brodogradnje.

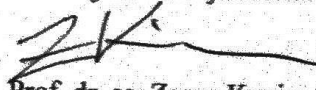
Zadatak zadan:
25. studenog 2014.

Zadatak zadao:


Prof. dr. sc. Bojan Jerbić

Rok predaje rada:
1. rok: 26. veljače 2015.
2. rok: 17. rujna 2015.

Predviđeni datumi obrane:
1. rok: 2., 3., i 4. ožujka 2015.
2. rok: 21., 22., i 23. rujna 2015.
Predsjednik Povjerenstva:


Prof. dr. sc. Zoran Kunica

Sadržaj:

POPIS SLIKA	II
POPIS TABLICA.....	III
POPIS KRATICA	IV
SAŽETAK.....	V
SUMMARY	VI
1 UVOD.....	1
2 Sveprisutni Internet.....	2
2.1 Uvod u Web tehnologije	3
2.1.1 Klijentska strana.....	3
2.1.1.1 HTML (HyperText Markup Language).....	3
2.1.1.2 JavaScript.....	4
2.1.1.3 CSS (Cascading Style Sheets)	4
2.1.1.4 DOM (Document Object Model).....	4
2.1.2 Poslužiteljska strana.....	5
2.1.2.1 Programski jezici s poslužiteljske strane	5
2.1.2.2 Baze podataka	6
2.1.2.3 Baze znanja	6
2.2 TCP/IP model.....	6
2.3 World Wide Web Consortium	8
2.4 Semantički Web	9
2.5 Struktura semantičkog <i>Web-a</i>	10
3 Ontologije.....	13
3.1 Vrste ontoloških jezika.....	13
3.2 Komponente OWL Ontologija.....	14
4 Robotika.....	16
4.1 Medicinski roboti	17
5 Ontologija operacijske sale.....	19
5.1 Software za izradu ontologija	21
5.2 Bayesove mreže	28
6 ZAKLJUČAK.....	30
LITERATURA.....	31
PRILOZI.....	32

POPIS SLIKA

Slika 1.	Koncept računalnih mreža.....	2
Slika 2.	Primjer jednostavne <i>HTML</i> stranice.....	3
Slika 3.	Razvoj <i>Web</i> -a	10
Slika 4.	Arhitektura semantičkog <i>Web</i> -a	12
Slika 5.	Grafički prikaz vrsta <i>OWL</i> -a.....	13
Slika 6.	Prikaz svojstava	15
Slika 7.	Prikaz klasa (Klase sadrže individue)	16
Slika 8.	Operacijska sala sa <i>RONNA</i> sustavom robota.....	18
Slika 9.	Grafički prikaz ontologije	20
Slika 10.	<i>Onto Graph</i>	21
Slika 11.	Izgled korisničkog sučelja <i>Fluent Editor</i> -a	22
Slika 12.	Generirani kod ontologije.....	23
Slika 13.	Klase korištene u ontologiji.....	24
Slika 14.	Definicija nekih klasa i relacija među njima.	24
Slika 15.	Prikaz definicije za <i>Zahvat1</i> u <i>Protege</i> software-u	25
Slika 16.	Primjer odgovora zaključivača za potrebne instrumente.	25
Slika 17.	Primjer odgovora zaključivača za upit o doktoru.....	26
Slika 18.	Primjer odgovora zaključivača za upit o instrumentima.	26
Slika 19.	Prikaz opsežnog odgovora zaključivača za jednostavan upit.....	26
Slika 20.	Primjer odgovora zaključivača sa 2 moguća ishoda.	27
Slika 21.	Izdvojeni dio ontologije.....	29

POPIS TABLICA

Tablica 1. OSI model..... 8

POPIS KRATICA

ARPANET	Advanced Research Project Agency NETwork
ASP	Active Server Pages
CSS	Cascading Style Sheets
DL	Descriptive Logic
DOM	Document Object Model
GUI	Graphical User Interface
HTML	HyperText Markup Language
IP	Internet protocol
JSP	JavaServer Pages
OWL	Ontology Web Language
PHP	Hypertext Preprocessor
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RIF	Rule Interchange Format
SPARQL	Simple Protocol and RDF Query Language
SQL	Structured Query Language
TCP	Transmission Control Protocol
URI	Uniform resource identifier
URL	Uniform Resource Locator
WWW	World Wide Web
XML	eXtensible Markup Language

SAŽETAK

Ontologije se koriste za oblikovanje znanja o nekoj domeni interesa. Ontologija opisuje pojmove u domeni interesa te međusobne odnose između pojmova. Različiti ontološki jezici pružaju različite sadržaje. Jedan od novijih standardnih jezika za ontologije je OWL (Ontology Web Language). OWL je baziran je na logičkom modelu koji omogućava definiranje i opisivanje pojmova. Kompleksni pojmovi stoga mogu biti konstruirani u definicije iz jednostavnijih pojmova.

U radu je razvijena ontologija operacijske sale i osoblja koja može poslužiti kao temelj za integraciju robotskog sustava unutar operacijske sale.

Ključne riječi: Ontologije, Baza znanja, Semantički Web

SUMMARY

Ontologies are used for knowledge shaping about some domain of interest. Ontology describes terms in the domain of interest and also mutual relations between these terms. Different ontology languages provide different content. One of newer ontology languages is OWL (Ontology Web Language). OWL is based on logic model which provides definition and description of term at once. Complex terms thus can be constructed in definitions from simple terms.

In this work operation room and personnel ontology is developed which can be used as base for robotic system implementation.

Key words: Ontology, Knowledge Base, Semantic Web

:

1 UVOD

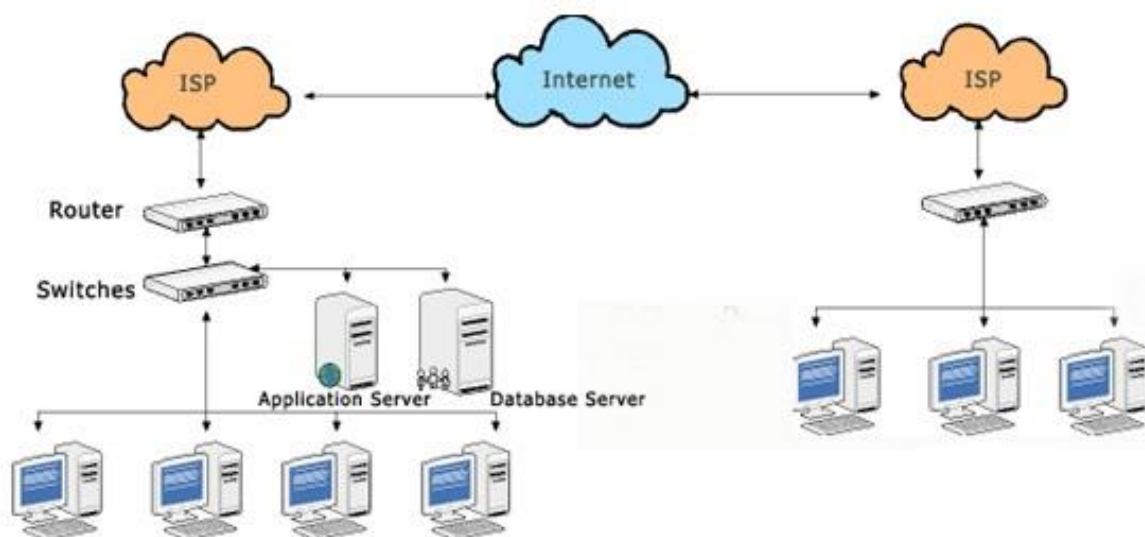
Semantika se kao grana lingvistike bavi proučavanjem značenja jezičnih znakova, dok s druge strane informacijske znanosti nisu usredotočene na značenje informacija. Informacijske znanosti promatraju informacije po strukturalističkom principu. Međutim područja umjetne inteligencije kao što su obradba prirodnog jezika, strojno učenje i strojno prevođenje, koja su ujedno i predmeti istraživanja informacijskih znanosti, zanima obradba prirodnog jezika na svim jezičnim razinama, pa tako i na semantičkoj. Semantički web ne zanima struktura poveznica među različitim mrežnim sjedištima, odnosno dokumentima (na temelju čega Google utvrđuje mjeru sličnosti i rang mrežnih stranica – PageRank, kao mjeru prema kojoj djelomično rangira rezultate pretraživanja), nego odnosi među elementima i njihovim svojstvima. Utvrđivanjem odnosa među elementima i njihovim svojstvima s pomoću metapodataka (npr. osoba, zanimanje, spol, datum rođenja, visina) omogućava se strukturiranje nestrukturiranih ili polustrukturiranih podataka na mreži. Semantički se web koristi malim jedinicama kako bi prezentirao informacije: subjektom, predikatom, objektom; objektom, ključem, vrijednošću; ID-om, glagolom, objektom. Vizija semantičkog weba u kratkoročnom bi razdoblju trebala uključivati "inteligentnije" pregledavanje i pretraživanje informacija, na temelju čega bi se moglo odrediti kojoj skupini pripada korisnik prema području interesa, kako bi mu se s obzirom na kontekst njegovih pregledavanja i pretraživanja isporučivao sličan sadržaj. Srednjoročno bi taj koncept trebao ponuditi rješenja za razvoj kontekstualnih aplikacija, odnosno aplikacija koje povezuju svijet računala sa svijetom ljudi, kako bi se poboljšala kvaliteta života. Dugoročno bi takav koncept trebao omogućiti razmjenu informacija među različitim uređajima te razvoj dijaloških sustava.

U radu će biti razvijena ontologija operacijske sale i osoblja koja će moći odgovarati na upite postavljene od strane korisnika i koja može poslužiti kao temelj za integraciju robotskog sustava unutar operacijske sale.

