

UNIVERZITET SINGIDUNUM
BEOGRAD
DEPARTMAN ZA POSLEDIPLOMSKE STUDIJE I
MEĐUNARODNU SARADNJU

DOKTORSKA DISERTACIJA

EDUKACIJA I POVERENJE NA SEMANTIČKOM WEBU

MENTOR

Prof. dr Ranko Popović

KANDIDAT

D 9126/08 Ivan Stanković

BEOGRAD, 2013. godina

SADRŽAJ

1.	UVOD	1
1.1	PREDMET ISTRAŽIVANJA.....	2
1.2	HIPOTEZA.....	3
1.3	CILJEVI ISTRAŽIVANJA.....	3
1.4	ANALIZA POJMOVA	3
1.5	METODE ISTRAŽIVANJA.....	4
1.6	PRIKUPLJANJE I ANALIZA PODATAKA	5
1.7	REZULTATI I DOPRINOS	5
1.8	STRUKTURA RADA	6
2.	PREGLED POSTOJEĆIH REŠENJA	7
2.1	MREŽE POVERENJA NA SEMANTIČKOM WEBU	7
2.1.1	IMPLEMENTACIJA.....	8
2.1.2	APLIKACIJE.....	12
2.2	UPRAVLJANJE POVERENJEM NA SEMANTIČKOM WEBU	14
2.2.1	INTERPRETACIJA ALGEBRE PUTANJE.....	15
2.2.2	VEROVATNOĆA INTERPRETACIJE	19
3.	VIZIJA SEMANTIČKOG WEBA	22
3.1	DANAŠNJI WEB.....	22
3.2	UPRAVLJANJE ZNANJEM.....	23
3.3	B2C ELEKTRONSKO POSLOVANJE	24
3.4	B2B ELEKTRONSKO POSLOVANJE	24
3.5	WIKIS.....	25
3.6	PERSONALNI AGENTI: SCENARIO BUDUĆNOSTI.....	25
3.7	AGENTI	26
3.8	POREĐENJE SEMANTIČKOG WEBA I VEŠTAČKE INTELIGENCIJE	27
4.	SEMANTIČKI PODACI	28
4.1	TRADICIONALNE METODE MODELOVANJA PODATAKA	28
4.1.1	TABELARNI PODACI.....	28
4.1.2	RELACIONI PODACI.....	29
4.1.3	RAZVOJ ŠEMA.....	31
4.1.4	VEOMA KOMPLIKOVANE ŠEME	33

4.1.5	FLEKSIBILNE ŠEME	35
4.2	SEMANTIČKE VEZE	36
4.3	METAPODATAK JE PODATAK.....	37
4.4	FLEKSIBILNI MODEL PODATAKA.....	37
4.5	VEČNA BETA.....	38
5.	IZRAŽAVANJE ZNAČENJA	39
5.1	PRIMER: PODACI O FILMOVIMA	41
6.	TEORIJA GRAFOVA	42
6.1	OSNOVNI POJMOVI.....	42
7.	STANDARDI	46
7.1	PRISTUP KOJI KORISTI SLOJEVE	46
7.2	RDF	48
7.2.1	ŠTA JE RDF?.....	48
7.2.2	RDF MODEL PODATAKA	49
7.2.3	FORMATI RDF SERIJALIZACIJE.....	52
7.3	SPARQL	62
7.3.1	SELECT FORMA UPITA	64
7.3.2	OPTIONAL I FILTER OGRANIČENJA.....	64
7.3.3	VIŠESTRUKI UZORCI GRAFA.....	66
7.3.4	CONSTRUCT FORMA UPITA	68
7.3.5	ASK I DESCRIBE FORME UPITA	68
7.3.6	KORISNI MODIFIKATORI UPITA.....	69
8.	ONTOLOGIJA.....	71
8.1	MODELI SU PODACI.....	72
8.2	MODELOVANJE PODATAKA.....	72
8.2.1	KLASE I SVOJSTVA.....	72
8.2.2	MODELOVANJE FILMOVA	74
8.3	OWL	77
8.4	PROTÉGÉ	78
8.4.1	KREIRANJE NOVE ONTOLOGIJE.....	78
8.4.2	IZMENA ONTOLOGIJE	79
8.4.3	FUNKCIONALNA I INVERZNO FUNKCIONALNA SVOJSTVA.....	83
8.4.4	INVERZNA SVOJSTVA.....	83
8.4.5	DISJOINT KLASE.....	84

8.5	OWL-S.....	86
8.5.1	SERVISI U SEMANTIČKOM WEBU	86
8.5.2	OWL-S ONTOLOGIJA.....	86
8.5.3	OWL-S i WSDL	87
8.6	ONTOLOGIJE SIGURNOSTI.....	87
8.6.1	ONTOLOGIJA UVERENJA.....	87
8.6.2	ONTOLOGIJA SIGURNOSNIH MEHANIZAMA.....	88
8.6.3	ONTOLOGIJA “PROŠIRENJE SIGURNOSNIH SERVISA”	89
8.6.4	ONTOLOGIJA “PROŠIRENJA SIGURNOSNIH AGENATA”	89
8.6.5	ONTOLOGIJA PRIVATNOSTI	89
8.6.6	ONTOLOGIJA KRIPTOGRAFSKOG OZNAČAVANJA INFORMACIJE OBJEKTA	90
8.6.7	KONTEKST-SADRŽAJ POVERENJE: ONTOLOŠKI PRISTUP	90
9.	POVERENJE	91
9.1	LJUDSKO POVERENJE	91
9.2	ZAŠTO LJUDI VERUJU	92
9.3	KRITERIJUMI ZA MERENJE POVERENJA	92
9.4	MODELOVANJE POVERENJA	93
9.5	POVERENJE, KREDIBILITET, POŠTENJE, POUZDANOST	94
9.6	SVOJSTVA POVERENJA	94
9.7	KAKO LJUDI OCENJUJU KREDIBILITET WEB SAJTA	95
9.8	POVERENJE U DISTRIBUIRANIM SISTEMIMA.....	96
9.9	POVERENJE U SISTEMU PREPORUKA	97
9.9.1	PRIMENJIVOST NA SEMANTIČKI WEB.....	98
9.10	ŠIRENJE POVERENJA: MODEL POVERENJA I METRIKA.....	99
9.10.1	PKI/X.509.....	99
9.10.2	PGP WEB POVERENJA	100
9.10.3	DISTRIBUIRANI MODEL POVERENJA.....	100
9.10.4	ADVOGATO	101
9.11	POVERENJE NA SEMANTIČKOM WEBU	101
9.11.1	ONTOLOGIJA POVERENJA.....	101
9.11.2	POVERLJIV SEMANTIČKI WEB.....	102
9.11.3	ŠIRENJE POVERENJA KROZ DRUŠTVENE MREŽE.....	102
10.	ODLUČIVANJE.....	104
10.1	PREGLED RAZVOJA NAUKE O ODLUČIVANJU	104

10.2	INTERDISCIPLINARNOST ODLUČIVANJA.....	105
10.2.1	EKONOMIJA I STATISTIKA	106
10.2.2	MATEMATIKA.....	106
10.3	PROCES DONOŠENJA ODLUKA	107
10.3.1	TRADICIONALNI, ANALITIČKI MODEL DONOŠENJA ODLUKA ..	108
10.3.2	TEORIJA „IMIDŽA“: INTUITIVNI PRISTUP DONOŠENJU ODLUKE.....	109
10.3.3	IZVESNE I NEIZVESNE ODLUKE.....	109
10.4	OGRANIČENJA ZA DONOŠENJE KVALITETNIH ODLUKA	110
10.5	VİSEKRITERIJUMSKO ODLUČIVANJE.....	111
11.	EDUKACIONI MODEL NA UNVERZITETIMA.....	112
11.1	OCENJIVANJE RADA NASTAVNIKA I STUDENATA.....	113
11.1.1	OSNOVNE PRETPOSTAVKE (HIPOTETIČKI SISTEM BEZ SAVETNIKA)	113
11.1.2	MODIFIKACIJA OCENE ZA NAŠE USLOVE	114
11.2	ODREĐIVANJE TEŽINSKIH KOEFICIJENATA KORISTEĆI AHP (ANALYTIC HIERARCHY PROCESS) METOD.....	115
12.	REALIZOVANO REŠENJE	118
13.	ONTOLOGIJA VISOKOŠKOLSKE INSTITUCIJE.....	123
14.	ZAKLJUČAK	133
15.	LITERATURA	134
16.	SPISAK SLIKA I TABELA	140

1. UVOD

Sveobuhvatni razvoj Interneta i socijalnih mreža i njihov nekontrolisan uticaj na svakodnevni život nametnuli su nove načine razmišljanja, rada i zaštite. Nove tehnologije, nova hardverska i softverska rešenja, veoma brzo se smenjuju u informacionoj globalizaciji. Jedan od načina rešenja ovog problema je uvođenje standardizacije kroz definisanje određenih pravila i relacija. Ovaj rad je pokušaj da se nađu rešenja za primenu semantičkog weba i poverenja u edukacionim sistemima gde se upravlja znanjem i koristi višekriterijumsko odlučivanje. Pretraživanje po ključnim rečima zamenjuje se odgovorima na upite gde se traženo „znanje” izdvaja i prikazuje na način koji je jednostavan za korišćenje.

Suština višekriterijumskog odlučivanja je da se uoče svi mogući kriterijumi za poređenje mogućih alternativa, da se sve alternative uporede po izborima relevantnih kriterijuma i tako izabere najpovoljnija alternativa. U realnim situacijama skoro nikad nema alternativa koje su najbolje po svim kriterijumima. Zbog toga su razvijene i razvijaju se posebne metode za višekriterijumsku optimizaciju. Važnu ulogu igraju relevantne težine koje dodeljujemo pojedinim kriterijumima. Pri određivanju ovih težina, pored subjektivnih procena, uzimamo u obzir i objektivne pokazatelje za njihovo određivanje, a koji proizilaze iz karakteristika problema kojima se bavimo.

RDF (**R**esource **D**escription **F**ramework) i semantičke tehnologije su opšti meta-modeli. Oni su okruženja koja omogućavaju kreiranje jezika za razmenu i integraciju podataka. Njihova prednost je njihova proširivost. Umesto “tvrđog” kodiranja formata podataka iz više izvora u aplikacijama i izgradnje prilagođenog koda za integraciju, semantičke tehnologije pružaju standardizovane alate i šablone za specifikaciju ovih operacija. Korišćenjem ovih tehnika, čak i kada izvor podataka nije semantički iskoristljiv, omogućuje se prilagodljivost novih podataka i proširuje se funkcionalnost bez većeg napora ili dodatnog rada na kodu.

Semantičke tehnologije nisu statične. Stalno se proširuju njihove mogućnosti. Kako su zasnovane na meta-modelima, ova proširenja ne remete postojeće korišćenje tehnologija. Pojavom standardizovanih pravila, one će biti izražene korišćenjem postojećih jezika za modelovanje i sistemi koji ih koriste radiće sa istim modelom. Na ovaj način ulaganje u semantičke tehnologije ne zavisi od promena u budućnosti, gde će biti moguće korišćenje novih tehnologija bez ikakvih promena. Očekuje se pojava više semantičkih programskih šablona, poboljšanje alata, i efikasnija primena semantičkih podataka. Postoje zajednice provajdera podataka koje dele velike baze znanja različitih domena. Semantičke tehnologije su realnost, koje se stalno usavršavaju, i čijom se primenom razvijaju kvalitetnije aplikacije.

Poverenje (trust) je neophodan preduslov za saradnju dva softverska agenta. Poverenje je bitno tamo gde nema potpunog znanja ili kada postoji neizvesnost bilo koje vrste. Poverenje se različito ocenjuje od strane različitih ljudi, i na taj način je subjektivna mera. Ono je verovanje agenta o drugom i nije svojstvo korisnika. Poverenje se javlja u različitim oblastima, domenima znanja, kategorijama ili subjektima. Poverenje saopšteno jednoj strani o drugoj se naziva preporuka. Mnogi modeli poverenja vide ove preporuke kao zaseban domen znanja. Metrika poverenja, koja pokušava da modeluje kako ljudi veruju u realnom svetu se koristi u distribuiranim sistemima i sistemima preporuka kako bi pomogla poboljšanju preciznosti sistema. Otporna je na napad i manipulacije zlonamernih strana. Mnogi sistemi na Internetu trenutno imaju neki model poverenja. X.509 (PKI - **P**ublic **K**ey **I**nfrastructure i/ili CA – **C**ertificate **A**uthority) i PGP (**P**retty **G**ood **P**rivacy) su takvi primeri. Smatra se da je decentralizovani model

poverenja najefikasniji za semantički web, gde društvene mreže mogu da se koriste za širenje poverenja.

Kako se povećava broj radova u ovoj oblasti, naučnici koji se bave razvojem i dizajnom treba da uzmu u obzir istraživanja algoritama za izračunavanje poverenja. Veoma su važna proučavanja koja uzimaju u obzir složenost algoritama (zavisno od veličine grafa), i problema društvenih mreža. U ovom radu definišaćemo poverenje na osnovu sistema ocenjivanja koji koristi višekriterijumsko odlučivanje i edukacioni model razvijen za obrazovni visokoškolski sistem u Srbiji. Rezultati su opšti i mogu se primeniti ne samo u oblasti edukacije nego i za: marketing, video igre, socijalne mreže itd.

1.1 PREDMET ISTRAŽIVANJA

Predmet istraživanja je definisanje poverenja koristeći semantički web i višekriterijumsko odlučivanje, i njegova primena u edukaciji. U radu je opisana ontologija visokoškolske obrazovne institucije i detaljno prikazano rešenje izračunavanja poverenja za nastavni kadar u edukacionom modelu. Predloženi su i načini korišćenja razvijenog rešenja u različitim softverskim modelima.

Postoji veliki broj radova u kojima se koristi višekriterijumsko odlučivanje pri rešavanju različitih problema kao što su: analiza i merenje pouzdanosti sadržaja različitih dokumenata, izračunavanje pouzdanosti servisa, prikupljanje podataka kod bežičnih senzorskih mreža zasnovano na bezbednosti i poverenju, konstrukcija socijalnih mreža, hibridni sistem preporuka za G2B personalizovane e-servise, korišćenje racionalnih i objektivnih procena pri izračunavanju poverenja u edukaciji. Nijedan od ovih radova ne definiše opšti softverski model koji se može lako primeniti za različite slučajeve u edukaciji a koji su bazirani na semantičkom webu.

U radu se koristi višekriterijumsko odlučivanje pri definisanju težinskih koeficijenata koji se koriste za izračunavanje poverenja (ocena) u nastavnike, studente i menadžment jedne visokoškolske ustanove. Ove ocene se koriste pri procesu povere predmeta za sledeću školsku godinu, izboru nastavnika u nova nastavnička zvanja (vanredni, redovni profesori), ocenjivanju studenata, razvoju karijere studenata, određivanju stipendija, i usavršavanju.

Težinski koeficijenti (poverenje) će se iskoristiti i pri različitom online anketiranju, koristeći sličnost između korisnika, familijarnost između korisnika i društvenu reputaciju korisnika.

U radu se istražuju algoritmi za izračunavanje poverenja. Ovi algoritmi su konceptualno dokazivi. Veoma su važna proučavanja koja uzimaju u obzir složenost algoritama (zavisno od veličine grafa) i probleme društvenih mreža.

Uspeh semantičkog weba zavisi od implementacije različitih tehnologija. Poverenje na najvišem nivou je ključni koncept koji primenjuje različite metode višekriterijumskog odlučivanja. Web postiže svoj puni potencijal kada korisnici imaju poverenja u njegovo funkcionisanje (sigurnost) i kvalitet informacija koji pruža.

Glavni razlozi za istraživanje problema povezivanja poverenja na semantičkom webu i edukacije su sledeći nedostaci:

- Potpuno integrisane realne i virtuelne laboratorije za edukaciju u oblasti semantike, poverenja i višekriterijumskog odlučivanja;
- Interaktivnost između studenata, profesora i menadžmenta u edukacionom sistemu;

- Kompletan informacijski sistem edukacionog modela visokoškolskih institucija.

1.2 HIPOTEZA

Krenuli smo od opšte hipoteze da na semantičkom webu pouzdanost podataka treba da se ocenjuje pre obrade, tako da rezultat koji se predstavlja korisniku bude precizano definisan. Koristeći posebnu hipotezu da izazovi semantičkog weba povezani sa informacijama (mnoštvo, nejasnoća, prevara, nesigurnost i nedoslednost) definišu poverenje i njegov uticaj na odgovarajuće servise, razvijen je nov edukacioni sistem na semantičkom webu.

1.3 CILJEVI ISTRAŽIVANJA

Opšti cilj istraživanja ovog rada je modelovanje edukacionog okruženja zasnovanog na poverenju, semantičkom webu i višekriterijumskom odlučivanju. Sistemi višekriterijumskog odlučivanja i automatskog rasuđivanja su primenjeni u naučnom cilju rešavanja problema poverenja. Semantički web može da se posmatra i kao web kolektivnog sistema znanja čiji je društveni cilj da pruži korisne informacije u socijalnim mrežama.

Praktični cilj ovog istraživanja je dizajniranje i implementacija web bazirane aplikacije nezavisne od primenjene platforme za edukacioni model jedne visokoškolske ustanove.

Na osnovu dosadašnjih naučno istraživačkih iskustava iz ove oblasti, kao i istraživanja u ovoj doktorskoj disertaciji, predložene su nove metode i algoritmi rada na rešavanju različitih problema poverenja (težina) u procesu ocene rada studenata, profesora i menadžmenta visokoškolske ustanove.

1.4 ANALIZA POJMOVA

U daljem tekstu slede definicije osnovnih pojmova u domenu semantičkog weba:

- **Semantički web** omogućava da se zahtevi ljudi i mašina „shvate i zadovolje“ za korišćenje sadržaja na webu;
- **Agenti** su softverski paketi koji rade autonomno i proaktivno. Razvili su se iz koncepta objektno-orijentisanog programiranja i razvoja softvera zasnovanog na komponentama;
- **Metadata** su podaci o podacima;
- **Triple** je izjava o resursu, u formi subjekat-predikat-objekat;
- **XML** (Extensible Markup Language) je fleksibilan format teksta za kreiranje strukturiranih računarskih dokumenata;
- **RDF** (Resource Description Framework) je standardni model za razmenu podataka na Internetu;
- **RDF Schema** je izvorni jezik za pisanje ontologija;
- **URI** (Uniform Resource Identifier) je niz znakova koji se koristi za identifikaciju imena ili resursa na Internetu;
- **N-Triples** je format RDF serijalizacije za skladištenje i prenos podataka;
- **N3** je kompakcija N-Triple formata;
- **RDF/XML** je najzastupljeniji format RDF serijalizacije za skladištenje i prenos podataka;
- **RDFa** je format RDF serijalizacije koji može da se ugradi u druge formate serijalizacije kao što je XHTML;

- **FOAF** (Friend of a friend) je mašinski čitljiva ontologija koja opisuje osobe, njihove aktivnosti i njihove odnose sa drugim osobama i objektima;
- **SPARQL** (Simple Protocol and RDF Query Language) je standardizovan jezik upita za RDF grafove;
- **Ontologija** je formalno predstavljanje znanja skupom koncepata u okviru domena i odnosa ovih koncepata;
- **OWL** (Web Ontology Language) je RDF jezik razvijen od strane W3C za definisanje klasa i svojstava, kao i za omogućavanje boljeg rasuđivanja i zaključivanja preko relacija;
- **OWL-S** je ontologija za opis semantičkih web servisa;
- **XML Signature** definiše XML sintaksu za digitalni potpis;
- **SOAP** (Simple Object Access Protocol) je specifikacija protokola za razmenu strukturiranih informacija u implementaciji web servisa u računarskim mrežama;
- **Credential** u informacionim sistemima se koristi za kontrolu pristupa informaciji ili nekom drugom resursu;
- **Authentication** je process verifikacije subjekta;
- **Authorization** je provera da li subjekat ima dozvolu za obavljanje određenih radnji ili pristup određenim resursima;
- **Public Key** je jedan od dva ključa koji se koristi u asimetričnoj kriptografiji;
- **Private Key** je neobjavljeni ključ u kriptografskim sistemima koji koriste dva ključa;
- **Certificate** je digitalni dokument koji sadrži attribute vezane za vlasnika;
- **PKI** (Public Key Infrastructure) je skup hardvera, softvera, ljudi, polisa i procedura potrebnih za kreiranje, upravljanje, distribuciju, korišćenje, skladištenje i ukidanje digitalnih sertifikata;
- **BioMetric** se u računarstvu koristi kao oblik upravljanja identitetom i kontrole pristupa;
- **X.509** je ITU-T standard za PKI;
- **PGP** (Pretty Good Privacy) je računarski program za šifrovanje i dešifrovanje podataka koji omogućava šifrovanje i autentifikaciju za prenos podataka;
- **Access Control** je sistem koji omogućava kontrolu pristupa određenim oblastima ili resursima u informacionom sistemu;
- **Data Integrity** je potvrda da poruka nije promenjena prilikom prenosa;
- **Confidentiality** obezbeđuje dostupnost informacije ovlašćenom subjektu;
- **Privacy** obezbeđuje kontrolu pristupa informacija određenom subjektu;
- **Policy** je definicija sigurnosti sistema;
- **Key Distribution** u simetričnoj kriptografiji znači da obe strane moraju da imaju tajni ključ koji moraju da razmene pre korišćenja bilo kog šifrovanja;
- **XKMS** (XML Key Management Specification) je protokol razvijen od strane W3C koji opisuje distribuciju i registraciju javnih ključeva;
- **DSS** (Decision Support System) je računarski sistem za podršku pri poslovnom ili organizacionom donošenju odluka.

1.5 METODE ISTRAŽIVANJA

Istraživanje je podržano saznanjima i rezultatima iz međunarodne naučne i stručne literature, odnosno saznanjima i rezultatima drugih autora koji su u svojim člancima i knjigama

istraživali problematiku kojom se bavi i ovaj rad. U toku istraživanja, korišćeni su sledeći principi naučnog saznanja: objektivnost, pouzdanost, opštost, sistematičnost, preciznost i razvojnost.

Na osnovu postavljenog cilja ovog istraživanja odabrane su metode koje svojom kombinacijom sačinjavati metodiku rada. Korišćeno je više metoda kao što su: modelovanje, analiza i sinteza, više matematičkih i statističkih metoda, i metoda iz teorije grafova.

1.6 PRIKUPLJANJE I ANALIZA PODATAKA

Prikupljanje i analiza velikog broja podataka, testiranje i provera mogućih zaključaka dovode do koncepta novog rešenja određivanja poverenja nastavnog kadra u edukacionom modelu.

1.7 REZULTATI I DOPRINOS

Opšti cilj istraživanja ovog rada je analiza, modelovanje, dizajn i realizacija edukacionog okruženja zasnovanog na poverenju, semantičkom webu i višekriterijumskom odlučivanju. Sistemi višekriterijumskog odlučivanja i automatskog rasuđivanja su primenjeni u naučnom cilju rešavanja problema poverenja. Semantički web može da se posmatra i kao web kolektivnog sistema znanja čiji je društveni cilj da pruži korisne informacije u socijalnim mrežama.

Naučni cilj istraživanja se može opisati kao potreba za definisanjem edukacionih okruženja, kroz integraciju softvera i hardvera i primenu multidisciplinarnih rešenja kao što su poverenje, semantički web i višekriterijumsko odlučivanje.

Praktični cilj ovog istraživanja je dizajniranje i implementacija web bazirane aplikacije nezavisne od primenjene platforme za edukacioni model jedne visokoškolske ustanove.

Na osnovu dosadašnjih naučno istraživačkih iskustava iz ove oblasti, kao i istraživanja u ovoj doktorskoj disertaciji, predložene su nove metode i algoritmi rada na rešavanju različitih problema poverenja (težina) u procesu ocene rada studenata, profesora i menadžmenta visokoškolske ustanove.

Stručni doprinosi:

- Pregled iskustava u oblasti edukacije i poverenja na semantičkom webu;
- Pregled i analiza postojećih softverskih alata za semantički web;
- Kreiranje modela edukacionog okruženja na semantičkom webu;
- Kreiranje softverskog paketa za različita anketiranja u edukacionom sistemu.

Naučni doprinosi:

- Pregled i analiza dosadašnjih istraživanja iz oblasti edukacije i semantičkog weba;
- Pregled i analiza postojećih softverskih implementacija u edukacionom sistemu na semantičkom webu;
- Analiza nedostataka postojećih modela i sistema;
- Analiza raspoloživih tehnologija za realizaciju softverskog rešenja;

- Nova matematička rešenja ugrađena u algoritme određivanja poverenja;
- Modelovanje i dizajn edukacionog sistema;
- Realizacija kroz razvoj softverskog paketa edukacionog sistema baziranog na semantičkom webu;
- Analiza predloženog rešenja;
- Predlog daljih pravaca rada na osnovu prethodnih istraživanja.

Rezultata opisani u ovoj doktorskoj disertaciji su objavljeni u više časopisa međunarodnog značaja i saopšteni na više naučnih skupova u zemlji i inostranstvu.

1.8 STRUKTURA RADA

Doktorska disertacija se sastoji od 16 glavnih celina, uključujući uvodna i zaključna razmatranja, kao i literaturu. U prvom delu rada dato je poređenje postojećeg weba i vizije semantičkog weba, kao i pregled standarda i alata neophodnih za realizaciju semantičkog weba. Dati su osnovni pojmovi iz teorije grafova kao i primena u oblasti računarstva i informatike. U drugom delu rada je prikazan koncept poverenja. Poverenje je neophodan preduslov za saradnju dva agenta. U trećem delu rada su prikazani osnovni pojmovi iz teorije odlučivanja i višekriterijumskog odlučivanja. Opisana je ontologija visokoškolske obrazovne institucije i detaljno je prikazano rešenje izračunavanja poverenja za nastavni kadar u edukacionom modelu.

2. PREGLED POSTOJEĆIH REŠENJA

2.1 MREŽE POVERENJA NA SEMANTIČKOM WEBU

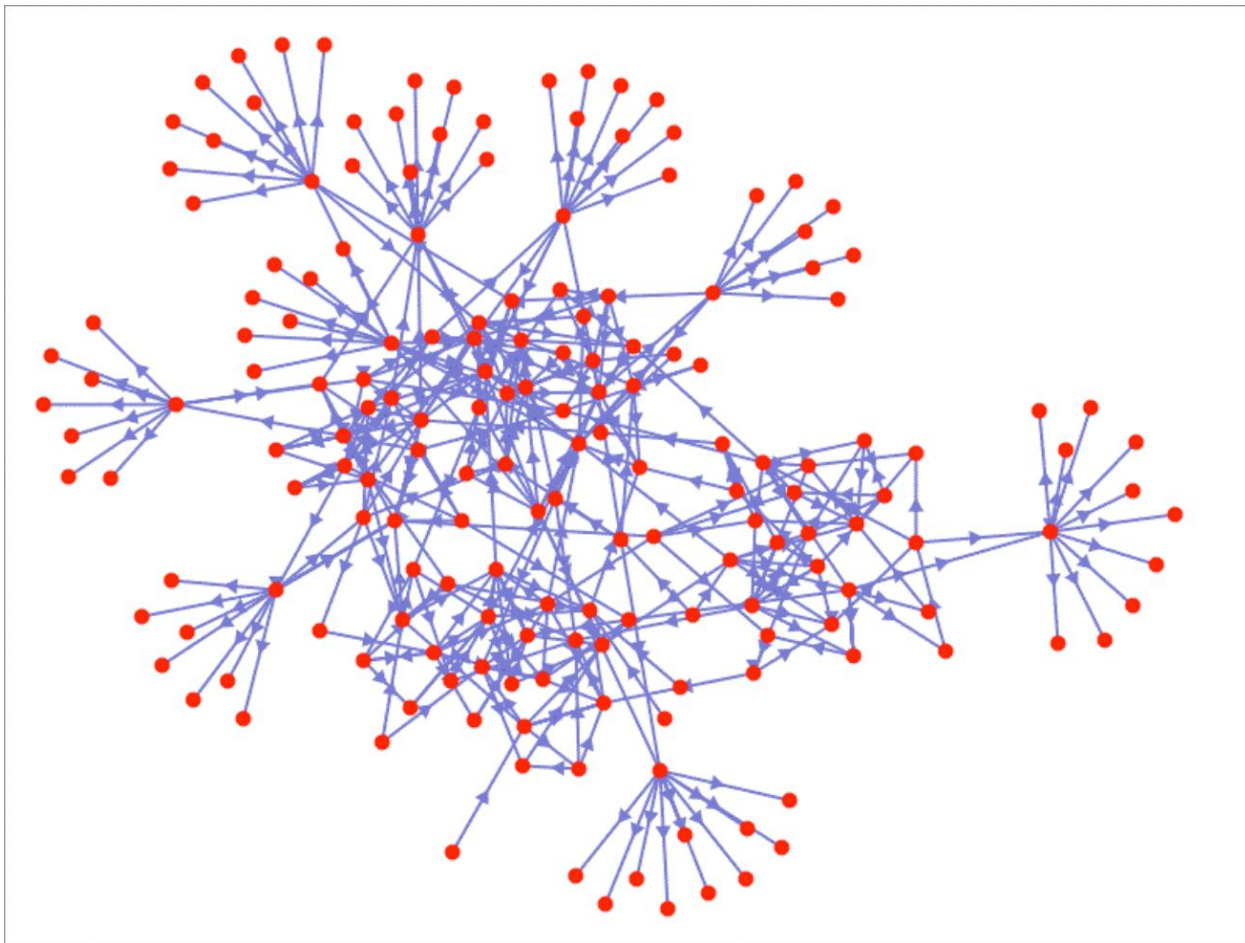
Skup stranica grupisanih po hiperlinkovima može da ukaže na uobičajene teme, ali to ne pokazuje više od generičkih odnosa između stranica. Stranice sa manje odlaznih linkova imaju manju šansu da se pojave u grupama, jer je njihova povezanost niža. Ove dve činjenice otežavaju korisniku da vidi vezu između specifičnih koncepata – klasifikacija nije dovoljno određena, i oslanja se na hiperlinkove koji ne moraju da budu prisutni [1].

Sa semantičkim webom se to menja. Semantički podaci su “mašinski-razumljivi”, i ne postoji potreba za korišćenjem heuristike za određivanje stranica. Semantički označene stranice su automatski povezane na distribuiranom webu.

Semantički web je jedan veliki graf (kolekcija čvorova i usmerenih grana koje ih povezuju). Resursi (i literali) su povezani predikatima. Kroz ovaj rad, resursi će biti označeni kao objekti, a predikati kao svojstva. Mapiranjem objekata u čvorove i svojstava u grane grafa, moguće je korišćenje algoritama i metoda za analiziranje grafa. Iako je ceo graf semantičkog weba zanimljiv, generisanjem podgrafa preciziranjem svojstava, grana i podskupa interesa moguće je prikazati veze distribuiranih podataka. Semantičkim označavanjem moguće je jednostavno preuzimanje instanci određenih klasa i skupa svojstava potrebnih za određeni projekat. Spajanje podataka iz više različitih izvora je postalo trivijalno. Generisanje društvenih mreža je takođe trivijalno ali daje korisne rezultate. Informacije o individui na mreži se održavaju u distribuiranim izvorima. Individue mogu da upravljaju svojim podacima i podacima prijatelja. Sigurnosne mere, kao što su digitalni potpisi fajlova sprečavaju širenje lažnih informacija na mreži. One grade poverenje o autentičnosti ili podacima, ali ne opisuju poverenje između korisnika na mreži.

Na društvenim mrežama postoji mnogo mera “poverenja”. Često je poverenje bazirano na poznanstvu. Označavanjem “People” kao čvora, i “knows” kao grane, dobija se usmereni graf (Slika 1). Ako A ne poznaje B , ali neki prijatelj A poznaje B , A je “blizu” poznavanja B u nekom smislu. Mnoge postojeće mreže koriste ovu meru.

Model “malog sveta” opisuje algoritme za razumevanje veza između čvorova. Isti algoritmi koji modeluju širenje zaraza [2], [3] u fizičkim društvenim mrežama mogu da se koriste za praćenje širenja virusa preko e-maila.



Sl. 1. Mreža poverenja

Postoji nekoliko faktora za razumevanje poverenja. Grane u mreži poverenja su usmerene. A može da ima poverenje u B , ali B ne mora da ima poverenje u A . Grane su ponderisane (dodeljene su im težine) nekom merom poverenja između dva korisnika. Izgradnjom takve mreže, moguće je izvesti zaključak koliko A treba da veruje nepoznatom korisniku na osnovu toga koliko njegovi prijatelji i “prijatelji-prijatelja” imaju poverenja u tog korisnika. Korišćenjem grana koje postoje u grafu, moguće je izračunati “nepostojeću” težinu grane.

2.1.1 IMPLEMENTACIJA

Semantički web poverenja zahteva od korisnika da opiše svoja uverenja o drugima. Informacije se automatski prevode i obrađuju kada korisnik kreira fajl koji opisuje koga poznaje i koliko im veruje.

2.1.1.1 Osnovna ontologija

“Prijatelj moga prijatelja” (Friend-Of-A-Friend - FOAF) [4] je projekat koji omogućava korisnicima da kreiraju i međusobno povezuju izjave o tome koga poznaju, i na taj način izgrade “Web poznanika”. FOAF šema [35] je RDF rečnik kojim korisnik može da opiše informacije o sebi i korisnicima koje poznaje. Korisnici mogu digitalno da potpišu ove fajlove kako bi informacije bile obeležene kao poznati ili eksplicitno anonimni izvor. U FOAF-u su korisnici identifikovani e-mail adresom, jer je e-mail adresa jedinstvena za svakog korisnika.

U radu [5] je opisana šema koja je dizajnirana da proširi foaf:Person, koja dozvoljava

korisniku da odredi nivo poverenja za korisnike koje poznaje. FOAF se koristi kao osoba, tako da su korisnici identifikovani na osnovu e-mail adrese. Šema poverenja dodaje svojstva u domenu foaf:Person. Svako od ovih novih svojstava precizira jedan nivo poverenja na skali od 1-9 (od apsolutnog nepoverenja do apsolutnog poverenja):

1. Distrusts absolutely
2. Distrusts highly
3. Distrusts moderately
4. Distrusts slightly
5. Trusts neutrally
6. Trusts slightly
7. Trusts moderately
8. Trusts highly
9. Trusts absolutely

Poverenje se može dati u celini, ili ograničeno na određenu temu. Korisnici mogu da odrede više nivoa poverenja za osobu, za različite teme. Korisnici mogu da odrede nivo poverenja za teme i na taj način poboljšaju mrežu. Na primer, *A* može da ima veliko poverenje u *B* kao lekara, ali uopšte nema poverenja u *B* kao programera. U ontologiji poverenja, različita ocenjivanja poverenja (“distrustAbsolutely”, “trustModerately” itd.) su svojstva “Person” klase, sa opsegom druge klase “Person”. Ova svojstva se koriste za opšte poverenje, i kodiraju se na sledeći način:

```
<Person rdf:ID="Joe">
  <mbox rdf:resource="mailto:bob@example.com"/>
  <trustsHighly rdf:resource="#Sue"/>
</Person>
```

Drugi skup svojstava je definisan za poverenje u određenim temama. Te teme odgovaraju vrednostima šeme poverenja, ali se označavaju kao poverenje u vezi sa određenom temom (“distrustAbsolutelyRe”, “trustsModeratelyRe” itd.). Opseg ovih svojstava je “TrustsRegarding” klasa, koja je definisana da grupiše “Person” i temu. “TrustsRegarding” klasa ima dva svojstva:

- “trustPerson” označava osobu od poverenja;
- “trustOnSubject” označava temu poverenja.