2 Sveprisutni Internet

Internet (često zvan *Web* ili *Net*) je svjetska rasprostranjena računalna mreža koja povezuje računala diljem svijeta. Korijeni Interneta sežu u kraj 60-ih godina prošlog stoljeća u SAD gdje se razvila mreža *ARPANET* (*Advanced Research Project Agency NETWORK*) financirana od strane američkog ministarstva obrane u svrhu povezivanja određenog broja računala. Prvotni cilj ARPANET-a bio je zaštita i dojava od nuklearnog napada za vrijeme hladnog rata, a u početku bila su povezana samo četiri računala.

Temelj komunikacije bio je i danas korišteni *TCP/IP* protokol. 1980.g. ministarstvo obrane SAD-a odlučilo je komercijalizirati Internet te je financiralo američke proizvođače računala da uključe *TCP/IP* u svoje mrežne protokole, a početkom 90-ih većina *ISP*-ova izgradila je svoje mreže za pristup Internetu, te je od tada Internet počeo naglo rasti. Najkorišteniji dio Interneta je *World Wide Web* koji koristi *hypertext* koji je metoda međureferenciranja, a reference vode na druge stranice i podatke na Internetu. Internet se proširuje u tolikoj mjeri da je napravljen *IPv6* standard adresiranja koja omogućuje još više spojenih uređaja od dosadašnjeg *IPv4* standarda.



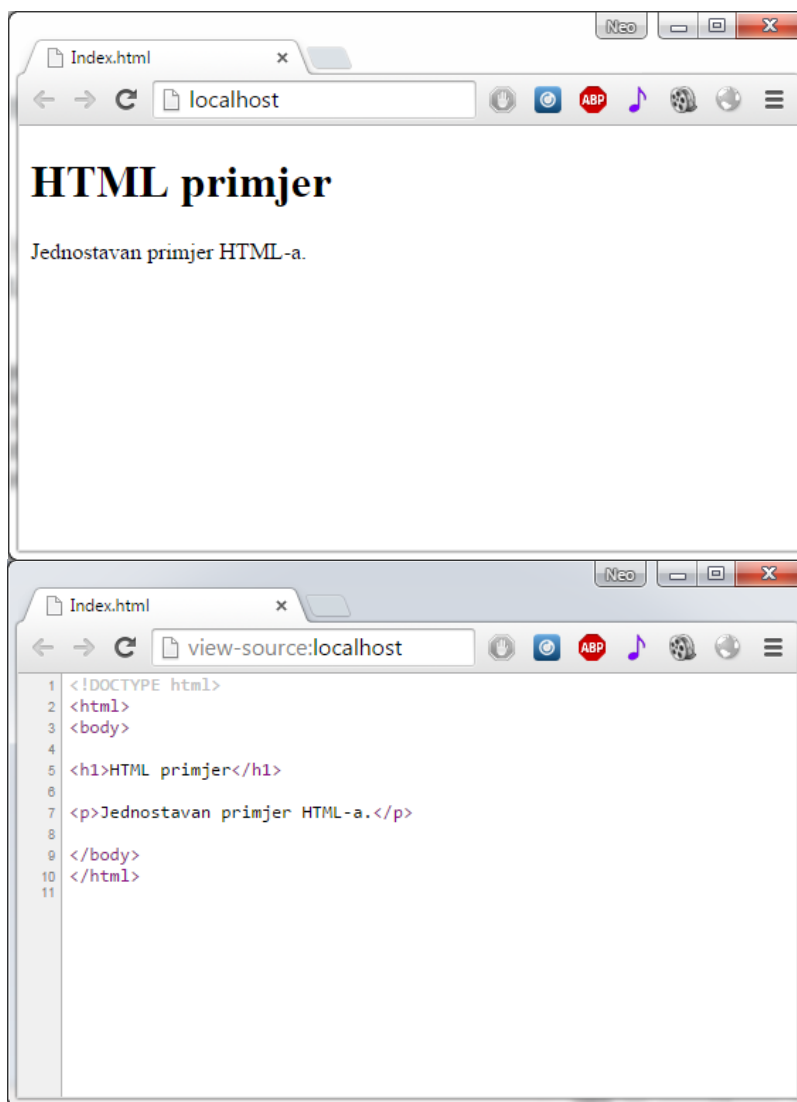
Slika 1. Koncept računalnih mreža

2.1 Uvod u Web tehnologije

2.1.1 Klijentska strana

2.1.1.1 HTML (HyperText Markup Language)

HTML je jezik za izradu web stranica čija je prva verzija *HTML 1.0* objavljena 1993. godine. te je u to vrijeme bio poprilično ograničen te se u njegove dokumente nisu mogle dodavati ni slike. Nakon njega slijedi daljnji razvoj *HTML-a 2*, *HTML-a 3* koji je predstavljen 1995. godine koji omogućava i definicije tablica, *HTML-a 4.0* i *4.01*. Sve do današnjeg *HTML-a 5.0* koji podržava reprodukciju videa bez potrebe za aplikacijama treće strane, *drag&drop* funkcionalnost te brojne nove elemente. Primjer *HTML* stranice prikazuje slika 2. gdje je gore prikazan izgled jednostavne stranice a dolje je prikazan njen *HTML* kod.



Slika 2. Primjer jednostavne *HTML* stranice.

2.1.1.2 JavaScript

JavaScript je programski jezik koji se izvršava na strani korisnika u njegovom web pregledniku. Jedan je od tri najkorištenije tehnologije za izradu Weba (Uz HTML i CSS) Sličan je Javi no nema mogućnosti objektnog programiranja. JavaScript nam omogućuje interaktivnost na web stranicama. Za razliku od HTML-a JavaScript je zadužen za izvršavanje zadataka (aritmetičke operacije, grafika,...), dok je HTML zadužen samo za definiranje sadržaja stranice.

2.1.1.3 CSS (Cascading Style Sheets)

CSS je stilski jezik čija je funkcija opisivanje izgleda stranice pisane u HTML-u ili nekom drugom markup language-u. Pomoću CSS-a opisujemo grafički izgled stranice, te je CSS prvenstveno napravljen da se omogući razdvajanje sadržaja dokumenata od njihovog izgleda i prezentacije kao što su boja, raspored, forma i fontovi. Ovakav način razdvajanja omogućuje pristupačnost sadržaja, pruža veću fleksibilnost sa dokumentom, omogućuje da više HTML stranica dijele formu stranice uz pomoć jedne CSS datoteke.

2.1.1.4 DOM (Document Object Model)

DOM je model za definiranje strukture HTML dokumenata. DOM odgovara na pitanja kao što su: koji sve tagovi postoje na stranici, na koji su način poredani, koliko ih ima, kako se oni prikazuju, ... Pomoću DOM-a izražava se struktura HTML dokumenata na univerzalan i sadržajno neovisan način.

2.1.2 Poslužiteljska strana

2.1.2.1 Programski jezici s poslužiteljske strane

PHP (Hypertext Preprocessor)

PHP je programski jezik orijentiran na *C* i *Pearl* sintaksu razvijen za razvoj web-a no koristi se kao alat za programiranje opće namjene. PHP kod se lako ugrađuje u HTML kod. PHP kod se izvršava na web poslužitelju (serveru). Nakon što se kod izvrši, poslužitelj šalje rezultat klijentu najčešće u obliku generirane web stranice. PHP može generirati i klijentu poslati HTML kod, sliku ili bilo koji drugi tip podataka.

ASP (Active Server Pages)

Za razliku od zahtjeva koji pošaljemo za dohvaćanjem HTML datoteke te nam poslužitelj odmah vrati željenu datoteku kada zatražimo ASP datoteku poslužitelj pregledava ASP datoteku i izvršava zadatke koji su u njoj zapisani te tek kada izvrši sve potrebne radnje vraća samo rezultat izvršavanja. Kod ASP-a korisnik ne može vidjeti izvorni kod stranice te koje se naredbe izvršavaju jer ASP datoteci pristup ima samo poslužitelj. ASP može pristupiti samo Microsoftovim bazama podataka (MS SQL i MS Access). ASP je kreiran od strane *Microsoft*-a te je kod baziran na *Visual Basic*-u.

JSP (JavaServer Pages)

JSP je sličan ASP-u samo što kod JSP-a sintaksa je bazirana na Javi budući da je kreirana od strane *Sun Microsystems*-a (tvoraca *Java*). Za razliku od ASP-a može se spojiti na bilo koju bazu podataka. JSP je besplatan za korištenje

2.1.2.2 Baze podataka

Baze podataka nam omogućuju dinamički sadržaj na web sjedištu. Za razliku od klasičnog HTML-a gdje je sav sadržaj definiran i nepromjenjiv za sve korisnike te nema mogućnosti prikazivanja samo dijelova koji su korisniku bitni i/ili zanimljivi. Baza podataka služi kao spremnik podataka na koje se odnose pravila. Klasičnim HTML-om bismo za svaku novonastalu promjenu koju je potrebno ažurirati morali kreirati novi statički HTML file koji bi vrijedio samo do sljedeće izmjene što bi bilo vrlo nepraktično. Neke od poznatijih baza podataka su: MS SQL , MS Access, Oracle, MySQL.

SQL (Structured Query Language)

SQL je računalni jezik za rad s bazama podataka koji omogućuje izradu, brisanje, ažuriranje i traženje podataka iz baza podataka. SQL se sastoji od jezika za definiranje podataka, jezika za manipulaciju podacima i jezika za nadzor podataka.

2.1.2.3 Baze znanja

Baza znanja (eng. *Knowledge Base*) sadrži znanje o domeni problema. U ekspertnim sustavima temeljenim na pravilima znanje je prikazano skupom pravila. Svako pravilo definira relacije, prijedloge, smjernice, strategije ili iskustva, koristeći IF (uvjet) THEN (djelovanje) strukturu. Ako je uvjet ispunjen, pravilo aktivira djelovanje odnosno niz posljedičnih akcija i zaključaka.

2.2 TCP/IP model

TCP/IP je oznaka za grupe protokola koja se često naziva i *IP* grupa protokola. Naziv *TCP* potječe od dva najvažnija protokola te skupine *TCP* (eng. *Transmission Control Protocol*) te prema samom *IP* protokolu. *TCP/IP* omogućava povezivost među mrežnim čvorovima specificirajući kako se podaci moraju konvertirati u pakete, adresirati, slati, prenositi te primiti na odredištu. *TCP/IP* spada u treći i četvrti sloj OSI referentnog modela. OSI je najkorišteniji apstraktni opis arhitekture mreže. Opisuje komunikaciju hardvera,

programa, softvera i protokola pri mrežni komunikacijama. Koriste ga proizvođači pri projektiranu mreža, kao i stručnjaci pri proučavanju mreža.

Fizički sloj definira električka i fizička svojstva mrežnih uređaja. Također definira naponske nivoe, brojeve pinova te uređaje kao što su ponavljači, mrežni koncentratori itd.

Podatkovni sloj brine se za razmjenu podataka između mrežnih uređaja i za detekciju/korekciju možebitnih grešaka na fizičkom sloju.

Mrežni sloj je zadužen za pretvaranje logičke IP adrese u fizičku MAC adresu kao bi podaci stigli od jednog mrežnog čvora do drugog.

Transportni sloj vodi računa o paketima koji putuju između dva računala. Primjeri protokola na transportnom sloju su TCP i UDP. Ako se neki paket "zagubi" na putu, TCP će tražiti da se ponovo pošalje.

Sloj sesije bavi se uspostavom veze između krajnjih korisnika, i sinkronizacijom iste.

Prezentacijski sloj služi za izvođenje kodiranja za sustav koji koristimo npr. TXT-datoteke na Mac OS-u, Unixu i Windowsima na različite načine označavaju prelazak u novi red. Prezentacijski sloj zadužen je za usklađivanje kodiranja.

Aplikacijski sloj je sloj OSI modela najbliži korisniku. On pruža mrežne usluge korisničkim aplikacijama. Od ostalih slojeva OSI modela razlikuje se po tome što ne pruža usluge drugim slojevima, već samo aplikacijama van OSI modela.

OSI model
7. Aplikacijski sloj
6. Prezentacijski sloj
5. Sloj sesije
4. Transportni sloj
3. Mrežni sloj
2. Podatkovni sloj
1. Fizički sloj

Tablica 1. OSI model

2.3 World Wide Web Consortium

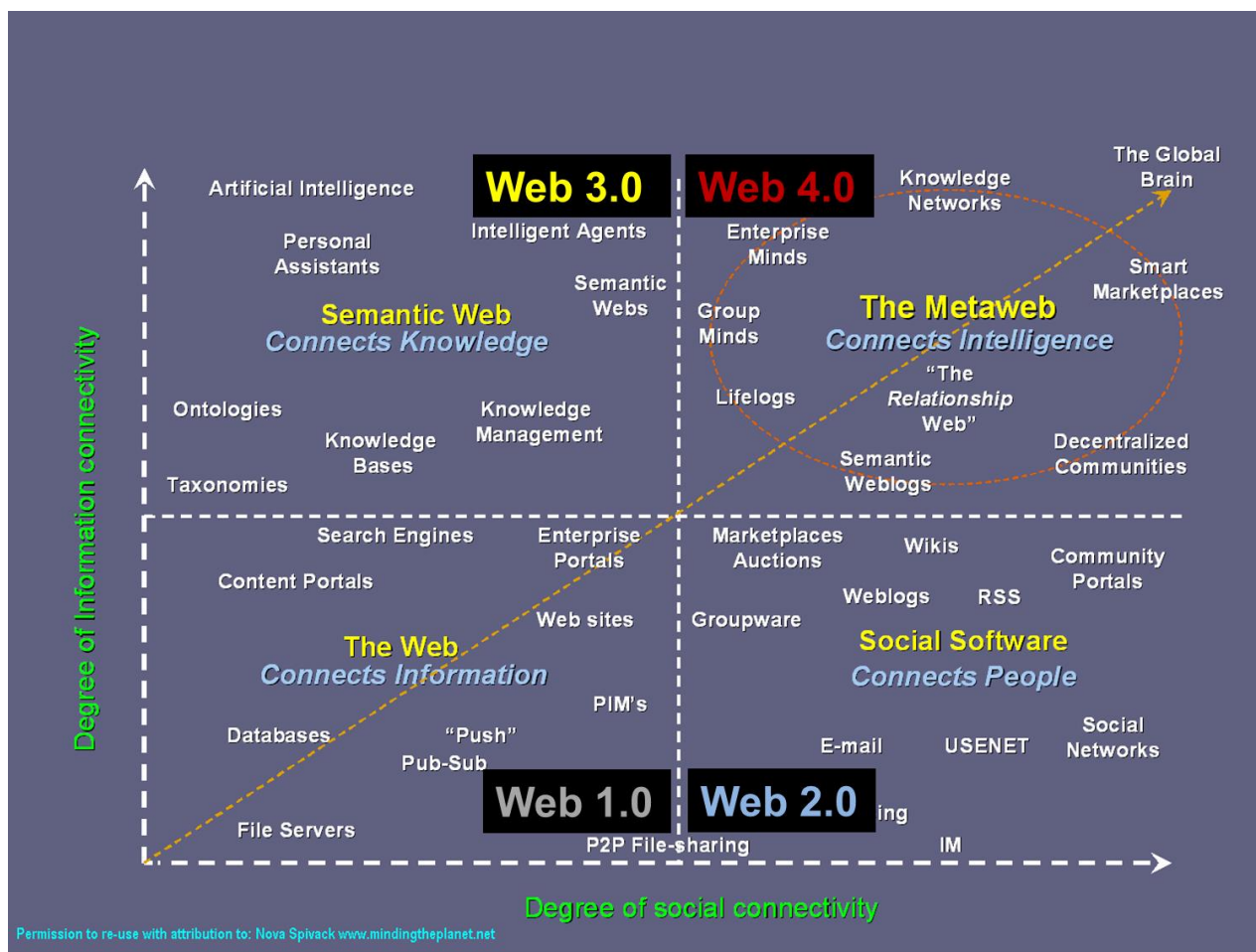
World Wide Web Consortium poznatija kao *W3C* (<http://www.w3.org/>) je organizacija koja se bavi standardizacijom tehnologija korištenih na webu. Osnovana je u listopadu 1994. godine u suradnji Massachusetts Institute of Technology (MIT) i Europske organizacije za nuklearna istraživanja (CERN). Inicijator osnivanja je autor *Web*-a Tim Berners-Lee. *W3C* djeluje kroz radne grupe te kreira i održava *WWW* standarde koji se nazivaju *W3C preporuke* (*W3C Recommendations*). Neko od poznatijih standarda *W3C*-a su:

- *HTML*
- *CSS*
- *XML*
- *SPARQL*
- *RDF*
- *OWL*

Te brojni drugi.

2.4 Semantički Web

Širenjem Interneta raste i količina podataka koja se na njemu nalazi. Ako imamo podatke ali ih ne možemo pronaći, Internet nam nije od velike koristi. Većina današnjeg weba sintaktičkog je karaktera, što znači da su pisane jezikom razumljivim ljudima, te računala iz njih mogu dohvatiti samo riječi. Semantički je web s druge strane strukturiran tako da se iz njega može izvući i značenje osim samog konteksta stranice. Zamislimo sljedeći primjer: Imamo web sjedište koje nam daje informaciju o vremenu za svaki grad na svijetu u *HTML* formi. Iako nam ta stranica pruža dinamički način prikaza informacija, ona je opisana sintaktično. Želimo li tu stranicu iskoristiti za dohvaćanje podataka (eng. *web scraping*) npr. za vremensku stanicu ili mobilnu aplikaciju koja je prilagođena našim potrebama lako ćemo to i postići. No ako se developeri te stranice odluče na promjenu izgleda stranice ili samog njezinog koda, naša aplikacija će se morati ponovo programirati kako bi se uskladila sa formatom novo izrađene stranice. Kada bi ta stranica bila napisana semantički, naš program bi i dalje mogao (ako je tako programiran) dohvaćati nama potrebne informacije, neovisno o samom izgledu stranice i mjestu elemenata. Posljednjih godina radi se na prilagodbi starih i nadolazećih podataka prema standardima *W3*-a koji uključuju uključivanje semantičkog weba u svrhu postizanja organiziranosti nad sadašnjom neorganiziranošću. Ideja za razvoj semantičkog weba potječe od izumitelja *World Wide Web*-a Tim Berners – Lee-a koji ga je definirao kao mrežu podataka koje računala (ili bilo koji mrežni uređaj) može razumjeti. Budući da se napredak računalnih i mrežnih tehnologija, razvoj *Internet of Things*-a, *SMART*-a i ostalih nadolazećih tehnologija temelji na umrežavanju biti će ključno da računalo dohvati potrebne i što je još važnije ispravne podatke (neispravan podatak nam pomaže kao i da ga nemamo, a u nekim slučajevima ishod može biti gori nego da ga nemamo) potpuno samostalno bez utjecaja čovjeka. Na slici 2. prikazan je napredak *Web*-a od *1.0* do *4.0*