Ne postoji opseg ograničenja za drugo svojstvo, što omogućava korisniku da precizira bilo koju temu iz bilo koje ontologije. U sledećem primeru data je sintaksa poverenja u jednu osobu u odnosu na dve teme:

```
<Person rdf:ID="Bob">
  <mbox rdf:resource="mailto:joe@example.com"/>
  <trustsHighlyRe>
    <TrustsRegarding>
      <trustsPerson rdf:resource="#Dan"/>
      <trustsOnSubject rdf:resource="http://example.com/ont#MedicalDoctor"/>
    </TrustsRegarding>
  </trustsHighlyRe>
</Person>
```

```

</trustsHighlyRe>
<distrustsAbsolutelyRe>
  <TrustsRegarding>
    <trustsPerson rdf:resource="#Dan"/>
    <trustsOnSubject rdf:resource="http://example.com/ont#Programmer"/>
  </TrustsRegarding>
</distrustsAbsolutelyRe>
</Person>

```

Sa tematski specifičnom informacijom, upiti mogu da traže nivo poverenja za datu temu i na taj način smanje mrežu samo na relevantne grane.

Kada korisnici povezuju svoj fajl sa drugim FOAF fajlovima, formira se velika mreža. U primeru je data referenca na fajl poverenja za “Dan”. Pretraživanjem ovih linkova omogućava se formiranje velikih grafova. Mnogi projekti su nastali iz FOAF mreža:

- Co-depiction project [6];
- SVG picture annotation [7];
- FoafNaut [8], IRC bot koji pruža FOAF podatke o svima na njegovoj mreži [4].

Preciziranjem nivoa poverenja na prethodno opisan način, mreža koja nastaje postaje korisnija i interesantnija. Generiše se usmereni graf sa težinama, gde grana od A do B označava da A ima određeno poverenje u B . Težina grane označava nivo poverenja.

2.1.1.2 Izračunavanje poverenja

Usmerene grane između čvorova u grafu sadrže eksplicitno izraženu vrednost poverenja. Bez znanja da li dati korisnik ima eksplicitno poverenje u drugog, graf može da se iskoristi za zaključivanje koliko poverenje korisnik ima prema drugim korisnicima sa kojima nije direktno povezan.

Nekoliko osnovnih izračunavanja je napravljeno na mreži:

- Maksimalni i minimalni kapacitet putanja: U algoritmima korišćenim u ovom istraživanju, poverenje sledi standardna pravila kapaciteta mreže – na bilo kojoj putanji, maksimalni iznos poverenja koji izvor može da da je ograničen da ne bude veći od težine najmanje grane na toj putanji. Funkcije maksimalnog i minimalnog kapaciteta putanje identifikuju kapacitet poverenja putanje sa najvećim i najmanjim kapacitetom. Ovo je korisno u identifikaciji raspona poverenja datog od „komšije“ X do Y ;
- Maksimalna i minimalna dužina staze: Pored poznavanja različitih težina putanja od X do Y , korisno je znati koliko koraka je potrebno da bi se stiglo od X do Y . Ove funkcije dužine putanja vraćaju punu težinu lanca koja čini maksimalnu i minimalnu dužinu putanja;
- Prosečna težina: U algoritmu opisanom u nastavku, ova vrednost predstavlja preporučeni nivo poverenja za X do Y .

Prosečna težina je dizajnirana da izračuna preporuku koliko X treba da ima poverenje u Y . Na osnovu formule, prosek je podrazumevana metrika poverenja koja se koristi u svim aplikacijama koje vrše upite na datoj mreži. Metrika je jednostavna, i razumno je da različiti

korisnici žele da koriste različite metrike. U opisanoj metrici, maksimalni kapacitet svake putanje se koristi kao vrednost poverenja te putanje. Ne postoji smanjenje za duge putanje. Korisnici će možda želeći nižu vrednost poverenja za drugog korisnika koji je udaljen sa mnogo linkova za razliku od direktnih „komšija“. Drugo pitanje se bavi „poverenjem“ i „nepoverenjem“. Maksimalni kapacitet obračuna ne dodeljuje eksternu vrednost ocene poverenja; oni se smatraju realnim vrednostima. Obzirom da postoji društveni značaj za te brojeve, poželjna je kalkulacija vrednosti poverenja jedne putanje. Na primer, ako A nema poverenja u B za određenu temu, a B nema poverenja u C za tu istu temu, moguće je da A visoko oceni C . Ako A i B imaju suprotne stavove, kao i B i C , to može da znači da su A i C blizu jedan drugog. Alternativno, A može da ima još veće nepoverenje u C , u odnosu na nepoverenje u B . Sledi da je C „specijalno“ nepoverljiv.

Varijante metrike poverenja su velike, i nije realno omogućiti različite funkcije za svaku varijantu. Umesto toga, dizajniran je interfejs gde korisnik može sam da kreira i koristi svoju metriku. Ovaj proces se opisuje dalje u radu. Po podrazumevanoj vrednosti, poverenje se izračunava po sledećoj jednostavnoj funkciji. Korišćenjem maksimalnog kapaciteta svake putanje do odredišta, primenjuje se jednostavan rekurzivni algoritam za izračunavanje prosečne vrednosti. Na putanji se zanemaruje svaki čvor koji ima usmerenu granu do odredišta, u tom slučaju se koristi vrednost težine te grane. Za svaki čvor koji nije direktno povezan za odredište, vrednost se određuje prosečnom težinom svih „komšija“ koji imaju putanju do odredišta. Izračunato poverenje t od čvora i do čvora s je dato sledećom funkcijom:

$$t_{is} = \frac{\sum_{i=0}^n \left\{ \begin{array}{l} (t_{js} \times t_{ij}) \text{ if } t_{ij} \geq t_{js} \\ (t_{ij}^2) \text{ if } t_{ij} < t_{js} \end{array} \right\}}{\sum_{i=0}^n t_{ij}} \quad (1)$$

gde i ima n „komšija“ sa putanjama s , i gde je j između čvora i i čvora s . U izračunavanju prosečne vrednosti se vodi računa da u formuli nema više poverenja u odredišni čvor od poverenja koje imamo u neki od čvorova na putanji.

Iako trenutna implementacija metrike pretražuje ceo graf, moguća je jednostavna modifikacija koja limitira broj putanja od svakog čvora. Kako je ovo društvena mreža koja sledi obrazac „malog sveta“, prosečna udaljenost kroz mrežu se povećava samo logaritamski sa brojem čvorova.

2.1.1.3 Web servis poverenja

Web servis je funkcija ili kolekcija funkcija kojima može da se pristupi preko weba. Pristup mreži poverenja je moguć kao web servis. Korisnici mogu da pruže dve e-mail adrese, i da dobiju prosečnu vrednost težine poverenja. Korist web servisa je dostupnost. Može da bude sastavljen od više web servisa ili da se koristi kao komponenta aplikacije. Takođe omogućava pristup vrednostima poverenja agentima u inteligentnim web aplikacijama.

Interfejs web servisa omogućava korisnicima da proslede svoj algoritam za izračunavanje poverenja. Dostupan je JAVA API za pretraživanje grafa poverenja. To uključuje i funkciju koja vraća spisak „komšija“, sa vrednostima poverenja za svaku granu, detektovanje putanje između dve individue, i pronalazak dužine putanje. Sa ovim informacijama, korisnici mogu da pišu male Java programe za izračunavanje poverenja na osnovu svojih algoritama.

2.1.2 APLIKACIJE

2.1.2.1 TrustBot

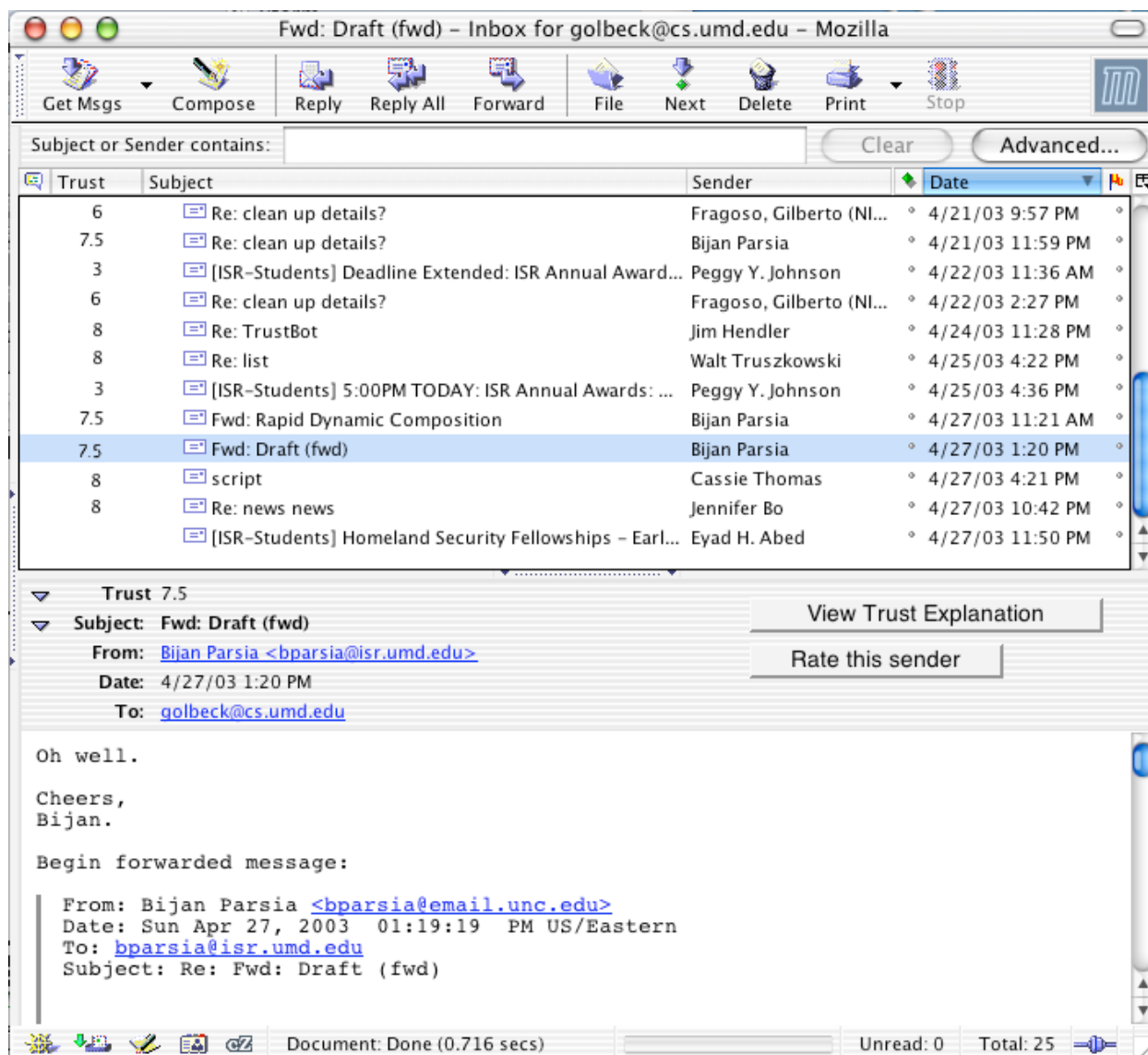
TrustBot je IRC (Internet Relay Chat) bot koji ima implementiran algoritam da definiše preporuke poverenja korisnicima na osnovu mreže poverenja koje gradi. Omogućava korisnicima transparentnu interakciju sa grafom jednostavnom serijom upita.

Pre pristupanja IRC mreži, TrustBot gradi internu prezentaciju mreže poverenja iz kolekcije distribuiranih izvora. Korisnici mogu da dodaju svoje URI-e botu u svako vreme, i da uključe podatke u graf. Bot čuva kolekciju URI-a koji su prikupljeni kada je bot startovan ili kada osveži graf. Na IRC kanalu je moguće izvršiti upite na botu kako bi se dobila prosečna vrednost težine, kao i maksimalna i minimalna dužina putanje i maksimalan i minimalan kapacitet putanje. TrustBot je trenutno aktivan na irc.freenode.net, i moguće je izvršenje upita na nadimku 'TrustBot'.

2.1.2.2 TrustMail

TrustMail je e-mail klijent, razvijen na vrhu Mozilla Messenger-a, koji pruža neposredno ocenjivanje poverenja za svaku e-mail poruku. Kako graf koristi e-mail adresu da jedinstveno identifikuje korisnika, aplikacija koristi graf poverenja da oceni e-mail. Kao kod TrustBot-a, korisnici mogu da kofigurišu TrustMail da prikaže nivo poverenja pošiljaoca na opštem nivou ili na nivou određene teme.

Za generisanje ocena, TrustMail poziva web servis i prosleđuje mu e-mail adrese pošiljaoca i mailbox-a gde je poruka isporučena. Neophodno je korišćenje mailbox e-mail adrese umesto „to“ adrese kako bi se sprečio nedostatak podataka zbog „cc“, „bcc“ i mailing list poruka. Slika 2 prikazuje primer poruke sa ocenom poverenja.



Sl. 2. TrustMail sa ocenom poverenja

Na dve stvari treba da se obrati pažnja. Prva je: ocena poverenja se izračunava na osnovu naslova e-maila (korišćenjem TrustRegarding). Ako korisnik ima ocenu poverenja na osnovu e-maila, koristi se ta vrednost. Ako ne postoji ocena poverenja na osnovu e-maila, ali postoji opšta ocena poverenja, onda se koristi ta vrednost.

Razmotrimo slučaj dve istraživačke grupe koje rade zajedno na projektu. Profesori koji vode grupe se poznaju, i svaki profesor poznaje studente u svojoj grupi, ali ne poznaje studente iz druge grupe. Ako, kao deo projekta, student pošalje e-mail profesoru druge grupe, kako će profesor da zna da je poruka od nekoga "vrednog" pažnje? Obzirom da je ime nepoznato, poruka se ne razlikuje od drugih "ne-tako-važnih" poruka u inboxu. Ovaj scenario je vrsta situacije koju TrustMail unapređuje. Profesori bi trebalo da ocene svoje studente i drugog profesora. Kako algoritam poverenja traži putanju u grafu (a ne usmerene grane), postojaće putanje od profesora do studenta preko drugog profesora. Na taj način student dobija visoku ocenu poverenja zbog srednjeg odnosa, iako se nikada nije sreo sa profesorom ili imao razmenjenu prepisku. Ako se ispostavi da student šalje junk poruke, a mreža proizvodi visoku ocenu, profesor može

jednostavno da direktno oceni pošiljaoca i smanji mu nivo poverenja. To neće da utiče na druge direktne ocene, ali će se uzimati u izračunavanje ocene kada se taj profesor bude nalazio na putanji.

TrustMail smanjuje troškove deljenja ocene poverenja u široko rasprostranjenim grupama i grupama ljudi sa slabom interakcijom, bar u kontekstu e-maila. To čini prikupljanjem mašinski čitljivih kodiranih tvrdnji o korisnicima i njihovoj poverljivosti, razumevanjem tih tvrdnji i predstavljanjem krajnjem korisniku.

Proširivanjem ove parcijalne automatizacije o proceni poverenja za druge kontekste je jednostavno. Moguće su implementacije različitih procedura zaključivanja (web servis to podržava), ili različite mreže odnosa. Na webu je teško da se to postigne, jer postoji samo jedna vrsta veza. Na semantičkom webu je moguće preklapanje različitih obrazaca odnosa poverenja nad istim skupom čvorova menjanjem predikata.

2.2 UPRAVLJANJE POVERENJEM NA SEMANTIČKOM WEBU

Sadržaj na semantičkom webu je (eksplicitno ili implicitno) u formi logičkih tvrdnji. Ako su sve ove tvrdnje skladne i sigurne, logička izračunavanja se mogu ostvariti njihovom kombinacijom. Ako nisu, koristi se izračunavanje verovatnoće (konstrukcija modela bazirana na znanju [9]). Fokus ovog rada nije na izvođenju uverenja za nove izjave date u početnom skupu izjava, već je predloženo rešenje problema uspostavljanje uverenja u izjavama koje su eksplicitno utvđene od strane jednog ili više izvora na semantičkom webu. Ova uverenja onda mogu da se koriste za izračunavanje uverenja u izvedenim izjavama. Osnovni model je da korisnikovo uverenje u izjavi treba da bude funkcija njegovog poverenja u izvore koje daju te izjave. Davanjem uverenja svakog izvora u izjavi i korisnikovo poverenje u svaki izvor, korisnikovo uverenje u izjavi može biti izračunato na različite načine, u zavisnosti od modela kako ljudi formiraju svoja uverenja. U radu [14] koristi se širok spektar kombinacija funkcija, kao što je linear pool [10], [11], noisy [12], i logistic regression [13]. Pregleda se koeficijent u ovim funkcijama (jedan po izvoru) kao mera korisnikovog poverenja u svakoj funkciji i odgovara se na pitanje: kako korisnik odlučuje koliko poverenje treba da ima u izvor koji ne poznaje direktno. Odgovor se dobija na osnovu rekurzivnog širenja poverenja: ako A ima poverenje u u B i B ima poverenje v u C , onda A treba da ima neko poverenje t u C koje je funkcija u i v . Postavljene su restrikcije na dozvoljene metode za kombinaciju poverenja koje omogućavaju efikasno i lokalno izračunavanje izvedenog poverenja. Slične restrikcije na kombinaciju uverenja omogućavaju samo lokalno izračunavanje [14].

U sistemu N korisnika koji su kao celina napravili M izjava, predstavljen je sistem koji ih tretira kao jednu.

- **Uverenja.** Svaki korisnik može da potvrdi lično uverenje u izjavi, koja je opsega $[0,1]$. Visoka vrednost znači da je izjava tačna, kredibilna, i/ili relevantna. Neka b_i predstavlja korisnika i i njegovo lično uverenje u izjavi. Ako korisnik i nije pružio ni jedno uverenje, vrednost b_i je 0. Na kolekciju ličnih uverenja u izjavi se odnosi kao na vektor kolone b ;
- **Poverenja.** Korisnik i može da odredi lično poverenje, t_{ij} , za bilo kog korisnika j . Poverenje je takođe vrednost opsega $[0,1]$, gde visoka vrednost znači da je korisnik kredibilan, poverljiv, i/ili deli slična interesovanja. Ako nije odredio, vrednost t_{ij} je 0. Kolekcija ličnog poverenja može biti predstavljena kao $N \times N$ matrica T . t_i predstavlja

vektor reda korisnikovog i ličnog poverenja u druge korisnike;

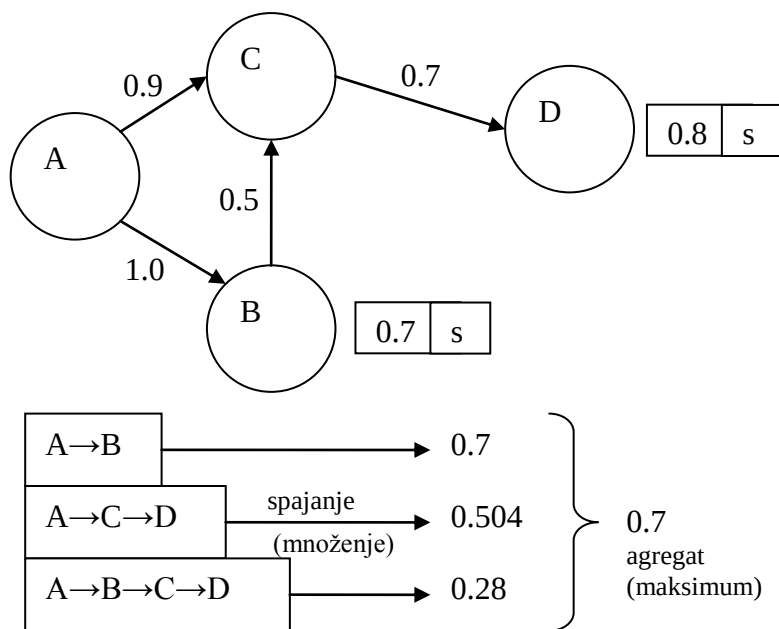
- **Spajanje.** Web poverenja pruža strukturu na kojoj je moguće izračunavanje uverenja u izjavu bilo kog korisnika. Na ovo se odnosi spojeno uverenje (σ), kako bi se razlikovalo od “korisnik-određenog” ličnog uverenja (b). Poverenje između dva korisnika je dato matricom spajanja poverenja (τ), za razliku od “korisnik-određene” matrice ličnog poverenja (T).

2.2.1 INTERPRETACIJA ALGEBRE PUTANJE

Da bi izračunali spojeno uverenje, potrebno je pojednostaviti pretpostavku da spojeno uverenje zavisi samo od putanje poverenja između korisnika i bilo kog drugog korisnika sa ličnim uverenjem u izjavi.

Definisana su spojena uverenja na osnovu interpretacije algebre putanje sa sledećim konceptualnim izračunavanjima [15]:

- Nabrojanje svih (moguć je eksponencijalni broj) putanja između korisnika i svakog drugog korisnika sa ličnim uverenjem u izjavi;
- Izračunavanje verovanja povezanih sa svakom putanjom primenom funkcije spajanja na poverenje duž putanje i ličnog uverenja poslednjeg čvora;
- Kombinacija ovih uverenja sa agregatnom funkcijom.



Sl. 3. Algebra putanje spajanja uverenja – primer weba poverenja

Moguća funkcije spajanja su množenje i minimalna vrednost. Moguće su agregatne funkcije kao što su sabiranje i maksimalna vrednost. Razne kombinacije dovode do mogućih verovanje-spajanja izračunavanja kao što su merenje najpouzdanije putanje ili maksimalan protok između korisnika i izjave.

Neka operatori $\circ$ i $\diamond$ predstavljaju spajanje i agregaciju funkcija. Na primer, $t_{ik} \circ t_{kj}$ je količina poverenja korisnika i u korisnika j , preko k , a količina kojom i ima poverenje u j preko bilo kog drugog čvora je $\diamond$ ($\forall k: t_{ik} \circ t_{kj}$). Ako je $\diamond$ sabiranje, a $\circ$ množenje, sledi $\diamond$ ($\forall k: t_{ik} \circ t_{kj}$) $\equiv \sum_k t_{ik} t_{kj}$. Definisana je operacija matrice $C = A \cdot B$ kao $C_{ij} = \diamond$ ($\forall k: A \circ B_{kj}$). Za prethodni primer, $A \cdot B$ je jednostavno množenje matrica.

2.2.1.1 Spajanje lokalnog uverenja

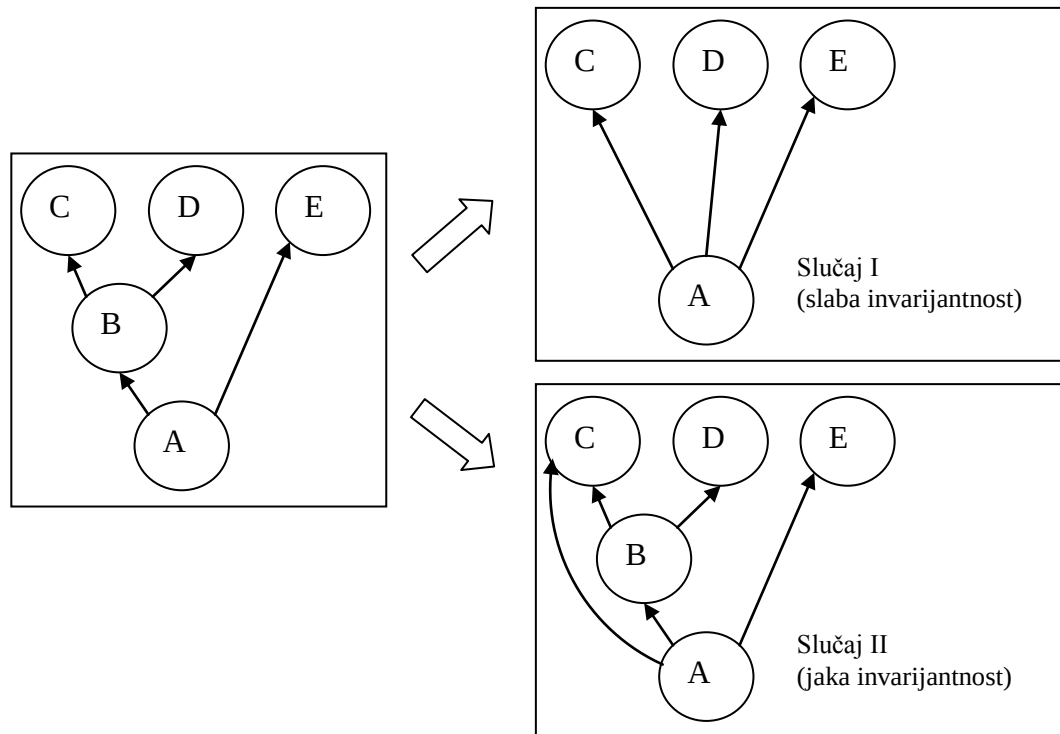
Opšte značenje uverenja koje je prethodno dato podrazumeva da korisnik potpuno poznaje mrežu uključujući i lično poverenje između svih korisnika, što je nemoguće. Moguće je spajanje uverenja lokalno, uz zadržavanje istog opšteg tumačenja. Neka problemi dobroformirane razložene putanje budu definisani kao oni za koje je operator $\diamond$ komutativan i asocijativan, i $\circ$ koji je asocijativan i distribuira se preko $\diamond$. To može da se izračuna korišćenjem opšteg algoritma tranzitivnog zatvorenja, koji koristi samo lokalne informacije. Jedan takav algoritam je:

1. $\sigma^{(0)} = b$
2. $\sigma^{(n)} = T \cdot \sigma^{(n-1)}$, ili alternativno $\sigma_i^{(n)} = \diamond$ ($\forall k: t_{ik} \circ \sigma_k^{(n-1)}$)
3. ponavljanje koraka 2 dok $\sigma^n = \sigma^{(n-1)}$

(gde $\sigma^{(i)}$ predstavlja vrednost σ u i -toj iteraciji)

U koraku 2., korisniku je potrebno samo spojeno uverenje svojih “komšija”, što mu omogućava da spoji uverenja lokalno, sa istom opštom interpretacijom. Uбудućе će se koristiti termin funkcija kombinacije uverenja koji se odnosi na dati algoritam i neki izbor od operatora $\circ$ i $\diamond$.

2.2.1.2 Jaka i slaba invarijantnost



Sl. 4. Jaka i slaba invarijantnost

- **Slučaj I:** Ako se čvor ukloni iz grafa weba poverenja, a grane se preusmere na njegove poverljive čvorove (kombinovanjem poverenja), i ako se spojeno uverenje preostalih korisnika ne promeni, u pitanju je funkcija kombinacije uverenja koja ima slabu opštu invarijantnost. Interpretacija putanje ima ovo važno svojstvo;
- **Slučaj II:** Ako se doda deo putanje poverenja usmereno od A do C , a poverenje između A i C se ne promeni, u pitanju je funkcija kombinacije uverenja koja ima jaku opštu invarijantnost. Bilo koja funkcija kombinacije uverenja sa slabom invarijantnošću za koju je agregatna funkcija idempotenta (sa značenjem, $\diamond(x, x) = \diamond(x)$), što daje jaku invarijantnost. Ovo sledi iz činjenice da je agregatna funkcija asocijativna.

2.2.1.3 Spajanje poverenja

Većina izračunavanja spajanja uverenja podrazumeva spajanje lanaca poverenja. Uverenja ulaze u obračun na kraju svake putanje. Umesto spajanja uverenja, moguće je spajanje poverenja i korišćenje tih vrednosti za izračunavanje spojenog uverenja.

Tumačenje opšteg spojenog poverenja je definisano na isti način kao i kod uverenja: Poverenje između korisnika i i korisnika j je agregatna funkcija primenjena na spajanja poverenja duž svake putanje između njih. Ako je $\diamond$ komutativno i asocijativno, a $\circ$ asocijativno i distribuirano preko $\diamond$, onda je moguće kombinovanje poverenja lokalno, održavajući opšte značenje:

1. $\tau^0 = T$

2. $\tau^{(n)} = T \cdot \tau^{(n-1)}$
3. ponavljanje koraka 2. dok $\tau^{(n)} = \tau^{(n-1)}$
 $(\tau^{(i)})$ je vrednost τ u iteraciji i)

Da bi izveo izračunavanje, korisnik treba da zna spojeno poverenje svojih “komšija”. Ovo dovodi do sledeće teoreme: za široke klase funkcija, spajanje poverenja ostvaruje isto što i spajanje uverenja.

Teorema 1

Ako je operator $\diamond$ komutativan i asocijativan, a operator $\circ$ asocijativan i distribuiran preko operatora $\diamond$, i T, τ, b i σ prethodno definisani, sledi $T \cdot \sigma = \tau \cdot b$.

2.2.1.4 Ciklusi

Do sada smo pretpostavljali da je graf acikličan. Međutim graf web poverenja nije acikličan. Epinions (aplikacija koja pomaže korisniku da izabere proizvod prilikom kupovine) web poverenja je povezan i cikličan. Funkcija kombinacije je nezavisna od ciklusa ako nema uticaja ciklusa na putanju između dva korisnika. Ako postoji nezavisnost od ciklusa, agregacija preko beskonačne putanje će konvergirati, jer jedino putanje bez ciklusa utiču na izračunavanje.

Predlog 1

Svi rezultati i teoreme predstavljeni do sada su daleko od primenjivosti na ciklične grafove ako $\diamond$ i $\circ$ definišu putanje nezavisne od ciklusa.

Na ciklične grafove, funkcija kombinacije koja nije zavisna od ciklusa ima upitno svojstvo kojim korisnik može da utiče na poverenje drugih prema njemu izmenom ličnih poverenja. Međutim, zahtevanje nezavisnosti od ciklusa funkcije kombinacije može biti previše restriktivno. U nastavku će biti predstavljena alternativna interpretacija koja omogućava korišćenje funkcija kombinacije koje su zavisne od ciklusa.

2.2.1.5 Izbor funkcije kombinacije uverenja

Izbor funkcije kombinacije uverenja može da zavisi od domena aplikacija, željenog uverenja i ciklusa semantike, i očekivanog tipičnog društvenog ponašanja u tom domenu. Idealna funkcija kombinacije može da bude zavisna od korisnika. U ovom radu se koristi množenje kod spajanja. Sledi kratak pregled tri različite funkcije agregacije.

- **Maksimalna vrednost:** Korišćenje maksimuma za kombinaciju uverenja je konzistentno sa fuzzy logikom, u kojoj se može pokazati da je najrazumnija funkcija uverenja za izvođenje generalizovane “or” operacije preko $[0, 1]$ [16]. Maksimum ima prednost jer je nezavisan od ciklusa, jako dosledan, i prirodno radi sa vrednostima koje nedostaju (dozvoljava im da budu 0). Sa maksimumom, korisnik će verovati bilo čemu što veruje barem jedan od korisnika u koga on ima poverenja – ovo je razumno, preterano optimistično ponašanje;
- **Minimalna vrednost:** Minimum je zavistan od ciklusa. U fuzzy logici, minimalna vrednost se koristi za izvođenje „and“ operacije. Sa minimumom, korisnik će verovati samo izjavi kojoj veruju svi korisnici u koje ima poverenja;
- **Prosek:** Prosek ne zadovoljava uslove za dobro-formiranu putanju prethodno

navedenu (prosek nije asocijativan). Međutim, prosek može da se izračuna korišćenjem dve agregatne funkcije: zbir i brojač (brojač jednostavno vraća ukupan broj putanja). Prosleđivanjem ove dve vrednosti, svaki čvor može lokalno da izračuna prosek. Prosek je zavistan od ciklusa.

2.2.1.6 Izračunavanje

Nezavisno od ciklusa, slabo dosledne funkcije kombinacije su problem dobro-formirane putanje. σ i τ mogu da se izračunaju korišćenjem standardnih algoritama tranzitivnog "zatvaranja" (transitive closure). Najjednostavniji od njih je "polu-naivan" algoritam [17], čija je složenost $O(N^4)$, i u suštini propisuje ponavljanje aplikacije ažuriranja jednačine uverenja. Ako je pokrenut kao peer-to-peer sistem, "polu-naivan" algoritam može lako da se paralelizuje, zahtevajući složenost $O(N^3)$ izračunavanja po čvoru [18]. Drugi algoritam je Warshall algoritam [19], koji izračunava tranzitivno zatvorenje čija je složenost $O(N^3)$. U praksi će većina korisnika da odredi samo nekoliko korisnika kao „komšije“, i broj iteracija potrebnih za potpuno širenje informacija je mnogo manji od N , što izračunavanje čini efikasnim. Teorema 1 omogućava da se dobije odluka da li korisnik želi spojeno poverenje ili uverenje. Efikasnost metoda zavisi da li je sistem implementiran kao peer-to-peer mreža ili kao server, od broja „komšija“ za datog korisnika, broja korisnika, broja izjava u sistemu i broja upita napravljenih od strane svakog korisnika.

2.2.2 VEROVATNOĆA INTERPRETACIJE

Ova formulacija se odnosi na verovatnoću tumačenja kombinacije opšteg uverenja. Tretman [20] je motivisan slučajnim kretanjem na Markovljevom lancu, koji prikazuje praktičnu primenu otkrivanja visoko-kvalitetnih web stranica, pretpostavio da se prati normalizovan skup ličnih poverenja datog korisnika. Nejednačine koje su definisali Markov, Chebyshev i Chernoff su klasični i široko korišćeni rezultati moderne teorije verovatnoće primenjene na oblasti teorije odlučivanja, dinamičkog programiranja, statistike, finansija, teorije grafova i zaštite. U radu [89] data je opšta diskretna nejednakost Chebyshev-ljevog tipa. Veliki broj poznatih nejednakosti su specijalni slučajevi dobijene nejednakosti.

Ako slučajni „surfer-znanja“ ide od korisnika do korisnika u potrazi za uverenjima, na svakom koraku surfer po nekoj verovatnoći bira „komšiju“, kod koga ide na osnovu distribucije poverenja trenutnog korisnika. Sa verovatnoćom jednakom vrednosti uverenja trenutnog korisnika, on kaže „da, veruje u izjavu“. U suprotnom kaže „ne“. Dalje, prilikom izbora sledećeg korisnika, surfer će nasumično sa verovatnoćom $\lambda_i \in [0,1]$, ignorisati poverenje i „skočiće“ direktno nazad na originalnog korisnika i . Definisan je metod kombinacija koji ima opštu verovatnoću tumačenja ako su zadovoljeni sledeći uslovi:

- τ_{ij} je verovatnoća kojom se korisnik i na svakom datom koraku nalazi kod korisnika j ;
- σ_i je verovatnoća kojom korisnik i na svakom datom koraku kaže „da“.