Slika 3. Razvoj Web-a

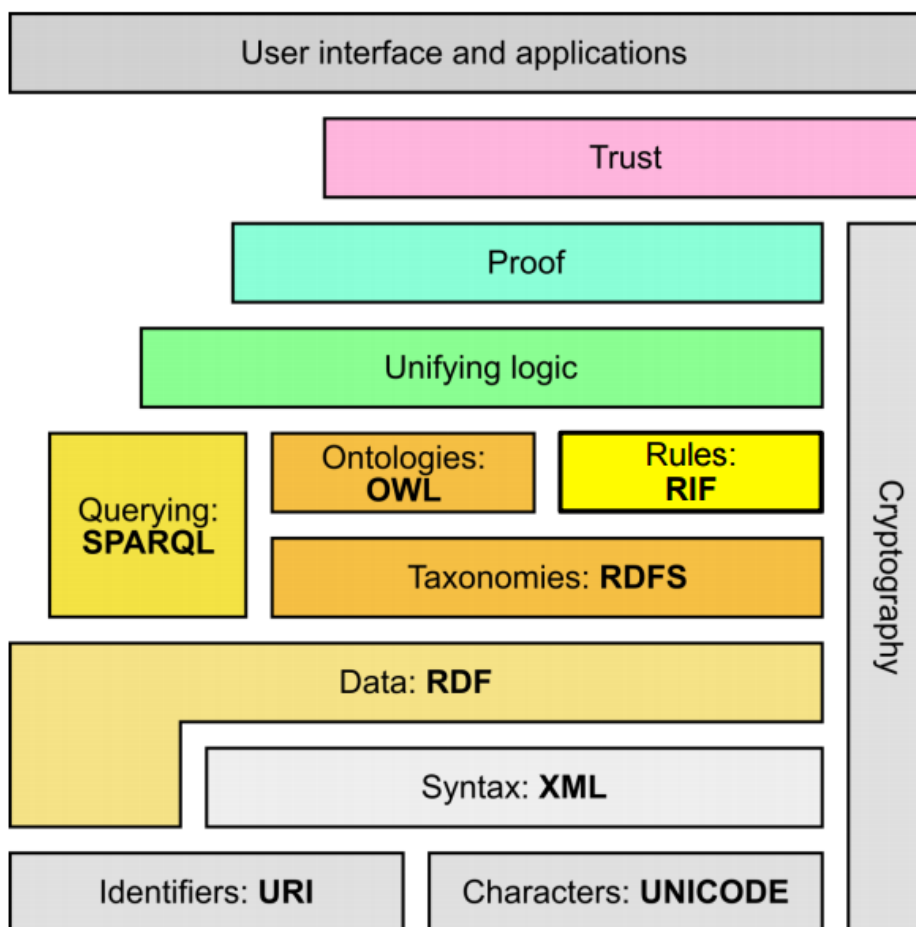
2.5 Struktura semantičkog Web-a

Kako bi struktura i značenje Web-a bili razumljivi krajnjim korisnicima i *software-u* za dohvaćanje podataka potrebno je na neki način standardizirati *metapodatke*¹ (eng. *metadata*) koji će omogućiti pretraživaču pronalazak željenih sadržaja. Kako *HTML* (*HyperText Markup Language*) opisuje kako će vizualno stranica izgledati prema korisniku (npr. Boja, font, veličina teksta, smještaj i veličina slika,...), semantički Web koristi jedan od dva standarda. Prvi je *XML* (*eXtensible Markup Language*). XML je kako i njegov naziv govori proširiv jezik za označavanje. XML nikako nije zamjena za *HTML* već je on kreiran da opiše što podaci znače, a ne kako izgledaju (što je slučaj *HTML-a*). Kod *XML-a* kreator Web sadržaja može kreirati svoje *Tag*-ove te nije ograničen na preddefinirane termine koji su definirani

¹ metapodaci su podaci o podacima

samim jezikom. Drugi standard je *RDF (Resource Description Framework)* koji se koristi za opisivanje resursa *Web* stranica i njihovog sadržaja, namijenjen je da ga čitaju računala. *RDF* je pisan u *XML*-u. *RDF* opis može sadržavati imena autora traženog resursa, datum kreiranja ili osvježavanja, organizaciju stranica, informacije o sadržaju, ključne riječi i mnoge druge informacije. Sve navedeno u svrhu postizanja da software razumije sadržaj te iz toga pruža kvalitetniju informaciju na temelju naših upita. Semantički *Web* ima definiranu strukturu koja se sastoji od nekoliko slojeva prikazanih na slici 4.

Prvi sloj *URI* i *UNICODE* slijedi važne značajke postojećeg *WWW*-a. *UNICODE* je standard kodiranja internacionalnih znakova te omogućava da se svi svjetski jezici mogu koristiti (čitati i pisati) na *Web*-u koristeći samo jednu standardiziranu formu. *URI* nam omogućava jedinstveno identificiranje resursa. Podset *URI*-a je *URL* koji sadrži mehanizme pristupa i lokaciju dokumenta kao na primjer <https://www.fsb.unizg.hr/> . Sljedeće redom idu već ranije navedeni *XML* i *RDF* standardi. Nakon njih slijedi *RDFS*, *SPARQL*, *OWL* i *RIF*. *RDFS (RDF Schema)* se koristi za opisivanja taksonomija klasa i svojstava te ih koristiti kao ontologije nižeg stupnja (Ontologije, klase i svojstva objašnjene su u poglavlju 3). Detaljnije ontologije opisuju se *OWL*-om (*Ontology Web Language*).



Slika 4. Arhitektura semantičkog Web-a

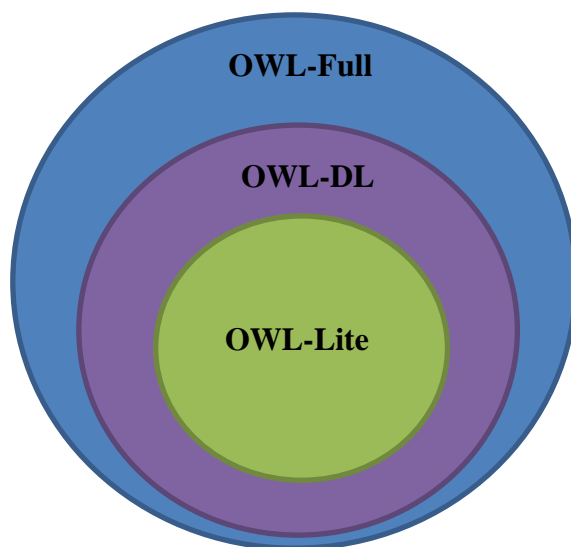
OWL je jezik nastao iz deskriptivne logike te pruža više opisa od *RDFS*-a. *OWL* se dijeli na više kategorija kako je objašnjeno u poglavlju 3.1. *SPARQL* služi za dohvaćanje upita i vraćanje rezultata upitu, *SPARQL* može pristupiti i *RDF* podacima. *RIF* (*Rule Interchange Format*) u sebi sadrži jezična pravila povezivanja. *RDFS*, *SPARQL*, *OWL* i *RIF* zajedno su objedinjeni u kategoriju Logike. *Proof* sloj zadužen je za objašnjenje dobivenih rezultata i podrijetlo podataka. *Trust* ili sloj povjerenja je dokaz da sustav radi ispravno te objasniti što radi. Zadnji sloj je zadužen za prezentaciju rezultata korisniku ili software-u koji je poslao zahtjev.

3 Ontologije

Ontologije se koriste za oblikovanje znanja o nekoj domeni interesa. Ontologija opisuje pojmove u domeni interesa te međusobne odnose između pojmova. Različiti ontološki jezici pružaju različite sadržaje. Najnoviji razvoj standardnih jezika ontologije je OWL (Ontology Web Language). OWL ima veći set operatora (npr. I, ILLI i negacija). Baziran je na logičkom modelu koji omogućava da pojam bude ujedno definiran te opisan. Kompleksni pojmovi stoga mogu biti izgrađeni u definicije iz jednostavnijih pojmova.

3.1 Vrste ontoloških jezika

Ontološki jezici mogu se kategorizirati u tri vrste ili podvrste jezika: OWL-Lite, OWL-DL i OWL-Full. Karakterizirajuća značajka svake podvrste ontološkog jezika je njegova izražajnost. OWL-Lite je najmanje izražajna podvrsta jezika. OWL-Full je najizražajnija podvrsta OWL-a. Izražajnost OWL-DL podvrste jezika spada između jezika OWL-Lite i OWL-Full. OWL-DL se može smatrati ekstenzijom OWL-Lite-a, a OWL-Full kao ekstenzija OWL-DL.



Slika 5. Grafički prikaz vrsta OWL-a

OWL-Lite

OWL-Lite je sintaktički najjednostavnija podvrsta jezika. Njegova namjena je korištenje u situacijama gdje postoje jednostavne hijerarhije pojmova te su potrebne jednostavne veze između pojmova. Kao primjer, predviđeno je da će OWL-Lite pružiti najbrži put migracije za postojeće leksikone i ostale jednostavne pojmovne hijerarhije.

OWL-DL

OWL-DL je puno elokventniji od OWL-Lite-a i baziran je na deskriptivnoj logici (otkud i nastavak DL). Opisna logika je odlučujući fragment logike prvog reda te je kao takva podložna automatskom zaključivanju. Zbog toga, moguće je automatski izraditi klasifikacijsku hijerarhiju i provjeriti dosljednost ontologije prema OWL-DL-u.

OWL-Full

OWL-Full je najelokventnija podvrsta OWL jezika. Namijenjen je korištenju u situacijama gdje je bitniji vrlo visok stupanj izražajnosti nego zajamčenost računske cjelovitosti samog jezika. Automatska zaključivanja nisu moguća u OWL-Full ontologiji. Vrijeme potrebno za obradu modela na osnovi ovog jezika je dulje nego na ostalim te se koristi samo u zahtjevnim primjenama.

3.2 Komponente OWL Ontologija

OWL ontologija se sastoji od individua (Individuals), svojstava (Properties) i klasa (Classes)

Individue (Individuals)

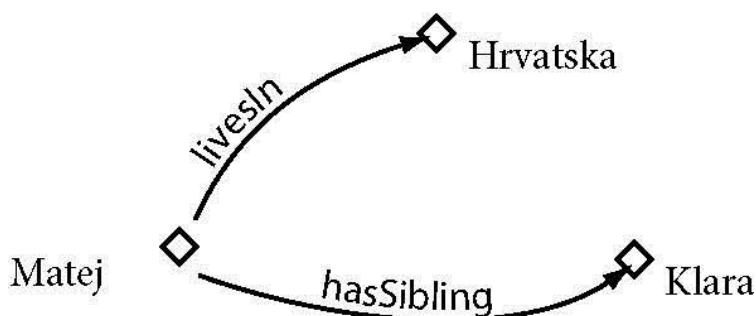
Individue prezentiraju objekte u domeni interesa. Bitno je naglasiti da OWL ne koristi pretpostavku jedinstvenog imena. Što znači da se dva različita imena mogu odnositi na istog

pojedince, npr. „Kraljica Elizabeta“, „Kraljica“, „Elizabeth Windsor“ se mogu odnositi na istu osobu. U OWL-u mora biti točno naglašeno da su individue iste, ili različite jedni drugima.

Individue se još zovu i instance (Instances). Individue se mogu protumačiti kao „instance klasa“.

Svojstva (Properties)

Svojstva su binarne relacije² na individuama. Za primjer, svojstva (Properties) povezuju zajedno dvije individue, kao npr. svojstvo **imaBraću** može povezivati individu **Matej** sa individuom **Klara**, ili svojstvo **imaDijete** može povezivati individu **Petar** sa individuom **Matej**. Svojstva mogu imati inverz, kao na primjer, inverz od **imaDijete** je **jeDijeteOd**. Svojstva mogu biti ograničena na samo jednu vrijednost, npr. biti funkcionalne (Functional). Također mogu biti tranzicijske (Transitive) ili simetrične (Symmetric)



Slika 6. Prikaz svojstava

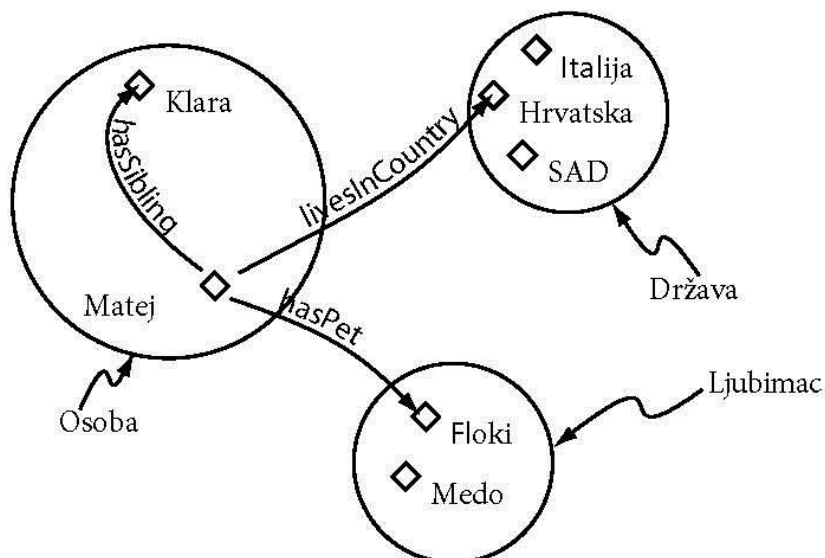
Klase (Classes)

OWL klase se interpretiraju kao društvo koje se sastoji od pojedinaca. Opisane su formalnim (matematičkim) opisima koji precizno navode zahtjeve za pripadanje klasi. Kao primjer klasa **Mačka** će sadržavati sve individue koji su mačke u našoj domeni interesa³. Klase mogu biti organizirane u hijerarhiju Superklasa-Subklasa koja je poznata pod imenom taksonomija (Taxonomy). Subklase se nalaze unutar superklasa. Zamislimo klasa **Životinja** i

² Binarna relacija je relacija između dvije stvari

³ Individue mogu pripadati u više klasa

Mačka. **Mačka** je subklasa superklase **Životinja**. Što ukazuje na tvrdnje „Sve su mačke životinje“, „Svi pripadnici klase **Mačka** su članovi klase **Životinja**“, „Ako je nešto **Mačka** povlači za sobom tvrdnju da je i **Životinja**“



Slika 7. Prikaz klasa (Klase sadrže individue)

4 Robotika

Robotika je grana znanosti koja predstavlja sinergiju strojarstva, elektrotehnike te računalnih znanosti. Bavi se dizajnom, konstrukcijom i primjenom robota, kao i računalnim sustavima za njihovu kontrolu, povratnu vezu od senzora i procesiranjem informacija. Ove se tehnologije bave automatiziranim strojevima koji mogu zamijeniti ljude u opasnim okruženjima, proizvodnim procesima gdje se zahtjeva težak i monoton rad a mogu i ne moraju nalikovati ljudima u smislu izgleda i ponašanja. Danas je robotika brzo rastuće znanstveno polje. Kako znanost napreduje. Istraživanje dizajn i gradnja novih robota pronalazi ulogu u civilnoj, komercijalnoj i vojnoj upotrebi. Da bi robot mogao obavljati svoju zadaću potrebno je nekoliko komponenti koje su zajedničke gotovo svim robotima:

- izvor napajanja
- aktuatore
 - elektromotori

- linearni aktuatori
- pneumatski mišići
- ...
- senzore
 - senzor sile
 - senzor momenta
 - vizijske senzori
 - ...
- naprave za manipulaciju
 - mehaničke hvataljke
 - pneumatske hvataljke
 - ...
- naprave za kretanje ili pozicioniranje
 - kotači
 - gusjenice
 - fiksno postolje
 - ...

4.1 Medicinski roboti

U ovom radu razviti će se ontologija za opis operacijske sale za integraciju robotskog sustava u kojem je korišten medicinski robot. Medicinski roboti dijele se na tri tipa:

- **Operacijski roboti** – ovakvi roboti ili daju veću preciznost kirurškim zahvatima od one postignute ljudskom rukom ili omogućuju daljinsku operaciju kada izvođač zahvata nije fizikalno prisutan kod pacijenta.
- **Rehabilitacijski roboti** – ova skupina robota namijenjena je rehabilitaciji ozlijeđenih ili starijih ljudi ili ljudi sa disfunkcijom dijelova tijela koje uzrokuju probleme pri kretanju

- **Bioroboti** – ova skupina robota napravljena je da imitira ponašanje ljudi i životinja

Roboti integrirani u operacijsku salu opisanu ovim radom spada u skupinu operacijskih robota pod nazivom *RONNAmaster* i *RONNAassistant*. *RONNAmaster* standardni je robot sa 6 stupnjeva slobode gibanja, dosegom od 900 mm, opteretivošću od 6 kg te ponovljivosti od $\pm 0,03$ mm. *RONNAmaster* je robot sa velikom krutošću i zbog toga se koristi za precizno vođenje instrumenata kirurga ili robota asistenta prema određenoj točki operacije. *RONNAassistant* je također standardni robot ali sa 7 stupnjeva slobode gibanja koji je opremljen sa sensorima zakretnog momenta koji omogućuju laganu interaktivno ponašanje robotske ruke i podešavanje potrebnog momenta. Robot je programiran na način da sprečava mogućnost ljudske ozljede. Slika 8. prikazuje operacijsku salu i *RONNA* sustav.



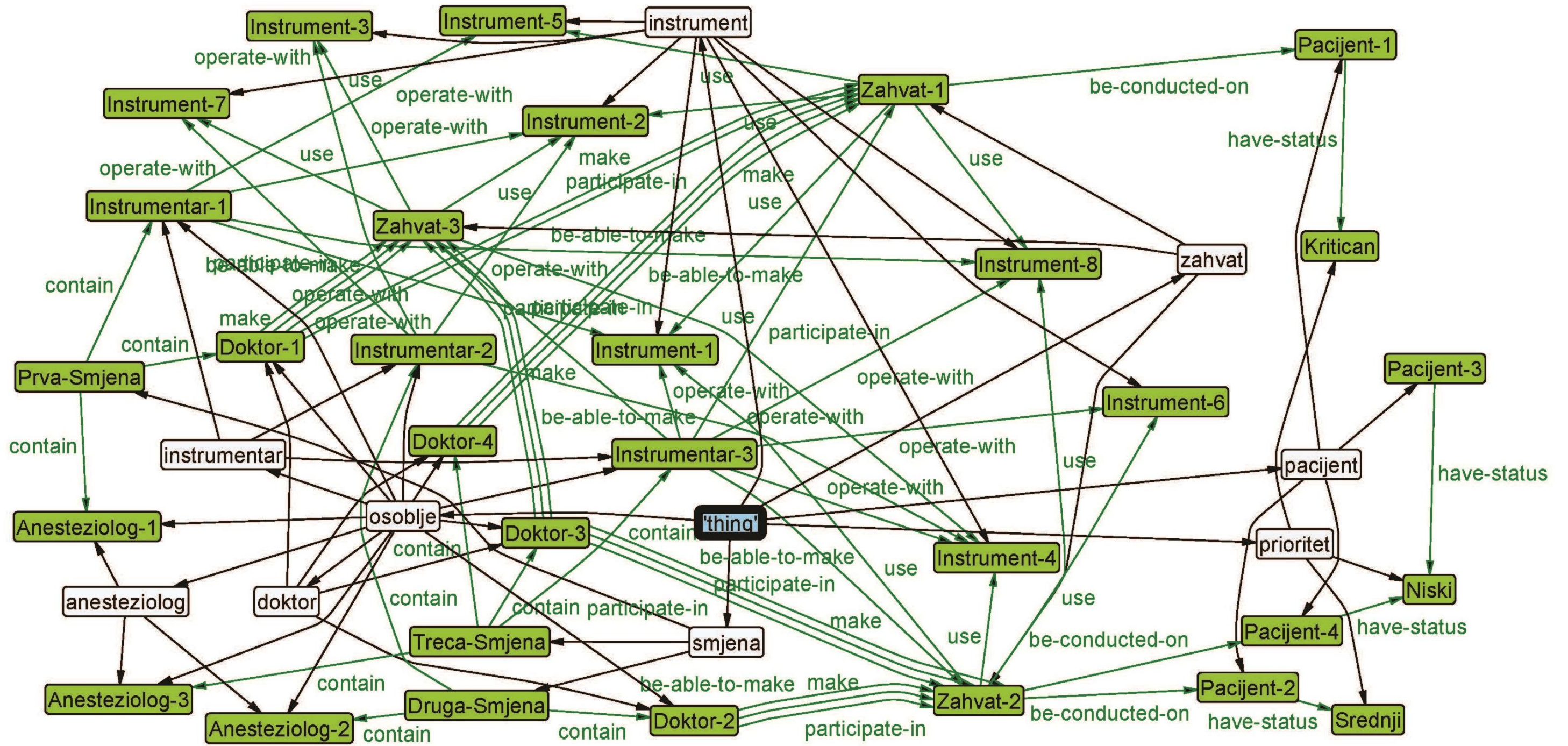
Slika 8. Operacijska sala sa *RONNA* sustavom robota