Konvergencija svojstava ovih slučajnih šetnji se detaljno proučava; σ i τ će konvergovati dok god je mreža nesmanjiva i aperiodična [21]. λ_i je samo-poverenje, i precizira težinu koju korisnik dodeljuje sopstvenim uverenjima i poverenjima. Ponašanje surfera-znanja je slično inteligentnom surferu predstavljenom u [22], koji je generalizacija PageRank-a koja dozvoljava neuniformisane tranzicije između web stranica. Ono što personalizuje izračunavanje korisnika i je slučajni restart, koji „prizemljuje“ surfera na poverenja i -tog korisnika. Rezultantna poverenja

mogu biti drastično različita od korišćenja PageRank-a, sve dok je broj „komšija“ tipično manji.

2.2.2.1 Izračunavanje

Poverenje i -tog korisnika u korisnika j je verovatnoća da je njegov slučajni surfer na korisniku k , pomnoženo sa verovatnoćom surferovog „prelaska“ na korisnika j , sumirano za svako k . Uzimajući λ_i u proračun takođe, sledi $\tau_{ij} = \lambda_i \delta(i - j) + (1 - \lambda_i) \sum_k \tau_{ik} t_{kj}$, gde je $\delta(0) = 1$ i $\delta(x \neq 0) = 0$ i svaki red t je normalizovan. U formi matrice:

$$\tau_i = \lambda_i I_i + (1 - \lambda_i) \tau_i T \quad (2)$$

gde je I_i i^{th} red koji matrice identifikacije. Kako bi se zadovoljila opšta verovatnoća tumačenja, σ_i mora da bude verovatnoća da korisnik i -tom slučajnom surferu kaže „da“. Ovo je verovatnoća da je on na datom korisniku ono vreme za koje veruje u izjavu:

$$\sigma_i = \sum_k \tau_{ik} b_k \text{ ili } \sigma_i = \tau_i b \quad (3)$$

2.2.2.2 Lokalno uverenje ili spajanje poverenja

U odeljku 2.2.1.1 autori su pokušali da izvedu ovo izračunavanje korišćenjem samo lokalnih informacija. Pokazali su da je to moguće samo u specijalnom slučaju gde je konstanta $\lambda_i = \lambda$. Sledi razvoj formule (2):

$$\tau = \lambda [\sum_{m=0}^{\infty} (1 - \lambda)^m T^m]. \quad (4)$$

Primećuje se da je $T^0 = I$. Zamenom u formuli (3),

$$\sigma = \lambda [\sum_{m=0}^{\infty} (1 - \lambda)^m T^m] b, \quad (5)$$

što je zadovoljeno rekurzivnom definicijom:

$$\sigma = \lambda b + (1 - \lambda) T \sigma. \quad (6)$$

U cilju izračunavanja spojenog uverenja, svaki korisnik treba samo da zna svoja lična uverenja i spojena uverenja svojih „komšija“. Pored intuitivnih „žalbi“, korisnik ima i verovatnoću tumačenja takođe: i -ti korisnik bira komšiju verovatnoćom definisanom na osnovu njegove distribucije poverenja T_i , i onda, sa verovatnoćom $(1 - \lambda)$, prihvata „komšijino“ (spojeno) uverenje, i sa verovatnoćom λ prihvata njegovo sopstveno uverenje. Treća formula je jednaka:

$$\tau = \lambda I + (1 - \lambda) T \tau. \quad (7)$$

Verovatnoća tumačenja za kombinaciju uverenja u osnovi uzima srednju vrednost težina „komšijskih“ uverenja. Treba obratiti pažnju da bi srednja vrednost težina imala smisla, ako korisnik nema određeno uverenje, potrebno je definisati vrednost. Mogu da se primene tehnike kao što su one korišćenje u kolaborativnom filtriranju [54] i Bayesian mrežama za rad sa nedostajućim vrednostima. Ako su samo relativne ocene uverenja neophodne, onda može biti dovoljno koristiti 0 za sva neodređena uverenja.

2.2.2.3 Sličnost verovatnoće i interpretacije putanje

Postoje mnoge sličnosti između verovatnoće i interpretacije putanje. Kod obe, uverenje može biti spojeno izvršenjem upita nad „komšijama“ zbog njihovih uverenja, množenjem (ili spajanjem) sa poverenjem za svakog komšiju, i sabiranjem (ili agregacijom) zajedno. Oba tumačenja dozvoljavaju izračunavanje spojenih uverenja tako što prvo spajaju poverenja. Ako je agregatna funkcija sabiranje, a funkcija spajanja množenje, onda je jedina razlika između ova dva

tumačenja faktor λ . Ako je $\lambda = 0$, onda je formula (6) za izračunavanje σ funkcionalno ista kao algoritam za izračunavanje σ u algebri interpretacije putanje. Ako je $\lambda = 0$, onda formula (2) za izračunavanje τ_i jednostavno pronalazi primarni sopstveni vektor matrice T . Kako postoji samo jedan primarni sopstveni vektor, ovo znači da će τ_i biti isti za sve korisnike (pod pretpostavkom da je graf aperiodičan i nesmanjiv). Kako je ovo moguće u algebri interpretacije putanje, u kojoj se očekuje različito poverenje vektora po korisniku? Odgovoravajuća funkcija kombinacije algebre putanje je zavisna od ciklusa, i kao rezultat korisnikova lična uverenja će biti „isprana“ od beskrajne agregacije uverenja drugih korisnika. Kao i u tumačenju verovatnoće, na kraju će svi korisnici imati ista spojena uverenja.

Obe metode imaju slične kompromise u vezi sa dizajnom arhitekture. Lako se mogu implementirati kao čvor-ka-čvoru (peer-to-peer) i klijent-server arhitekture. Očekuje se da će sistem biti robustan iz razloga što zlonamerni korisnici dobijaju manje poverenje vremenom. Dalje, kako je podrazumevana vrednost poverenja u korisnika 0, nije „dobro“ za korisnika da kreira više drugih imena, i korisnici su motivisani da održavaju kvalitet informacija.

Izračunavanje weba poverenja nije podložno „link-spamu“, fenomenu u PageRank-u gde korisnik može da poveća nečije poverenje u njega generisanjem više virtuelnih korisnika koji imaju poverenja u tog korisnika. U PageRank-u, uniformisani slučajni skokovi surfera znače da svaki virtuelni korisnik daje mali iznos PageRank-a, koji su „dati“ spameru, čime se povećava njegov rang. Sa webom poverenja, ovim tehnikama se ne dobija ništa osim ako je korisnik u mogućnosti da ubedi druge da imaju poverenja u njegove virtuelne korisnike, što će se desiti samo ako korisnik omogući korisne informacije.

3. VIZIJA SEMANTIČKOG WEBA

3.1 DANAŠNJI WEB

World Wide Web je promenio način na koji ljudi međusobno komuniciraju, posluju, dolaze do informacija i kako se zabavljaju. Web transformiše razvijeni svet prema ekonomiji znanja i šire u društvo znanja. Ovaj razvoj je takođe promenio način na koji se razmišlja o računarima. U početku su računari korišćeni za numeričke proračune. Danas su najdominantniji u obradi informacija, tipičnim aplikacijama sa bazama podataka, obradi teksta i "igricama". Trenutno se na računare gleda kao na polaznu tačku globalnog informacionog prostora.

Većina današnjeg web sadržaja je prilagođena ljudskoj upotrebi. Tipično korišćenje weba uključuje ljude koji traže i koriste informacije, pronalaze i ostaju u kontaktu sa drugim ljudima, pregledaju kataloge online prodavnica i naručuju proizvode popunjavajući forme. Ove aktivnosti nemaju odgovarajuću podršku softverskih alata. Pretraživači koji koriste ključne reči u pretragama kao što su Yahoo i Google su glavni alati za korišćenje današnjeg weba. Web ne bi postigao toliki uspeh da nema ovih alata. Međutim, postoje ozbiljni problemi u korišćenju ovih alata:

- Veliki odziv, mala preciznost. Čak i ako su u rezultatu pretrage relevantni dokumenti, oni vrlo lako mogu da postanu neupotrebljivi ako postoji još ogroman broj srednje relevantnih ili nerelevantnih dokumenata. U ovom slučaju višak informacija predstavlja problem;
- Mali odziv ili bez odziva. Često se dešava da se ne dobije ni jedan odgovor na zahtev ili da rezultat nije zadovoljavajući. U ovom slučaju manjak informacija predstavlja problem;
- Rezultati su veoma osetljivi na rečnik. Često ključne reči ne vraćaju željeni rezultat, u ovom slučaju relevantni dokumenti koriste različitu terminologiju od originalnog upita. Ovo je nezadovoljavajuće, jer slični semantički upiti treba da vrate slične rezultate;
- Rezultati su pojedinačne web stranice. Ako su potrebne informacije koje se nalaze u više raznih dokumenata, korisnik mora da pokrene više upita da bi prikupio relevantne dokumente i da onda ručno izvuče delove sadržaja koji su mu potrebni.

Uz svakodnevna poboljšanja u tehnologiji pretraživača, teškoće u osnovi ostaju iste. Količina web sadržaja nadmašuje tehnološki napredak. Čak i kada pretraga uspe, osoba je ta koja mora da pregleda dokument i izdvoji informacije koje su joj potrebne. Tu nema zadovoljavajuće podrške za pregledanje informacija i zahteva se puno vremena. Termin "pretraživanje informacija" koji se koristi u pretraživačima nije baš najbolji, "pretraživač lokacija" je adekvatniji pojam. Rezultati web pretraga nisu čitljivi drugim softverskim alatima jer su pretraživači često izolovane aplikacije. Glavna prepreka u pružanju bolje podrške korisnicima na webu je ta što značenje web sadržaja nije dostupno mašinama. Naravno postoje alati koji mogu da preuzmu tekst, razdvoje ga na delove, provere pravopis, izbroje reči, ali kada dođe do interpretacije rečenica i izdvajanja korisnih informacija za korisnike, mogućnosti ovih alata su veoma ograničene. Teško je razlikovati značenje:

- Ja sam profesor računarskih nauka.

od

- Ja sam profesor računarskih nauka ako vi tako mislite.

Jedno od rešenja za poboljšanje trenutne situacije u obradi teksta je da se na osnovu sadržaja razvija napredna tehnika na bazi veštačke inteligencije i računarske lingvistike. Ovaj pristup se koristi već neko vreme ali uprkos napretku zadatak i dalje izgleda previše ambiciozan.

Alternativni pristup je da se web sadržaj prikaže u formi koju mašine lakše procesiraju i da se koriste inteligentne tehnike kako bi se iskoristio ovaj prikaz. Ovaj pristup se odnosi na “revoluciju” weba kroz Semantic Web inicijativu. Semantički web neće biti novi globalni izvor informacija paralelan postojećem webu. On se postepeno razvija i evoluirao iz postojećeg weba [23].

Semantički web propagira internacionalno telo za standardizaciju weba, World Wide Web konzorcijum (W3C). Inicijator semantičkog weba je Tim Berners-Lee, osnivač WWW u kasnim 1980-tim. Očekuje se realizacija “originalnog” weba, uz viziju gde značenje informacija ima mnogo veću ulogu u odnosu na današnji web.

Semantički web se veoma brzo razvija, i sve je veći broj projekata i programa u svim delovima sveta.

3.2 UPRAVLJANJE ZNANJEM

Upravljanje znanjem se odnosi na prikupljanje, pristup i održavanje znanja u velikim organizacijama. Velika preduzeća upravljanje znanjem vide kao ključnu aktivnost zbog internih intelektualnih resursa na osnovu kojih mogu značajno da povećaju svoju produktivnost, kreiraju nove vrednosti i povećaju svoju konkurentnost. Upravljanje znanjem je posebno važno za međunarodne organizacije sa geografski distribuiranim delovima.

Većina informacija je trenutno dostupna u slabo strukturiranom obliku, na primer: tekst, audio i video. Iz perspektive upravljanja znanjem ograničenja su u sledećim oblastima [24]:

- Pretraživanje informacija - Kompanije obično zavise od limitiranog pretraživanja na osnovu ključnih reči;
- Izdvajanje informacija - Ljudski trud i napor su potrebni za pregledanje preuzetih dokumenata i za izdvajanje relevantnih informacija;
- Održavanje informacija - Trenutno postoje problemi kao što su nedoslednost u terminologiji i neuspeh da se izuzmu zastarele informacije;
- Otkrivanje informacija - Do novih znanja može da se dođe data mining-om u korporativnim bazama podataka. Međutim, ovaj zadatak je još uvek težak u slabo strukturiranim kolekcijama dokumenata;
- Pregled informacija - Često je poželjno da se zabrani pristup određenim informacijama određenoj grupi zaposlenih. “View” u bazama podataka može da sakrije određene informacije ali je realizacija teža preko intraneta ili weba.

Cilj semantičkog weba je da omogući sisteme za napredno upravljanje znanjem:

- Znanje će biti organizovano u konceptualnom prostoru u skladu sa svojim značenjem;
- Automatizovani alati će podržavati održavanje proverom nedoslednosti i izdvajanjem

novih znanja;

- Pretraživanje na osnovu “ključnih reči” zameniće “odgovori na upite”: traženo znanje će biti izdvojeno i prikazano na način jednostavan za korišćenje;
- “Odgovori na upite” će biti mogući kroz više dokumenata;
- Definisane ko može da vidi određeni deo informacije (čak i deo dokumenta) će biti moguće.

3.3 B2C ELEKTRONSKO POSLOVANJE

Tipičan scenario za B2C (**B**usiness-**T**o-**C**onsumer) uključuje korisnika koji posećuje jednu ili više online prodavnica, pregleda njihovu ponudu, odabira i naručuje proizvode.

U najboljem slučaju korisnik će da sakupi sve informacije o cenama i uslovima kupovine u svim važnijim online prodavnicama i izabere najbolju ponudu. Ovakvo ručno pregledanje je veoma zahtevno i traži puno vremena. Najčešće korisnik poseti jednu ili par online prodavnica pre donošenja odluke.

Postoje alati za kupovinu na webu u vidu “shopbotova” koji posećuju online prodavnice, izdvajaju proizvode i njihove cene i kreiraju pregled tržišta. Njihova funkcionalnost je moguća korišćenjem wrapper (omotača) programa, koji izdvajaju informacije iz online prodavnice. Potrebno je da se za svaku prodavnicu razvije po jedan omotač. Ovaj pristup ima nekoliko nedostataka [25]:

- Informacije se izdvajaju iz online prodavnica na osnovu ključnih reči i drugih značenja na osnovu analize teksta. Ovaj proces koristi pretpostavke o blizini određenih informacija. Na primer, cena je označena rečju "cena" praćena simbolom \$ iza kog sledi pozitivan broj. Ovaj heuristički pristup je podložan greškama i ne garantuje uvek tačnu informaciju;
- Drugi problem je nemogućnost izdvajanja svih informacija. Ne postoje informacije o vremenu isporuke, troškovima dostave, ograničenju na zemlje u kojima su moguće isporuke, uslovima korišćenja i sl. Sve ove dodatne informacije su važne za donošenje odluke korisnika. Pored toga programiranje omotača zahteva puno vremena, i svaka promena u online prodavnici zahteva dodatno programiranje.

Semantički web će omogućiti razvoj softverskih agenata koji mogu da interpretiraju sve informacije o proizvodu i uslove korišćenja usluge:

- Cena i informacije o proizvodu biće izdvojene i upoređene sa potrebama korisnika;
- Dodatne informacije o reputaciji online prodavnice biće preuzete iz nezavisnih izvora. Na primer, nezavisna agencija za procenu ili potrošačko telo;
- Dodatno programiranje omotača (wrapper) neće biti potrebno;
- Sofisticirani agenti za kupovinu biće u mogućnosti da vode automatske pregovore na ime kupca sa agentima prodaje.

3.4 B2B ELEKTRONSKO POSLOVANJE

Većina korisnika povezuje komercijalni deo weba sa B2C e-poslovanjem, ali najveći

ekonomski rast od svih online tehnologija leži u delu B2B (**B**usiness-**T**o-**B**usiness) e-poslovanja.

Tradicionalno, kompanije razmenjuju podatke koristeći tehnologiju Electronic Data Interchange (EDI). Ova tehnologija je komplikovana i razumljiva samo od strane eksperata. Teška je za programiranje i održavanje, i podložna je greškama. Svaka B2B komunikacija zahteva odvojeno programiranje, tako da je ovakva komunikacija skupa. EDI je izolovana tehnologija, podaci koji se razmenjuju ne mogu jednostavno da se integrišu u druge poslovne aplikacije.

Internet je idealna infrastruktura za B2B komunikaciju. Kompanije sve više ulažu u web aplikacije, i novi poslovni modeli kao što su B2B portali su sve više zastupljeni. Ipak, B2B je otežan zbog nedostatka standarda. HTML je isuviše slab da bi efikasno podržao navedene aktivnosti, ne podržava ni strukturu ni semantiku informacija. Standard XML je veliki napredak ali je upotrebljiv samo u slučaju gde je poznat rečnik koji se koristi i njegovo značenje.

Realizacija semantičkog weba će omogućiti kompanijama da ostvare partnerstva mnogo jednostavnije. Razlike u terminologiji će biti rešene korišćenjem standarda apstraktnog modela domena, a podaci će se razmenjivati korišćenjem servisa za prevođenje. Softverski agenti će moći automatski da izvrše servise kao što su licitacije, pregovori i izrade ugovora [25].

3.5 WIKIS

Korišćenje WWW-a je prošireno na alate koji omogućavaju korisnicima aktivno učešće na webu. Ovaj razvoj je bio revolucionaran i dobio je naziv Web 2.0.

Deo tog pravca uključuje wiki, kolekciju web stranica koja omogućava korisniku da doda sadržaj preko web interfejsa (uglavnom strukturiran tekst i veze ka drugim stranicama). Wiki sistemi omogućavaju stvaranje zajedničkog znanja, tako što dozvoljavaju korisnicima da unose i menjaju informacije bez vlasništva nad sadržajem, ograničenja u pristupu i strogih propisa. Wiki sistemi se koriste u različite svrhe:

- Razvoj skupa znanja od strane širokog kruga korisnika. Najpoznatiji rezultat je Wikipedia;
- Upravljanje znanjem o raznim aktivnostima ili projektima. Primeri su razmena ideja, koordinacione aktivnosti i razmena zapisa sa sastanaka.

Još uvek nisu definisani nedostaci i ograničenja ove tehnologije. Wiki sistemi definitivno mogu da imaju koristi od upotrebe semantičke tehnologije. Osnovna ideja je da se napravi struktura koja koristi nasleđivanje povezivanjem stranica, i koja je dostupna mašinama i bez navigacije. Ovo se može napraviti obogaćivanjem strukture teksta i hiperveza sa semantičkim napomenama koje se odnose na osnovni model znanja u wikiju. Na primer vezu iz Beograd na Srbija moguće je obogatiti napomenom sa informacijom da je "lociran u". Ovakve informacije mogu da se iskoriste za verifikaciju, napredno pretraživanje i kontekst sadržaja [26].

3.6 PERSONALNI AGENTI: SCENARIO BUDUĆNOSTI

Ako se zamisli sledeći scenario: korisnik hoće da ide u bioskop i da gleda neku komediju, a posle bioskopa hoće da jede u nekom dobrom italijanskom restoranu. Korisnik otvara Google, pronalazi bioskope u svojoj blizini, pregleda njihov repertoar, pročita kratak sinopsis o svakom filmu kako bi doneo odluku koji će film da gleda, posle toga pretražuje restorane sa italijanskom kuhinjom u blizini bioskopa, čita utiske ljudi koji su posetili restoran i na kraju donosi odluku.

Sve ovo zahteva previše vremena i pregledanje dosta web stranica.

Umesto više različitih pretraga, potrebno je samo napisati kompleksnu rečenicu ili dve i personalni agent će da uradi ostalo. U ovom primeru, korisnik bi napisao “Hoću da idem u bioskop i da gledam neku komediju, posle bioskopa hoću da jedem u nekom dobrom italijanskom restoranu. Koje su moje opcije?”. Personalni agent će da analizira upit, i na osnovu lokacije korisnika da pretraži Internet za najbolje moguće odgovore i da organizuje rezultat za korisnika. Takođe, personalni agent će vremenom da nauči više o korisniku i njegovim interesovanjima pa će moći da postavlja jednostavnija pitanja. Na primer, može da postavi pitanje “Gde da idem na ručak?”, agent će da konsultuje svoje zapise o korisniku i da na osnovu lokacije predloži listu restorana [27].

3.7 AGENTI

Agenti su softverski paketi koji rade autonomno i proaktivno. Razvili su se iz koncepta objektno-orijentisanog programiranja i razvoja softvera zasnovanog na komponentama.

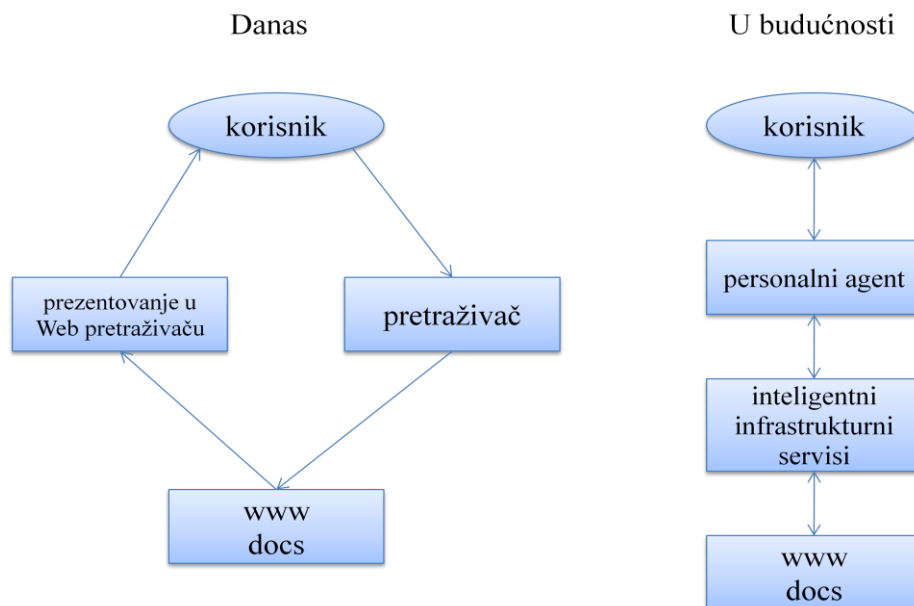
Personalni agent na semantičkom webu dobija neki zadatak i karakteristike od strane korisnika, pretražuje informacije u web izvorima, komunicira sa drugim agentima, upoređuje informacije sa zahtevima i karakteristikama korisnika, odabira određene mogućnosti i daje odgovore korisniku. Primer je agent iz “3.6 Scenario budućnosti”.

Treba napomenuti da agenti neće zameniti korisnike (ljude) na semantičkom webu, niti će neminovno sami donositi odluke. U mnogim slučajevima (možda i u većini), njihova uloga će biti da sakupi i organizuju informacije, i predstave mogućnosti korisniku, kao u primeru izbora bioskopa, filma i restorana na osnovu datih karakteristika korisnika.

Semantički web agenti će koristiti sve rapoložive tehnologije:

- Matepodaci će se koristiti za identifikaciju i izdvajanje informacija iz web izvora;
- Ontologije će se koristiti kako bi pomogle u web pretragama za interpretaciju preuzetih informacija i za komunikaciju sa drugim agentima;
- Logika će se koristiti za obradu preuzetih informacija i donošenje odluka.

Druge tehnologije kao što su jezici za komunikaciju agenata će takođe biti potrebne. Takođe, za napredne aplikacije će biti korisno da formalno zastupaju poverenja, zahteve i namere agenata, i da kreiraju i održavaju modele korisnika [28].



Sl. 5. Inteligenti personalni agenti

3.8 POREĐENJE SEMANTIČKOG WEBA I VEŠTAČKE INTELIGENCIJE

Većina tehnologija potrebnih za realizaciju semantičkog weba je izgrađena na osnovu istraživanja na području veštačke inteligencije (AI). S obzirom da AI ima dugu istoriju, koja nije uvek bila komercijalno uspešna, najgora stvar koja bi mogla da se dogodi semantičkom webu je da ponovi greške AI.

Ova zabrinutost je neopravdana. Realizacija vizije semantičkog weba se ne oslanja na ljudski nivo inteligencije. Izazovima se pristupa na potpuno drugačiji način. Problem kod AI-a je naučne prirode, možda može da se uporedi sa centralnim problemima fizike (objasniti fizički svet) ili biologije (objasniti živi svet). Postoje predviđanja da će se teškoće u ostvarivanju ljudskog nivoa veštačke inteligencije rešiti u narednih deset ili dvadeset godina.

Delimična rešenja mogu da rade na semantičkom webu. Krajnji cilj AI-a je da se izgradi inteligentni agent sa ljudskim nivoom inteligencije. Cilj semantičkog weba je da pomogne korisnicima u svakodnevnim online aktivnostima. Semantički web u velikoj meri koristiti aktuelnu AI tehnologiju. Napredak u toj tehnologiji dovodi do boljeg semantičkog weba [25].

Postojeća tehnologija nam omogućava rešenje većine problema. Trenutna najveća potreba je integracija, standardizacija, razvoj alata i prihvatanje od strane korisnika. Dalji tehnološki napredak će dovesti do naprednijih rešenja u odnosu na trenutne mogućnosti semantičkog weba. Osnova semantičkog weba je "Web podataka".

4. SEMANTIČKI PODACI

4.1 TRADICIONALNE METODE MODELOVANJA PODATAKA

4.1.1 TABELARNI PODACI

Postoji mnogo načina za modelovanje podataka. Najjednostavniji skup podataka je tabelarni prikaz podataka. Tabelarni podaci su bilo koji podaci koji su prikazani u tabeli kao što je Excel tabela ili HTML tabela. Prednost ovih podataka je u tome što su vrlo jednostavni za pregled i korišćenje [29].

Restoran	Adresa	Kuhinja	Cene	Radno vreme
Deli Llama	Peachtree Rd	Deli	\$	Mon, Tue, Wed, Thu, Fri
Peking Inn	Lake St	Chinese	\$\$\$	Thu, Fri, Sat
Thai Tanic	Branch Dr	Thai	\$\$	Tue, Wed, Thu, Fri, Sat, Sun
Lord of the Fries	Flower Ave	Fast Food	\$\$	Tue, Wed, Thu, Fri, Sat, Sun
Marquis de Salade	Main St	French	\$\$\$	Thu, Fri, Sat
Wok This Way	Second St	Chinese	\$	Mon, Tue, Wed, Thu, Fri, Sat, Sun
Luna Sea	Autumn Dr	Seafood	\$\$\$	Tue, Thu, Fri, Sat
Pita Pan	Thunder Rd	Middle Eastern	\$\$	Mon, Tue, Wed, Thu, Fri, Sat, Sun
Award Weiners	Dorfold Mews	Fast Food	\$	Mon, Tue, Wed, Thu, Fri, Sat
Lettuce Eat	Rustic Parkway	Deli	\$\$	Mon, Tue, Wed, Thu, Fri

Tabela 1. Tabela restorana

Podaci koji se čuvaju u tabeli su generalno laki za prikaz, sortiranje, štampanje i uređivanje. Podaci iz Excel tabela ne izgledaju kao "modelovani", ali svako smeštanje podataka u redove i kolone daje svakom podatku posebno značenje. Za razliku od modernih metoda modelovanja, ne postoji puno varijacija na način pregleda tabelarnih podataka. Često se kaže tabela umesto baza podataka.

Interesantno je da se pomene da postoji semantika u tabelama, redovima i kolonama u koje se smeštaju podaci, na primer kuhinja u restoranu, objašnjava onome ko pregleda podatke koja se hrana služi u restoranu. Činjenica da se "Chinese" nalazi u redu gde je restoran "Peking In", a u koloni "Kuhinja" govori da se u "Peking In-u služi kineska hrana". Ovo je jasno na osnovu toga što je poznato šta su restoran i kuhinja i kako se čita tabela. Ovo možda izgleda trivijalno, ali je važno ovo imati na umu u istraživanju različitih načina razmišljanja o podacima.

Podaci upisani na ovaj način imaju očigledna ograničenja. Svi radni dani su upisani u poslednjoj koloni "Radno vreme". Ovo je u redu za pregled tabele, ali ako je potrebno da se dodaju informacije kao što je vreme kada je restoran otvoren ili noćni specijaliteti dobijamo veliku zbrku. U teoriji je ovo moguće, na primer da se dodaju ove informacije u zagradi nakon dana.

Restoran	Adresa	Kuhinja	Cene	Radno vreme
Deli Llama	Peachtree Rd	Deli	\$	Mon(11-16), Tue(11-16), Wed(11-16), Thu(11-19), Fri(11-20)
Peking Inn	Lake St	Chinese	\$\$\$	Thu(17-22), Fri(17-23), Sat(17-23)

Tabela 2. Previše podataka u tabeli

Ovako upisani podaci ne mogu da se koriste u pretrazi restorana koji je otvoren do kasno u petak uveče. Sortiranje kolone ne pomaže zbog nemogućnosti da se shvati dublji smisao unetog teksta. Program jednostavno ne razume da je u koloni "Radno vreme" uneto više informacija različitih vrednosti.

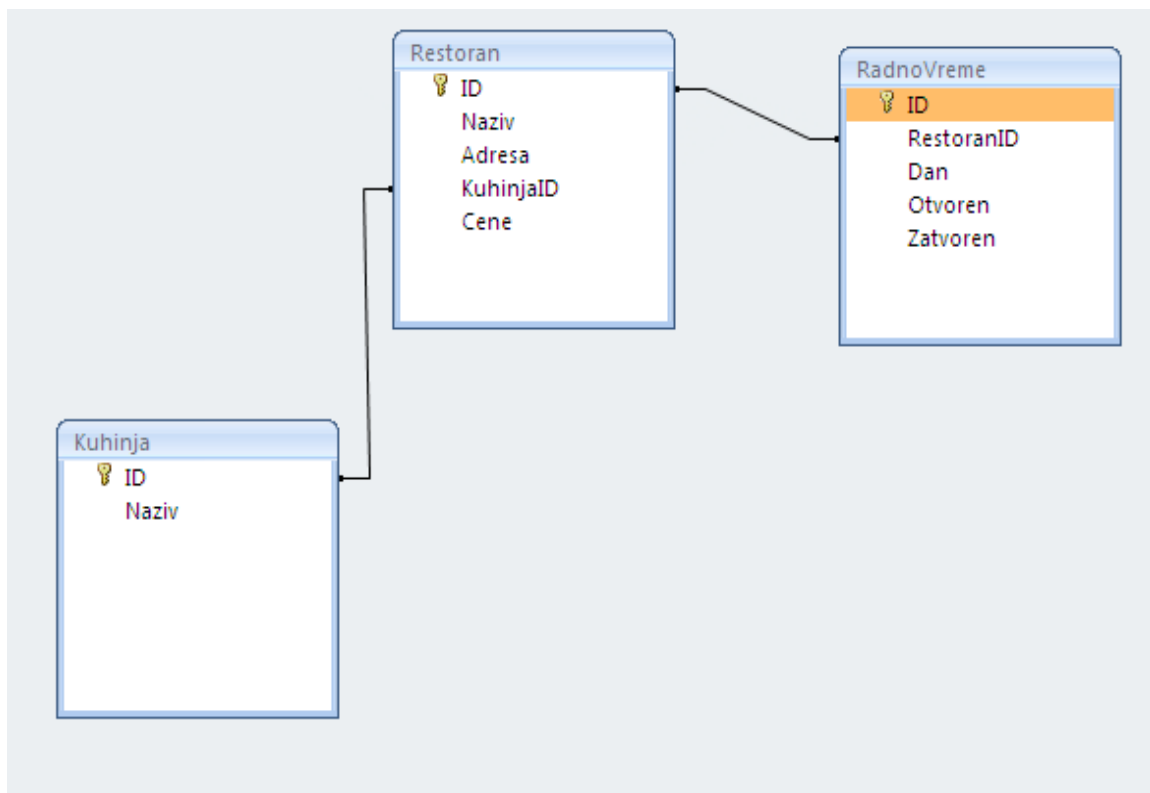
Problemi sa tabelama se komplikuju kada postoji više tabela koje imaju referencu na iste podatke. Na primer, ako postoji još jedna tabela sa kritikama restorana, ne postoji jednostavan način da se pretraže dva dokumenta da bi se pronašao restoran u blizini koji ima dobre kritike. Excel eksperti često koriste makroe za pretraživanje tabela kako bi dobili željeni rezultat, ovaj model je limitiran i često nije promenljiv od strane drugih korisnika.

Potreba za sofisticiranim modelom podataka je više nego očigledna.

4.1.2 RELACIONI PODACI

Relacione baze podataka se koriste u raznim aplikacijama. Proizvodi kao što su Oracle DB, MS SQL, MySQL i PostgreSQL su rezultat višegodišnjih istraživanja, razvoja i optimizacije. Relacione baze podataka su veoma brzi i moćni alati za skladištenje velikih skupova podataka gde je model podataka dobro poznat i gde je način upotrebe predvidiv [29].

U suštini relacione baze podataka omogućavaju da se više tabela spoji na standardizovan način. Za skladištenje podataka o restoranima moguće je da se definiše šema kao na slici 6. Ovo omogućava da se podaci o restoranima predstave na korisniji i fleksibilniji način.



Sl. 6. Jednostavna šema podataka o restoranima

ID	Naziv	Adresa	Cene	Add New Field
1	Deli Llama	Peachtree Rd	\$	
ID	Dan	Otvoren	Zatvoren	Add New Field
1	Mon	11	16	
2	Tue	11	16	
3	Wed	11	16	
4	Thu	11	19	
5	Fri	11	20	
*	(New)			
*	(New)			
2	Chinese			
ID	Naziv	Adresa	Cene	Add New Field
2	Peking Inn	Lake St	\$\$\$	
ID	Dan	Otvoren	Zatvoren	Add New Field
6	Thu	17	22	
7	Fri	17	23	
8	Sat	17	23	
*	(New)			
*	(New)			
*	(New)			

Sl. 7. Relacioni podaci o restoranu

Sada je umesto sortiranja i filtriranja u jednoj koloni moguće izvršenje sofisticiranijih upita. Upit koji pronalazi sve restorane koji su otvoreni petkom u 22h može da se napiše koristeći

SQL upit na sledeći način:

```
SELECT Restoran.Naziv, Kuhinja.Naziv, RadnoVreme.Otvoren, RadnoVreme.Zatvoren
FROM Restoran, Kuhinja, RadnoVreme
WHERE Restoran.KuhinjaID=Kuhinje.ID
AND Restoran.ID=RadnoVreme.RestoranID
AND RadnoVreme.Dan="Fri"
AND RadnoVreme.Otvoren<22
AND RadnoVreme.Zatvoren>22
```

što daje sledeći rezultat:

Restoran.Naziv	Kuhinja.Naz	Otvoren	Zatvoren
Peking Inn	Chinese	17	23

Sl. 8. Rezultat pretrage

Semantika podataka je eksplicitnija u relacionom modelu. Značenje vrednosti je opisano šemom. Pregledom tabela se vidi nekoliko tipova entiteta, tip „Restoran“, tip „Dan“. Ovi entiteti imaju specifičnu vezu. Baza podataka ne zna šta „Restoran“ znači, ali može da odgovori na zahtev da prikaže sve restorane sa datim osobinama. Svaki podatak je označen na osnovu tabele i kolone u kojoj se nalazi.

4.1.3 RAZVOJ ŠEMA

Relacione baze podataka su odlične za skupove podataka gde je model podataka unapred poznat i gde se zna kako će ti podaci da se koriste. Mnoge aplikacije kao što su katalogi proizvoda ili kontakt liste se uklapaju odlično u relacione šeme, jer su to generalno fiksni skupovi polja i imaju tipičan model korišćenja [29].