Na gornjoj slici prikazana je operacijska sala sa *RONNA* sustavom robota *RONNAmaster* – desno i *RONNAassistant* – lijevo, sa operacijskim stolom i modelom pacijentove glave koja je objekt operacije

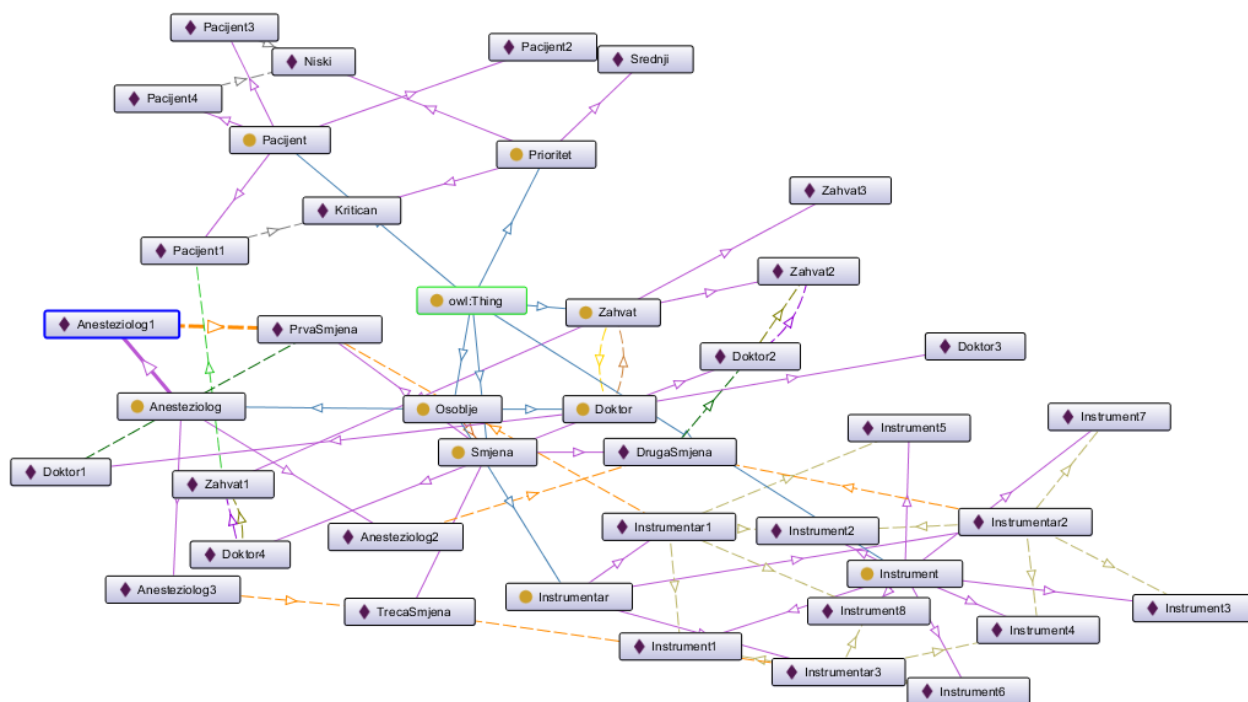
5 Ontologija operacijske sale

Kako ontologije imaju svrhu predstavljanja znanja za promatranu domenu interesa, domena kao takva mora se opisati, navesti komponente koje ona sadrži te definirati ovisnosti među njima. Operacijska sala sastoji se od: osoblja koje se sastoji od doktora, anesteziologa i instrumentara, instrumenata koji se koriste za različite zahvate, zahvata koji se izvode, prioriteta koji se dodjeljuju pojedinim zahvatima te smjena. Isto tako potrebno je i navesti relacije koje povezuju pojedine komponente ontologije. Budući da je program koji koristim za opis ontologije na engleskom jeziku te je gramatički osjetljiv, relacije će biti pisane na engleskom jeziku. Neke od relacija su: *conduct*, *be-able-to-make*, *operate-with*, *use*, ... Grafički prikaz ontologije dan je na slici 9. Kada se ontologija izveze u Protege (program koji nam pruža grafički prikaz relacija unutar ontologije tzv. *Onto Graph*.) možemo dobiti vizualno ljepši prikaz ontologije kao na slici 10. Slike 9. i 10 prikazuju kompleksnost veza između pojedinih komponenata.

Kada se ontologija izveze u Protege (program koji nam pruža grafički prikaz relacija unutar ontologije tzv. *Onto Graph*.) možemo dobiti vizualno ljepši prikaz ontologije kao na slici 10. Slike 9. i 10 prikazuju kompleksnost veza između pojedinih komponenata. Svrha izrade ontologija je kreiranje baze znanja za operacijsku salu koji će se na temelju upita moći davati odgovarajuće odgovore.

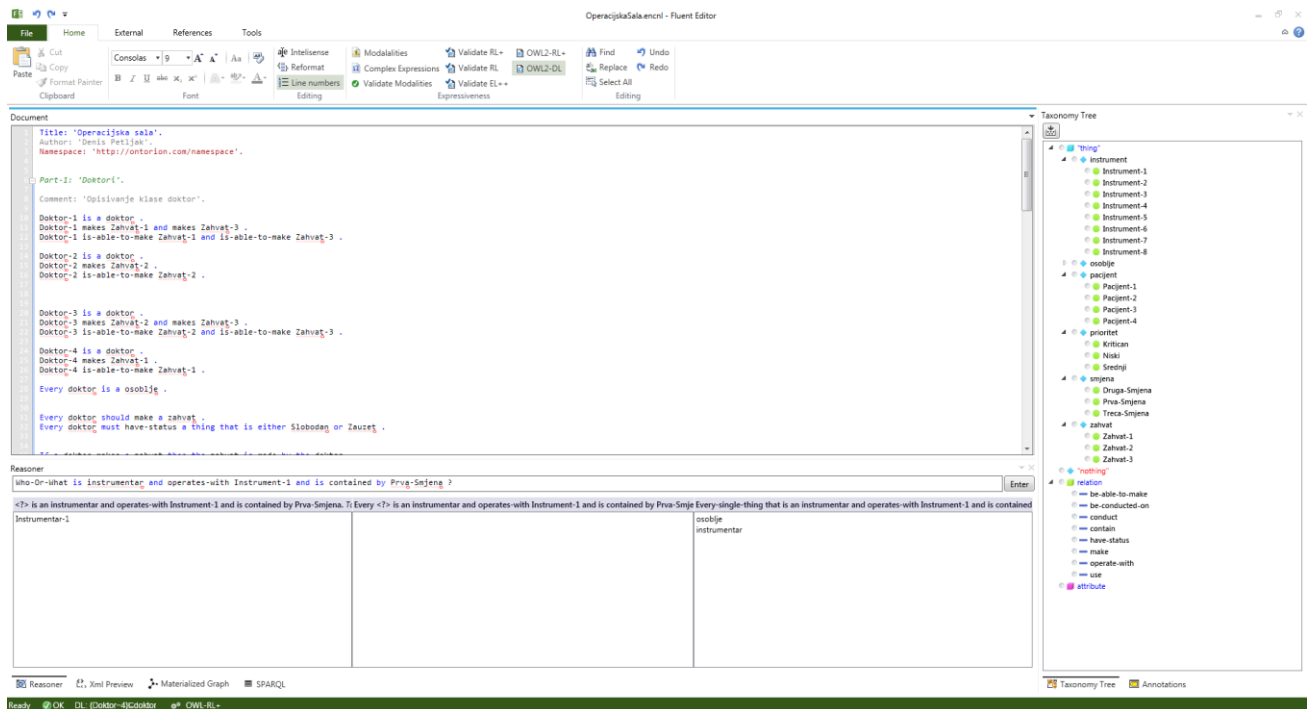


Slika 9. Grafički prikaz ontologije

Slika 10. *Onto Graph*

5.1 Software za izradu ontologija

Za izradu ontologija koristio sam program *Fluent Editor 2014*, koji je specijaliziran software za izradu i manipuliranje kompleksnim ontologijama koristeći prirodni jezik za opis ontologija. Glavna mu je mogućnost korištenje kontroliranog engleskog jezika kao jezika za modeliranje baze znanja. Tako da se ontologija može kreirati vrlo jednostavno te prirodnim jezikom. Prikaz korisničkog sučelja nalazi se na slici 11. Uz *Fluent Editor 2014*, koristio sam i *Protege v5.0.0*, koji je specijaliziran besplatni *open-source* software za izradu ontologija. *Protege* pruža grafičko korisničko sučelje, tzv. *GUI* za definiciju ontologija. Ovaj software razvijen je na Sveučilištu u Stanfordu u suradnji sa Sveučilištem u Manchesteru. Budući da ima nešto „ljepše“ grafičko sučelje koristiti ću ga za prikaz relacija i opisa dijelova ontologije.

Slika 11. Izgled korisničkog sučelja *Fluent Editor*-a

Zaključivač (*Reasoner*)

Zaključivač ili eng. *Reasoner* je iznimno bitan dio software-a za ontologije. Uloga zaključivača je provjera konzistentnosti ontologije što znači da provjerava i prikazuje sve eventualno nastale nelogičnosti npr. ako smo definirali klasu A i klasu B te ih definirali kao nezavisne (Disjoint) klase i ako negdje u ontologiji želimo pridružiti jednu klasu drugoj, zaključivač će izbaciti grešku. Naravno takva greška bila bi teško uočena kada se ne bi stvarala preko software-a sa *GUI*-om i zaključivačem nego ručnim upisivanjem koda čiji je maleni dio prikazan na slici 12..

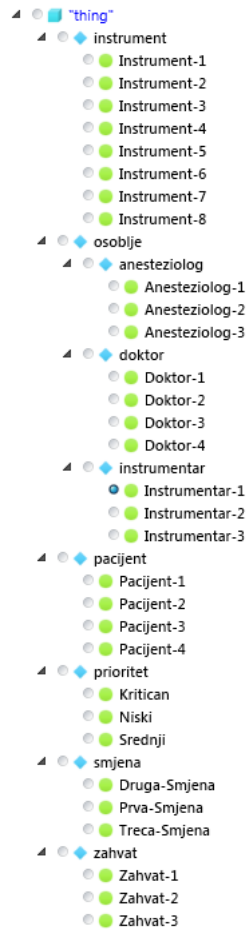
```

<!-- http://ontorion.com/namespace#Instrumentar3 -->
▼<owl:NamedIndividual rdf:about="http://ontorion.com/namespace#Instrumentar3">
  <rdf:type rdf:resource="http://ontorion.com/namespace#Instrumentar"/>
  <operatesWith rdf:resource="http://ontorion.com/namespace#Instrument1"/>
  <operatesWith rdf:resource="http://ontorion.com/namespace#Instrument4"/>
  <operatesWith rdf:resource="http://ontorion.com/namespace#Instrument6"/>
  <operatesWith rdf:resource="http://ontorion.com/namespace#Instrument8"/>
  <contains rdf:resource="http://ontorion.com/namespace#TrecaSmjena"/>
</owl:NamedIndividual>
<!-- http://ontorion.com/namespace#Kritican -->
▼<owl:NamedIndividual rdf:about="http://ontorion.com/namespace#Kritican">
  <rdf:type rdf:resource="http://ontorion.com/namespace#Prioritet"/>
</owl:NamedIndividual>
<!-- http://ontorion.com/namespace#Niski -->
▼<owl:NamedIndividual rdf:about="http://ontorion.com/namespace#Niski">
  <rdf:type rdf:resource="http://ontorion.com/namespace#Prioritet"/>
</owl:NamedIndividual>
<!-- http://ontorion.com/namespace#Pacijent1 -->
▼<owl:NamedIndividual rdf:about="http://ontorion.com/namespace#Pacijent1">
  <rdf:type rdf:resource="http://ontorion.com/namespace#Pacijent"/>
  <hasStatus rdf:resource="http://ontorion.com/namespace#Kritican"/>
</owl:NamedIndividual>
<!-- http://ontorion.com/namespace#Pacijent2 -->
▼<owl:NamedIndividual rdf:about="http://ontorion.com/namespace#Pacijent2">
  <rdf:type rdf:resource="http://ontorion.com/namespace#Pacijent"/>
  <hasStatus rdf:resource="http://ontorion.com/namespace#Srednji"/>
</owl:NamedIndividual>
<!-- http://ontorion.com/namespace#Pacijent3 -->
▼<owl:NamedIndividual rdf:about="http://ontorion.com/namespace#Pacijent3">
  <rdf:type rdf:resource="http://ontorion.com/namespace#Pacijent"/>
  <hasStatus rdf:resource="http://ontorion.com/namespace#Niski"/>
</owl:NamedIndividual>
<!-- http://ontorion.com/namespace#Pacijent4 -->
▼<owl:NamedIndividual rdf:about="http://ontorion.com/namespace#Pacijent4">
  <rdf:type rdf:resource="http://ontorion.com/namespace#Pacijent"/>
  <hasStatus rdf:resource="http://ontorion.com/namespace#Niski"/>
</owl:NamedIndividual>
<!-- http://ontorion.com/namespace#PrvaSmjena -->
▼<owl:NamedIndividual rdf:about="http://ontorion.com/namespace#PrvaSmjena">
  <rdf:type rdf:resource="http://ontorion.com/namespace#Smjena"/>
  <contains rdf:resource="http://ontorion.com/namespace#Doktor1"/>
</owl:NamedIndividual>
<!-- http://ontorion.com/namespace#Slobodan -->
<owl:NamedIndividual rdf:about="http://ontorion.com/namespace#Slobodan"/>
<!-- http://ontorion.com/namespace#Srednji -->
▼<owl:NamedIndividual rdf:about="http://ontorion.com/namespace#Srednji">
  <rdf:type rdf:resource="http://ontorion.com/namespace#Prioritet"/>
</owl:NamedIndividual>
<!-- http://ontorion.com/namespace#TrecaSmjena -->
▼<owl:NamedIndividual rdf:about="http://ontorion.com/namespace#TrecaSmjena">
  <rdf:type rdf:resource="http://ontorion.com/namespace#Smjena"/>
  ▼<rdf:type>
    ▼<owl:Class>
      ▼<owl:intersectionOf rdf:parseType="Collection">

```

Slika 12. Generirani kod ontologije

Prvo što je potrebno napraviti je definirati klase. Kako je ranije navedeno klase su skupine koje se sastoje od pojedinaca. Klasa koja sadrži pojedince naziva se superklasa eng. *Superclass* a pojedinci su podklasa eng. *Subclass*. Definirane klase prikazane su na slici 13. Sve klase su podklase *owl:Thing*-a koja je superklasa cijele ontologije. Nadalje potrebno je i opisati klase te relacije među pojedinim klasama. Neki opisi prikazani su slikom 14.



Slika 13. Klase korištene u ontologiji

```

Zahvat-1 uses Instrument-1 and uses Instrument-2 and uses Instrument-5 and uses Instrument-8 .
Zahvat-2 uses Instrument-1 and uses Instrument-4 and uses Instrument-6 and uses Instrument-8 .
Zahvat-3 uses Instrument-2 and uses Instrument-3 and uses Instrument-4 and uses Instrument-7 .
  
```

```

Anesteziolog-1 is anesteziolog .
  
```

```

Anesteziolog-1 is contained by Prva-Smjena .
  
```

```

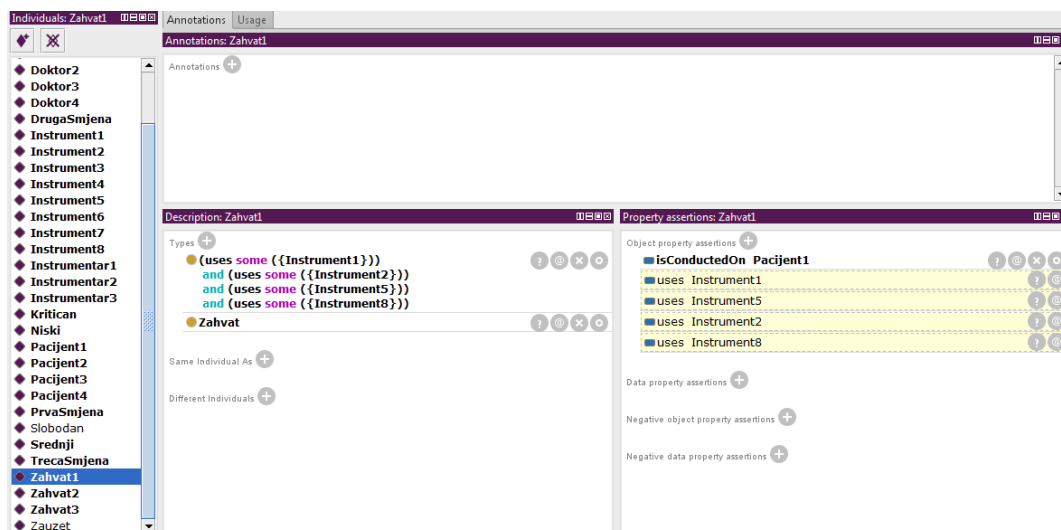
Anesteziolog-2 is anesteziolog .
  
```

```

Anesteziolog-2 is contained by Druga-Smjena .
  
```

Slika 14. Definicija nekih klasa i relacija među njima.

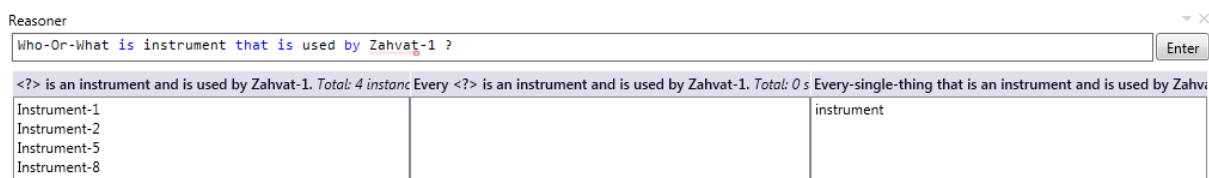
Kako bi se te relacije urednije prikazale, potrebno izvesti ontologiju u program *Protege* te slika 15. prikazuje opis sa **Zahvat1**.



Slika 15. Prikaz definicije za Zahvat1 u Protege software-u

Nakon završenog opisa ontologije možemo upisati neki upit u zaključivač (eng. *Reasoner*). Te ćemo dobiti traženi odgovor.

Recimo npr. da želimo znati koji nam instrumenti trebaju za određeni zahvat u zaključivač upisujemo: „Who-Or-What is instrument that is used by Zahvat-1 ?“ Te nam zaključivač daje odgovor kao na slici 16.



Slika 16. Primjer odgovora zaključivača za potrebne instrumente.

Isto tako možemo na primjer poslati upit koji doktor u trećoj smjeni može izvesti Zahvat-3. U zaključivač upisujemo:

„Who-Or-What is doktor that is contained by Treca-Smjena and is-able-to-make Zahvat-3 ?“ te dobivamo odgovor Doktor-3. Kao što je prikazano na slici 17.

Reasoner

Who-Or-What is doktor that is contained by Treca-Smjena and is-able-to-make Zahvat-3 ? Enter

<?> is a doktor and is contained by Treca-S Every <?> is a doktor and is contained by Tr Every-single-thing that is a doktor and is co

Doktor-3		osoblje doktor
----------	--	-------------------

Slika 17. Primjer odgovora zaključivača za upit o doktoru.

Želimo li saznati koji instrumenti su zajednički nekim zahvatima u ovom primjeru Zahvat-1 i Zahvat-2 u zaključivač upisujemo:

„Who-Or-What is instrument that is used by Zahvat-1 and is used by Zahvat-2 ?“

Također imamo i mogućnost prikaza tko je zadužen za rukovanje tim instrumentima kao na slici 18.