Međutim integraciju podataka preko weba karakteriše brzo menjanje tipova podataka, i programeri nikad ne znaju šta će biti na raspolaganju i kako će korisnici hteti da ih koriste. Pretpostavimo da je baza podataka restorana aktivna, i da je potrebno da se integrišu novi podaci koji sadrže dodatne informacije koje se ne nalaze u trenutnoj šemi baze restorana kao što je prikazano u tabeli 3.

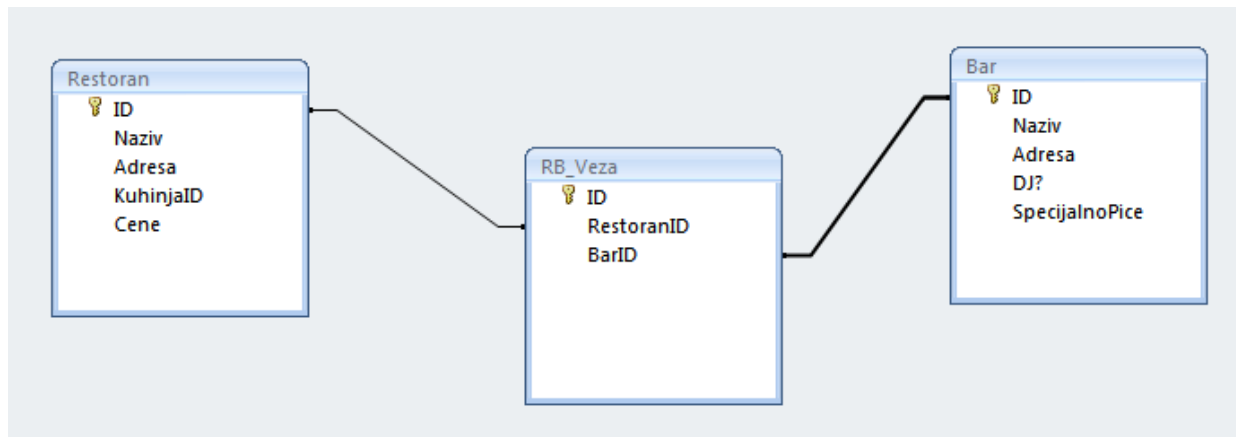
Bar	Adresa	DJ	Specijalno piće
The Bitter End	14th Ave	No	Beer
Peking Inn	Lake St	No	Scorpion Bowl
Hammer Time	Wildcat Dr	Yes	Hennessey
Marquis de Salade	Main St	Yes	Martini

Tabela 3. Novi skup podataka barova

Naravno, mnogi restorani takođe imaju barove, gde u večernjim satima prestaju sa služenjem hrane. Nova tabela barova pokazuje da je Marquis de Salade pored toga što je Francuski restoran, takođe i bar sa DJ-em. Tabela takođe pokazuje i specijalno piće koje daje još dodatnih informacija o Marquis. Ove baze su za sad odvojene, ali postoje korisnici koji će hteti da ih pretražuju i pronađu mesto gde se služi francuska hrana i kasnije ostanu na martini.

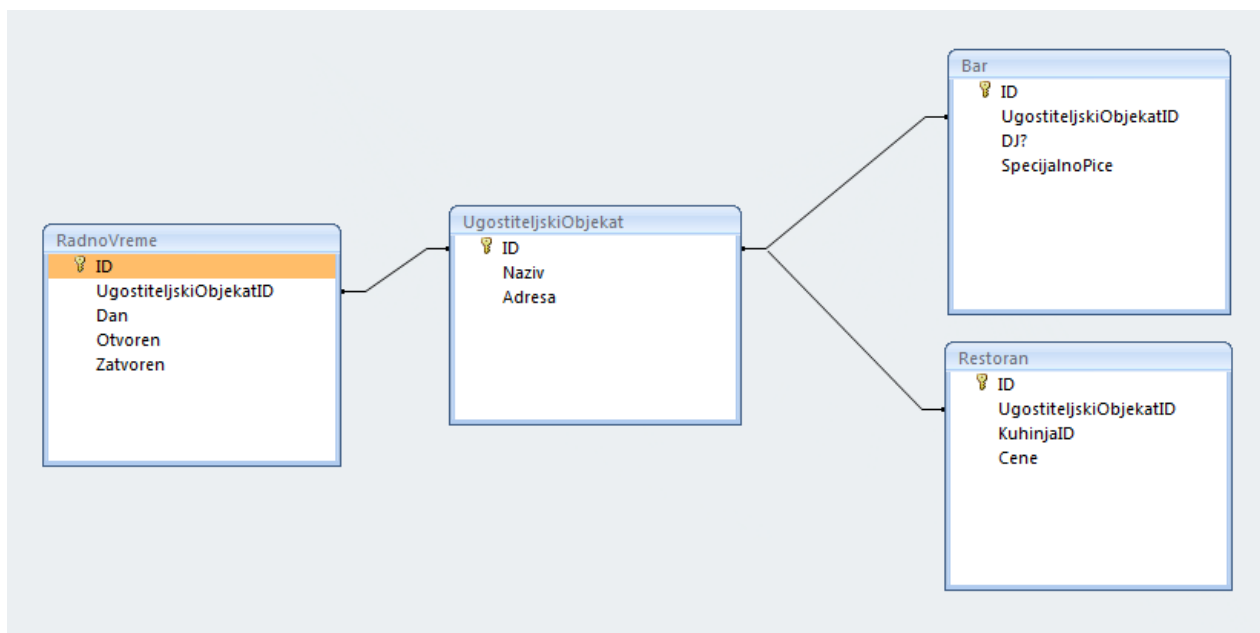
Najjednostavniji način da postojeća baza podrži nove podatke je veza ka novoj tabeli, na

ovaj način ne bi bilo potrebno da se menja postojeća struktura baze podataka.



Sl 9. Veza barova na postojeću šemu

Sada je upit moguć, ali se nailazi na novi problem, postoje dva naziva i adrese u bazi podataka za ugostiteljske objekte koji su i barovi i restorani i samo veza među njima koja nam kaže da je u pitanju isto mesto. Da bi se izvršila pretraga na osnovu adrese, upit bi morao da obuhvati obe tabele. Takođe, vrsta hrane je povezana samo sa restoranom, a ne i sa barom. Unošenje i ažuriranje podataka je mnogo komplikovanije. Precizniji način modelovanja bi bio da se kreira nova tabela UgostiteljskiObjekat koja bi odvojila bar i restoran kao svoje podtipove.



Sl. 10. Normalizovana šema koja odvaja ugostiteljski objekat od njegove svrhe

Ovo izgleda kao rešenje problema, ali su svi postojeći podaci u starom modelu podataka i moraju da se transformišu u novi model. Ovaj proces se naziva migracija. Pored migracije podataka, potrebno je migrirati i sve upite i odgovarajući programski kod. Ako postoji web sajt sa bazom restorana, potrebno je pronaći način kako da se postojeći kod, upiti i baza podataka ažuriraju bez značajne pauze u radu sajta. Nova grana softverskog inženjerstva se bavi ovim pitanjima, koristeći tehnike kao što su uskladištene procedure i objektno-relaciono mapiranje

(ORM) slojeva koji pokušavaju da razdvoje osnovnu šemu iz sloja biznis logike i uproste promenu šeme. Ove tehnike su korisne, ali i unose složenost i probleme.

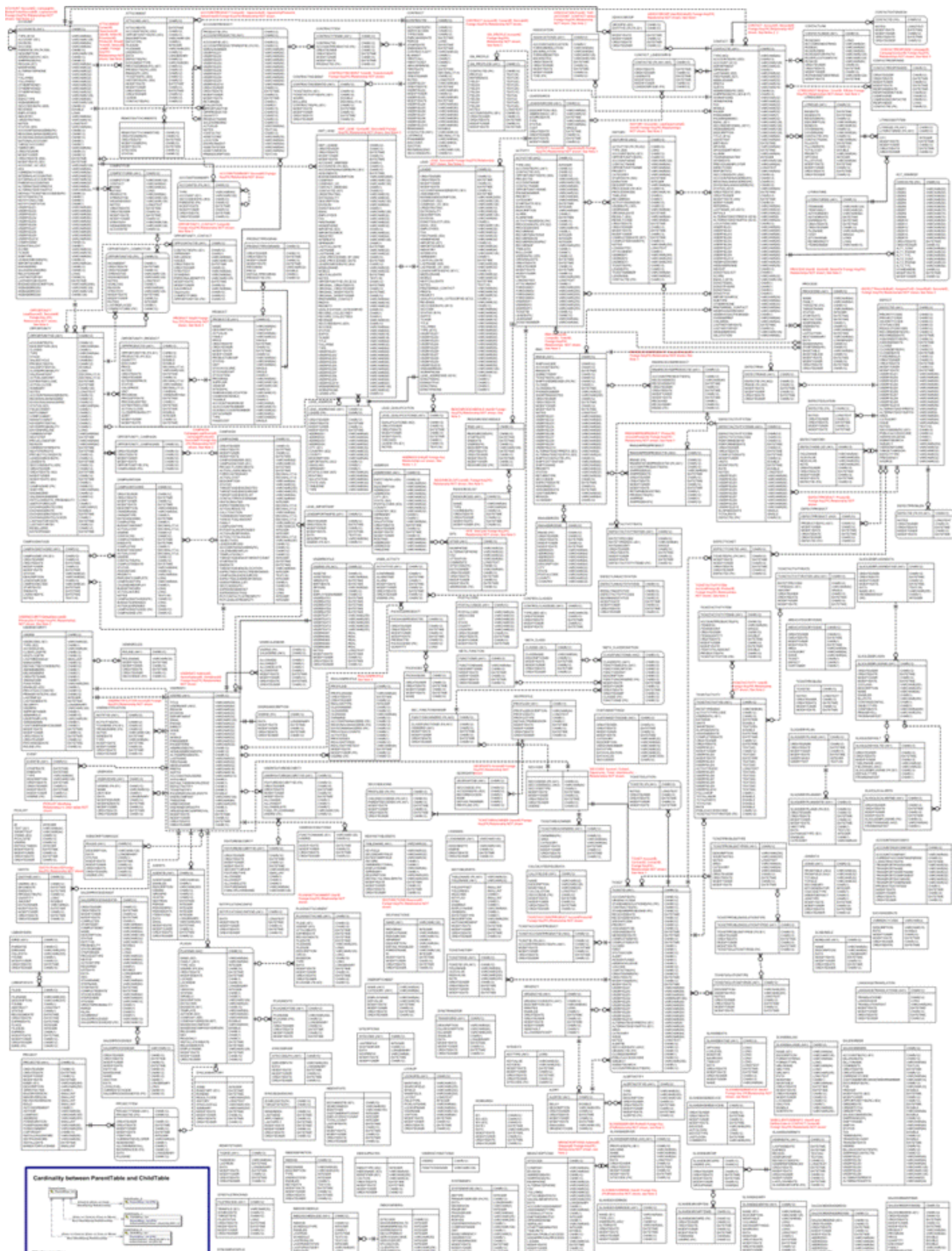
Pretpostavimo da aplikacija raste, tako da ovi ugostiteljski objekti mogu da imaju i druge namene kao što su dvorana za muziku uživo ili prostor za iznajmljivanje. Kada se radi integracija podataka preko weba, ili čak u manjim okruženjima koja se konstantno susreću sa novim skupovima podataka, migracija šema i podataka često ne može da se kontroliše.

4.1.4 VEOMA KOMPLIKOVANE ŠEME

Pored potrebe migracije kako se podaci razvijaju, drugi problem u relacionim bazama podataka je da šema može da bude veoma komplikovana. Slika 11 je primer male sekcije šeme za Customer Relationship Management (CRM) aplikaciju [29].

CRM sistem se koristi za skladištenje informacija o potencijalnim klijentima i odnosima sa postojećim klijentima. Ovo je velika aplikacija, ali je veoma mali deo u odnosu na to šta je potrebno za vođenje poslovanja. Enterprise Resource Planning (ERP) aplikacija kao što je SAP, može da pokrije mnogo više potreba za podacima za velika preduzeća. Šeme ERP aplikacija su nepristupačne, tako da postoji čitava industrija konsultantskih kompanija koje se isključivo bave njima.

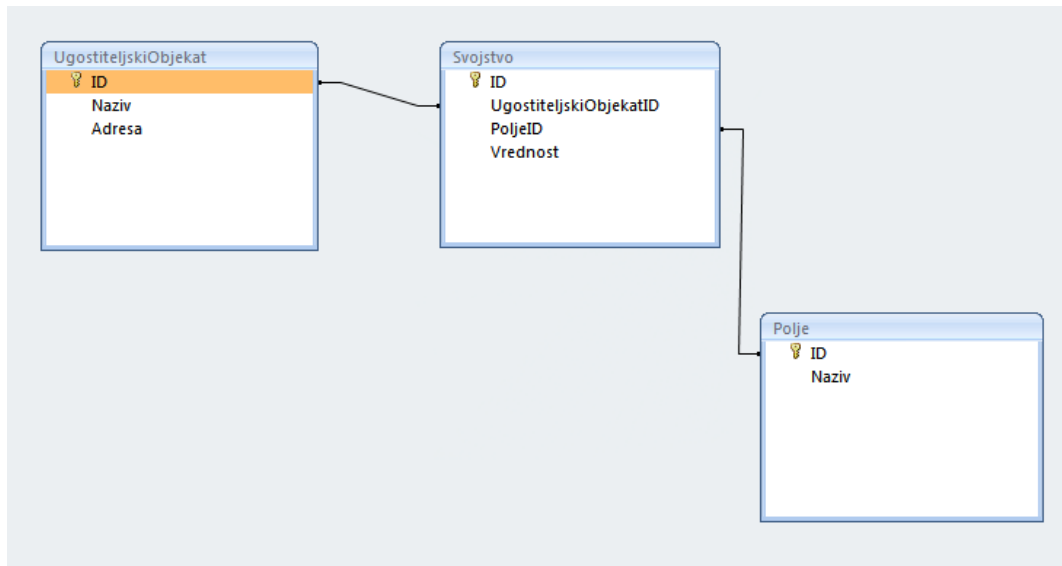
Složenost je jedva upravljiva u poznatom domenu industrije CRM i ERP-a, i ona postaje još veća u oblastima koje se ubrzano razvijaju kao što su biotehnologija i internacionalni razvoj. Umesto dugačkih listi dobro kategorizovanih podataka postoje stotine ili hiljade skupova podataka koji se odnose na različite stvari. Pokušavanje da se ti skupovi podataka normalizuju u jedinstvenu šemu je složen proces.



Sl. 11. Primer velike šeme

4.1.5 FLEKSIBILNE ŠEME

Prethodno razmatranje dovodi do pitanja da li je moguće definisati dovoljno fleksibilnu šemu koja će moći da rukovodi raznim tipovima podataka koji se stalno menjaju, a da još uvek održava određeni nivo čitljivosti. Šema može da se dizajnira da bude otvorena za nove svrhe ugostiteljskog objekta i da ponudi prilagođena polja za njih (kao na slici 12). Šema je izgubila koncept barova i restorana u potpunosti. Sada sadrži samo spisak ugostiteljskih objekata sa prilagođenim svojstvima za njih [29].



Sl. 12. Šema ugostiteljskih objekata sa kompletno prilagođenim svojstvima

Ovo se obično ne preporučuje jer se narušavaju normalizacije koje su bile moguće ranije i verovatno će da naruši performanse baze podataka. Međutim, omogućuje da se na jednostavan način dodaju nove svrhe ugostiteljskog objekta. Na slici 13 se vidi da tabela svojstvo sadrži sve prilagođene informacije za svaki ugostiteljski objekat.

ID	Naziv	Adresa	Add New Field
1 Deli Llama		Peachtree Rd	
ID	PoljeID	Vrednost	Add New Field
		1 Deli	
	2	2 \$	
*	(New)		
2 Peking Inn		Lake St	
ID	PoljeID	Vrednost	Add New Field
	3	1 Chinese	
	4	2 \$\$\$	
	5	3 Scorpion Bowl	
	6	4 No	
*	(New)		
3 Thai Tanic		Branch Dr	
(New)			

ID	Naziv	Add New Field
	1 Kuhinja	
	2 Cene	
	3 DJ	
	4 Specijalno piće	
*	(New)	

Sl. 13. Podaci o ugostiteljskim objektima u fleksibilnoj formi

Ovo znači da aplikacija može da se proširi, na primer, da uključi i koncertna mesta. Moguće je da se dodaju nova polja u tabelu “Polje” i da se onda dodaju prilagođena svojstva ugostiteljskom objektu. Na slici 14 se vidi primer gde je dodato da Thai Tanic ima uživo džez muziku. U tabeli “Polje” su dodata dva nova zapisa “Uživo muzika” i “Muzički pravac”.

ID	Naziv	Adresa	Add New Field
1	Deli Llama	Peachtree Rd	
2	Peking Inn	Lake St	
3	Thai Tanic	Branch Dr	

ID	PoljeID	Vrednost	Add New Field
1	1	Deli	
2	2	\$	
3	3	Chinese	
4	4	\$\$\$	
5	5	Scorpion Bowl	
6	6	No	
7	7	Peking Inn	
8	8	Lake St	
9	9	Yes	
10	10	Jazz	

ID	Naziv	Add New Field
1	Kuhinja	
2	Cene	
3	DJ	
4	Specijalno piće	
5	Uživo muzika	
6	Muzički pravac	

Sl. 14. Dodavanje novih svojstva ugostiteljskom objektu bez menjanja šeme

Ova proširiva šema koja koristi ključ/vrednost nije nova. Dosta “prilagodljivih” aplikacija kao što je salesforce.com koriste ovakvu šemu. Ovaj način, tabele u bazi podataka, stavlja po strani i zbog toga se javlja problem sa performansama, tako da ovo nije najbolja ideja. Gubi se dosta na normalizaciji jer se sve svodi na ključ/vrednost parove.