Who-Or-What is instrument that is used by Zahvat-1 and is used by Zahvat-2 ? Enter

<?> is an instrument and is used by Zahvat-1 and is used by Zahvat-2.	is used by	is operated-with by
Instrument-8	Zahvat-2 Zahvat-1	Instrumentar-3 Instrumentar-1
Instrument-1	Zahvat-2 Zahvat-1	Instrumentar-3 Instrumentar-1

Slika 18. Primjer odgovora zaključivača za upit o instrumentima.

Želimo li prikazati pregled svih elemenata ontologije koji su nam važni za npr. Zahvat-2, u zaključivač upisujemo: „Who-Or-What participates-in Zahvat-2 ? “ i dobivamo opsežan odgovor kao na slici 19.

Who-Or-What participates-in Zahvat-2 ? Enter

<?> participates-in Zahvat-2.	participates-in	operates-with	is contained by	makes	is-able-to-make
Instrumentar-3	Zahvat-2	Instrument-8 Instrument-6 Instrument-4 Instrument-1	Treca-Smjena		
Doktor-3	Zahvat-3 Zahvat-2		Treca-Smjena	Zahvat-3 Zahvat-2	Zahvat-3 Zahvat-2
Doktor-2	Zahvat-2		Druga-Smjena	Zahvat-2	Zahvat-2
Anesteziolog-3	Zahvat-2		Treca-Smjena		

Slika 19. Prikaz opsežnog odgovora zaključivača za jednostavan upit.

Nekada u ontologiji imamo više mogućih odgovora kao na primjer kada pitamo koji doktor izvršava zahvat nad Pacijentom-2 u zaključivač upisujemo „ Who-Or-What is doktor that participates-in zahvat that is-conducted-on Pacijent-2 ? “. Za odgovor dobivamo 2 moguća ishoda: Doktor-2 i Doktor-3.

Reasoner ▼ ×

Who-Or-What is doktor that participates-in zahvat that is-conducted-on Pacijent-2 ? Enter

<?> is a doktor and participates-in somethi | Every <?> is a doktor and participates-in soi | Every-single-thing that is a doktor and part

Doktor-2		osoblje
Doktor-3		doktor

Slika 20. Primjer odgovora zaključivača sa 2 moguća ishoda.

Kada se dogodi da nam ontologija ponudi dva moguća rješenja potrebno je odabrati jedno, no ta odluka nije jednostavna kada je treba donijeti računalo. Konačan odgovor može ovisiti o puno utjecajnih faktora koje je također potrebno nadgledati te ih baza mora sadržavati. Ontologija se može proširiti sa povezivanjem svih utjecajnih faktora koji će dovesti do konačne odluke, te je ontologiju potrebno povezati sa Bayesovom mrežom.

5.2 Bayesove mreže

Bayesova mreža je probabilističko grafički model koji prikazuje mnoštvo varijabla i njihovih zavisnosti. Takva struktura prikazuje znanje o neizvjesnoj domeni razmatranja. Svaki čvor u Bayesovoj mreži predstavlja varijablu, dok veze između čvorova predstavljaju probabilističke zavisnosti među spomenutim čvorovima. Probabilističke zavisnosti između čvorova određuju se pomoću poznatih statističkih i računskih metoda (koristeći znanja vjerojatnosti, statistike, računalnih znanosti,...)

Kod Bayesovih mreža razlikujemo 3 vrste vjerojatnosti:

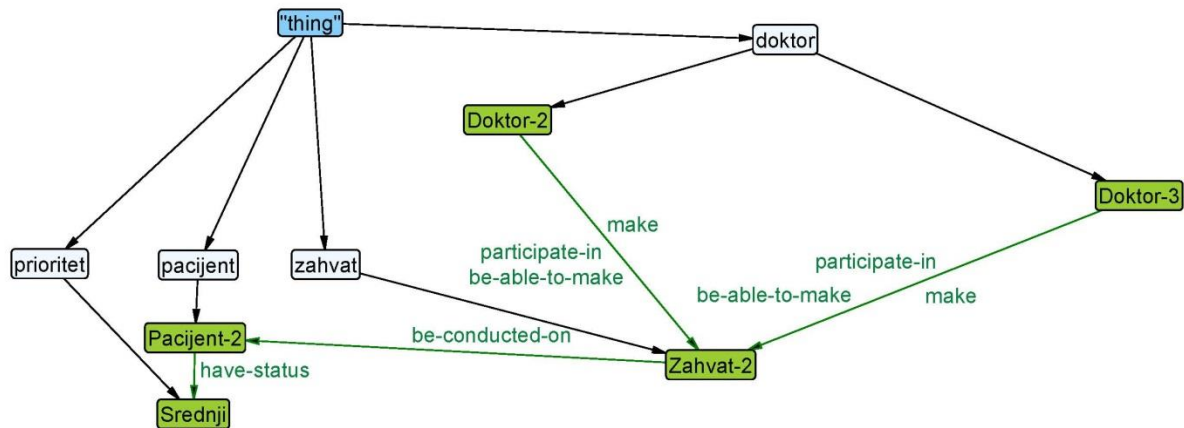
- A priori vjerojatnost
- Uvjetna vjerojatnost
- Združena vjerojatnost

Kod A priori vjerojatnosti, vjerojatnost se procjenjuje na temelju provjerenih modela koje nije potrebno provjeravati (npr. kod bacanja novčića vjerojatnost da će pasti „pismo“ ili „glava“ su jednake i iznose 50%)

Uvjetna vjerojatnost je kada nam je poznato da je nešto istina uz poznate dodatne informacije o svijetu (npr. poznato je da je vani oblačno, koja je vjerojatnost da pada kiša)

Združena vjerojatnost je kombinacija dva ili više događaja te se koristi kada je potrebno prikazati ishode svih mogućih kombinacija.

Bayesove mreže nam daju mogućnost zaključivanja no treba uzeti u obzir da zaključci nisu apsolutni nego su rezultat modela koji je proizašao iz vjerojatnosti. Za stvaranje Bayesove mreže potrebno je definirati veliki broj vrijednosti a nekada ne možemo odrediti sve kombinacije vjerojatnosti.



Slika 21. Izdvojeni dio ontologije.

Slika 19. prikazuje izdvojeni dio ontologije u kojemu su moguća dva doktora (Doktor-2 i Doktor-3) za izvođenje zahvata nad pacijentom-2. Želimo li Bayesovom mrežom doći do rješenja možemo definirati vjerojatnosti pojedinog doktora da izvršava konkretni zahvat. Uzmimo npr. 0,7 za Doktor-2 i 0,3 za Doktor-3. Tada će rezultat biti Doktor-2 kao najbolje rješenje za naš upit. Sada se postavlja pitanje kako možemo odrediti koja vjerojatnost pripada određenom doktoru. Takve vjerojatnosti mogu biti unaprijed izračunate i biti zadane ili mogu biti ishod brojnih utjecaja. U ontologiju bi bilo potrebno dodati za svakog doktora neke elemente koji određuju konačan odabir doktora, za razliku od preddefiniranih vjerojatnosti, ovakav princip imao bi mogućnost promjene tijekom vremena te bi omogućio pouzdaniji odabir najboljeg rezultata. Tako na primjer u ontologiju možemo dodati instance koje će govoriti o broju uspješnih zahvata pojedinog doktora, broj sati rada u trenutnoj smjeni, te razne druge faktore koji utječu na odabir.

6 ZAKLJUČAK

Ontologije su prije bile korištene samo u laboratorijima umjetne inteligencije no danas su česte na Internetu. Koriste se za različite primjene od tražilica do stranica za prodaju stvari. Semantički Web u današnje vrijeme vrlo je važan jer omogućava brzo pronalaženje traženih informacija u sve brže rastućem mnoštvu informacija. Pomoću njih dajemo računalima razumijevanje značenja informacija te mogućnost zaključivanja te čak i donošenja odluka. Smatram da će se ovakav pristup sve više koristiti na Internetu te će traženje informacija biti lakše, brže i preciznije. Veliki potencijal ovakvog pristupa kod robotike te autonomnog rada robota koji će uz napredovanje ove tehnologije moći samostalno izvršavati složene zadatke. Isto tako smatram da bi se ontološke baze trebale koristiti u zdravstvu kod dijagnoza bolesti, upotpunjavanjem baze sa velikom količinom podataka dijagnoza i liječenje moglo bi biti brže i kvalitetnije te pomoći liječnicima u velikoj mjeri.

Implementacija Bayesove mreže na ovaj rad ugradila bi autonomno zaključivanje no time bi se i povećala kompleksnost cijele ontologije.

Ovaj koncept može biti primijenjen na robotski sustav za implementaciju unutar robotske sale te služiti kao temelj za izradu ontološke mreže.

LITERATURA

- [1] http://www.apacsoftware.com/App_Themes/Default/images/InternetOnline570.jpg
Pristupljeno 05.07.2015
- [2] <https://journey125.files.wordpress.com/2013/11/web-1-c3a0-4.png>
Pristupljeno 05.07.2015
- [3] https://en.wikipedia.org/wiki/OSI_model#Layer_3:_Network_Layer
Pristupljeno 08.09.2015
- [4] <https://en.wikipedia.org/wiki/Internet> Pristupljeno 02.08.2015
- [5] http://teaching-wiki.sti2.at/uploads/5/54/02_SW-Architecture.pdf
Pristupljeno 03.04.2015
- [6] <https://en.wikipedia.org/wiki/HTML> Pristupljeno 11.08.2015
- [7] https://en.wikipedia.org/wiki/Semantic_Web Pristupljeno 03.04.2015
- [8] <http://www.w3.org/> Pristupljeno 28.03.2015
- [9] https://en.wikipedia.org/wiki/Internet_protocol_suite Pristupljeno 04.09.2015
- [10] <http://www.ronna.fsb.hr/> Pristupljeno 03.04.2015
- [11] http://smw.referata.com/wiki/File:ProtegeOWLTutorialP3_v1_0.pdf
Pristupljeno 03.04.2015

PRILOZI

I. CD-R disc