4.2 SEMANTIČKE VEZE

Iako ovo nije najbolja praksa, treba nastaviti sa ovim progresom i videti šta se dešava. Iz ove perspektive naziv i adresa ugostiteljskog objekta su samo svojstva, zato ih treba premestiti u tabelu “Svojstvo” [29].

Ugostiteljski	PoljeID	Vrednost	Add New Field
1	1	Deli	
1	2	\$	
1	7	Deli Llama	
1	8	Peachtree Rd	
2	1	Chinese	
2	2	\$\$\$	
2	3	Scorpion Bowl	
2	4	No	
2	7	Peking Inn	
2	8	Lake St	
3	5	Yes	
3	6	Jazz	
3	7	Thai Tanic	
3	8	Branch Dr	

ID	Naziv	Add New Field
1	Kuhinja	
2	Cene	
3	DJ	
4	Specijalno piće	
5	Uživo muzika	
6	Muzički pravac	
7	Naziv	
8	Adresa	
(New)		

Sl. 15. Svojstva ugostiteljskog objekta

Ovo je interesantno, ali relacija između tabela “Svojstvo” i “Polje” je i dalje poznata samo znanju u postojećem upitu. Ako se to znanje eksplicitno pridruži, dobiće se parametrizovani prikaz.

UgostiteljskiObjekatID	Polje	Vrednost	Add New Field		
1	Kuhinja	Deli			
1	Cene	\$			
1	Naziv	Deli Llama			
1	Adresa	Peachtree Rd			
2	Kuhinja	Chinese			
2	Cene	\$\$\$			
2	DJ	Scorpion Bowl			
2	Specijalno piće	No			
2	Naziv	Peking Inn			
2	Adresa	Lake St			
3	Uživo muzika	Yes			
3	Muzički pravac	Jazz			
3	Naziv	Thai Tonic			
3	Adresa	Branch Dr			

Sl. 16. Parametrizovani prikaz ugostiteljskih objekata

Sada je svaki podatak opisan svojstvima koja ga definišu. Na ovaj način je uzeta semantika iz tabela i kolona i predstavljena kao podatak u tabeli. Ovo je osnova semantičkog modelovanja podataka: fleksibilne šeme gde su veze opisane preko samih podataka.

4.3 METAPODATAK JE PODATAK

Jedan od izazova u korišćenju tuđih podataka je razumevanje kako su razne tabele povezane međusobno. Ove informacije o “podacima o predstavljanju podataka” se često nazivaju metapodatak i predstavljaju znanje o tome kako podaci mogu da se koriste [30]. Ovo znanje se uglavnom predstavlja eksplicitno u definiciji podataka kroz vezu sekundarnog ključa, ili implicitno u logici upita. Veoma često su podaci arhivirani, objavljeni ili deljivi bez ovih metapodataka. Iako otkriće ovih relacija može da bude zanimljiva vežba za korisnika, kod velikih šema ovo je skoro nemoguć posao [29].

U prethodnom primeru parametrizovani podaci ugostiteljskog objekta su napravili izuzetno fleksibilan model. Kada se uči o novim karakteristikama ugostiteljskog objekta dovoljno je da se doda novi zapis čak i ako karakteristike prethodno nisu poznate. Parametrizovanje podataka je omogućilo jednostavnu upotrebu. Potrebno je jako malo znanja o organizaciji podataka da bi ti podaci mogli da se koriste. Jednom kada je poznato da povezani redovi dele identičan UgostiteljskiObjekatID, moguće je da se sazna sve što je potrebno o ugostiteljskom objektu izborom svih redova sa istim UgostiteljskiObjekatID-em. Iz ove perspektive o parametrizovanim podacima treba razmišljati kao o “samo-opisujućim podacima”. Metapodaci relacione šeme koji opisuju koje kolone idu zajedno kako bi opisali jedan entitet su postali delovi samih podataka.

4.4 FLEKSIBILNI MODEL PODATAKA

Pakovanjem podataka i metapodataka zajedno u jednom predstavljanju nije samo napravljena šema budućnosti, već je aplikacija izolovana od “poznavanja” previše o obliku podataka. Jedina stvar koju aplikacija treba da zna o podacima je da ugostiteljski objekti imaju ID u prvoj koloni, svojstva u drugoj i vrednosti svojstva u trećoj koloni. Kada se doda totalno novo svojstvo ugostiteljskom objektu, aplikacija može da odluči da li će ignorisati svojstvo ili će rukovati sa njim na uobičajen način [29].

Predstavljanje podataka fleksibilnim modelom omogućava da neko drugi jednostavno intergiše informacije o lokacijama aparata za espresso, čime dozvoljava aplikaciji da pokrije ne

samo restorane i barove, već i kafiće, knjižare i benzinske pumpe. Dobro dizajnirana aplikacija bi trebalo da bude u mogućnosti da na jednostavan način integriše nove semantičke podatke, i semantičke baze podataka bi trebalo da su u stanju da rade sa raznim aplikacijama.

Mnogi alati za kreiranje sadržaja i slika sada podržavaju Extensible Metadata Platform (XMP) podatke za praćenje informacija o autoru i licenci kreativnog sadržaja. XMP standard razvijen od strane Adobe Systems obezbeđuje standardan skup šema i omogućava korisniku da proširi model podataka na isti način kao što su prošireni podaci o ugostiteljskim objektima. Korišćenjem samo-opisujućeg modela, alat ne mora da promeni način kontrole sadržaja XMP-a, i nije potrebna promena čak i ako se sadržaj značajno promeni. Kako su alati za kreiranje slika u osnovi za kreativno izražavanje, neophodno je da korisnici ne budu ograničeni fiksnim skupom opisnih polja.

4.5 VEČNA BETA

Jasno je da je web promenio ekonomiju razvoja aplikacija. Web aplikacije su značajno smanjile troškove izbacivanjem kompleksnih mrežnih poruka niskog nivoa. Aplikacija sposobna da opslužuje više korisnika na Internetu, sa krajnjim rokom razvoja softvera koji je određen proizvodnjom i kanalima distribucije se više ne koristi. Aplikacije su sada u mogućnosti da se poboljšaju na kontinuiranoj i nezavisnoj osnovi. Razvojni ciklusi kao što su ažuriranje aplikacije na mesečnom, nedeljnom ili dnevnom nivou se više ne smatraju neobičnim. Fraza “večna beta” odražava da softver nikad nije “zamrznut” na webu [29].

Procesi malih ažuriranja su osnova konkurentnosti na velikim portalima. Na primer, Yahoo ima širok spektar medijskih sajtova koji pokrivaju teme o zdravlju, deci, putovanjima, filmovima. Sadržaj se kontinuirano menja kada stigne nova vest, kada se uređivački proces završi, i kada korisnik prokomentariše informaciju. U nastojanju da se smanji vreme potrebno za kreiranje specijalizovanih sajtova i da se omoguće novi tipovi personalizacije i pretrage, Yahoo! je počeo da dodaje semantičke metapodatke svom sadržaju koristeći proširive šeme. Kako podaci i metapodaci postaju jedno, nove aplikacije mogu da dodaju svoje komentare bez modifikacije osnovne šeme. Ova sloboda proširenja postojećih metapodataka omogućava konstantni razvoj funkcija bez uticaja na zastarevanje aplikacija, i omogućuje jednoj aplikaciji da ima korist od informacija iz druge.

Ovo dovodi do stalnog poboljšanja i razvoja aplikacija praćenih većim interesom za “script” jezike kao što su Python, Perl i Ruby. Jednostavnost puštanja u rad neke aplikacije sa minimalnim dizajnom i lakoćom dodavanja novih funkcija daje ovim “script” jezicima prednost u odnosu na statičke jezike koji su više dizajnirani za softversko inženjerski pristup. Međutim, većina okruženja koja koristi ove jezike još uvek zavisi od baza podataka za skladištenje podataka, i još uvek zahteva unapred poznat model podataka i posvećenost šemi koja ne podržava nove izvore podataka koje će buduće aplikacije da zahtevaju. Iako je večna beta od velike koristi za korisnike, ubrzani razvoj ciklusa može da bude izazov za upravljanje podacima. Kako se razvijaju nove funkcije aplikacije, tako se šema podataka takođe često menja. Fleksibilne semantičke strukture podataka i aplikacijski obrasci koji rade sa njima su dobro dizajnirani za korišćenje u svetu večne bete.

5. IZRAŽAVANJE ZNAČENJA

U trenutnom obliku “potpuno parametrizovani ugostiteljski objekat” tabela omogućava predstavljanje proizvoljnih činjenica o hrani i muzici. Ne postoji ništa specifično u formi tabele da ograniči podatke samo na ugostiteljske objekte, na ovaj način postoji mogućnost predstavljanja činjenica i o drugim entitetima u istom formatu tri-kolone.

Ovaj format tri-kolone je poznat kao triple, i čini fundamentalni element semantičke reprezentacije. Svaki triple se sastoji od subjekta, predikta i objekta. Tripleti mogu da se posmatraju kao jednostavni jezički iskaz, gde svaki element odgovara delu gramatike koja se koristi za dijagram kratke rečenice [31].



Sl. 17. Dijagram rečenice koji prikazuje subjekat-predikat-objekat odnos

Generalno, subjekat u triplu odgovara entitetu “stvar” za koji postoji konceptualna klasa. Ljudi, mesta i drugi konkretni objekti su entiteti kao i manje konkretne stvari kao što su vremenski periodi i ideje. Predikat je svojstvo entiteta kojem je priključen. Ime osobe, datum rođenja ili poštanska adresa su primeri predikta. Objekti spadaju u dve klase, entiteti koji mogu da budu subjekti u drugim tripletima ili literalne vrednosti kao što su stringovi ili brojevi.

Višestruki tripleti mogu da se povezuju pomoću istog subjekta i objekta u različitim tripletima, spajanje ovih lanaca odnosa formira usmereni graf. Usmereni grafovi su dobro poznata struktura podataka u računarskim naukama i matematici, i mogu se koristiti za predstavljanje podataka.

Model grafa može da se primeni na podatke o ugostiteljskim objektima. Moguće je dodati na primer informacije o lokaciji.

Subjekat	Predikat	Objekat
S1	Kuhinja	“Deli”
S1	Cene	“\$”
S1	Naziv	“Deli Llama”
S1	Adresa	“Peachtree Rd”
S2	Kuhinja	“Chinese”
S2	Cene	“\$\$\$”
S2	Specijalno piće	“Scorpion Bowl”
S2	DJ	“No”
S2	Naziv	“Peking Inn”
S2	Adresa	“Lake St”
S3	Uživo muzika	“Yes”
S3	Muzički pravac	“Jazz”
S3	Naziv	“Thai Tanic”
S3	Adresa	“Branch Dr”
S4	Naziv	“North Beach”

S4	se nalazi u	“San Francisco”
S5	Naziv	“SOMA”
S5	se nalazi u	“San Francisco”
S6	Naziv	“Gourmet Ghetto”
S6	se nalazi u	“Berkeley”

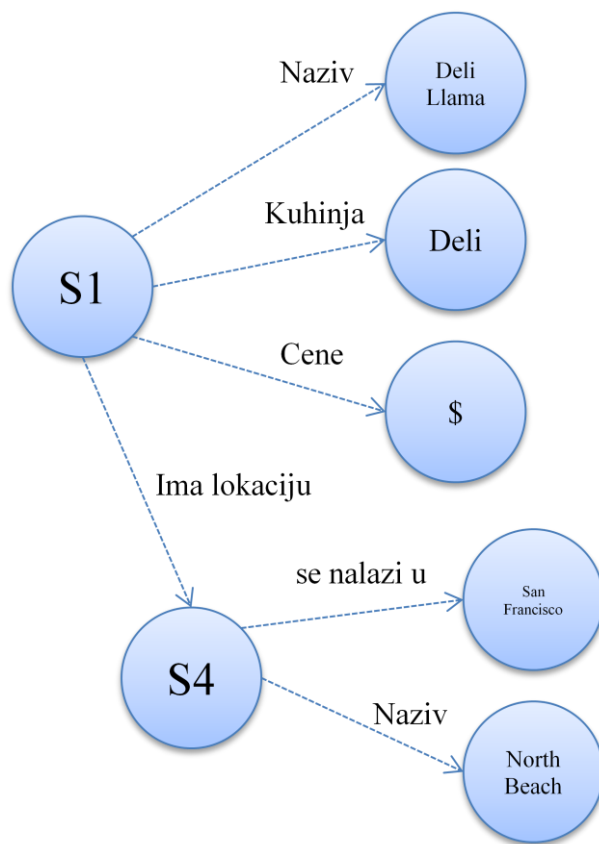
Tabela 4. Proširena tabela ugostiteljskih objekata sa lokacijom

Ugostiteljski objekti i lokacije su predstavljeni koristeći isti model, ali ih ništa ne povezuje. Kako objekti iz jednog tripla mogu da budu subjekti u drugom, moguće je dodati tvrdnju koja određuje koji se ugostiteljski objekat gde nalazi.

Subjekat	Predikat	Objekat
S1	Ima lokaciju	S4
S2	Ima lokaciju	S6
S3	Ima lokaciju	S5

Tabela 5. Povezivanje susednih ugostiteljskih objekata

Slika 18 je dijagram tripleta strukturiranih u graf, sa subjektom i objektom kao čvorovima i prediktom kao usmerenom granom.

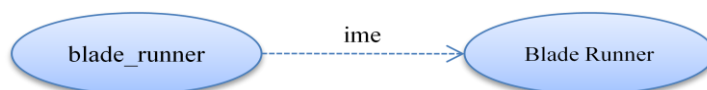


Sl. 18. Graf tripleta koji prikazuje informacije o ugostiteljskom objektu

Sada prateći lanac utvrđivanja, može da se dođe do informacije da je moguće da se jede jeftino u San Francisku. Samo je potrebno znati gde da se pogleda.

5.1 PRIMER: PODACI O FILMOVIMA

Moguće je koristiti ovaj triple model za izradu jednostavnog prikaza filma. Na početku je potrebno prikazati naziv filma Blade Runner sa triplom blade_runner ime “Blade Runner”. Ovaj triple je grana koja predstavlja predikat koji se zove “ime” i koji povezuje subjekat blade_runner sa objektom “Blade Runner” [31].

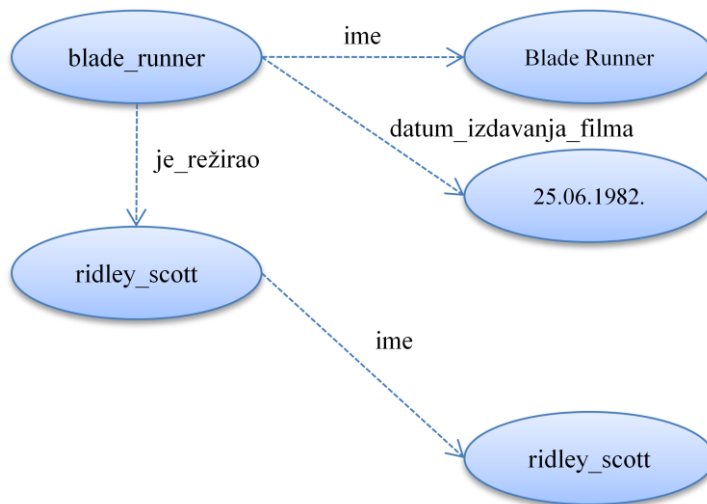


Sl. 19. Triple koji opisuje naziv filma Blade Runner

Sada je potrebno dodati datum izdavanja filma. Ovo je moguće sa drugim triplom blade_runner datum_izdavanja_filma “25.06.1982.”. Koristi se isti ID blade_runner koji govori da se ovaj iskaz odnosi na isti subjekat. Ovo je potrebno kako bi se izgradio graf, u suprotnom bi postojala gomila nepovezanih činjenica i nikako ne bi znali da se odnose na isti entitet.

Potrebno je utvrditi da je Ridley Scott režirao film. Najjednostavniji način je da se doda triple blade_runner je_režirao “Ridley Scott”. Postoji problem sa ovim pristupom, Ridley Scottu nije dodat ID, tako da nije moguće povezivanje njega i drugih filmova koje je on režirao. Takođe, ako postoji neki drugi Ridley Scott, neće postojati mogućnost razlike među njima po imenu.

Ridley Scott je osoba i režiser što ga kvalifikuje da bude entitet. Ako mu se dodeli ID ridley_scott, moguće je utvrditi još neke činjenice o njemu: ridley_scott ime “Ridley Scott” i blade_runner je_režirao ridley_scott. Primećuje se da je svojstvo ime iskorišćeno ponovo. Oba entiteta “Blade Runner” i “Ridley Scott” imaju imena, tako da korišćenje svojstva ime dok god je u skladu sa pravilnom upotrebom ima smisla.



Sl. 20. Graf koji opisuje film Blade Runner

6. TEORIJA GRAFOVA

Teorija grafova je samostalan i veoma važan deo matematike. Grafovi su posebno zanimljivi jer je pomoću njih moguće jednostavno modelovanje složenih problema, kao što su semantički podaci, računarske mreže, postavljanje saobraćajnica, postavljanje električnih mreža i sl.

Različite diskretne strukture koje se pojavljuju u računarstvu pogodno se opisuju grafovima.

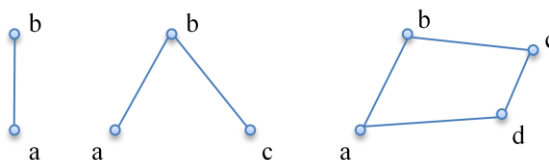
Prvi problem i njegovo rešenje izvedeno na način koji je bio drugačiji u odnosu na prethodne probleme i koji se može smatrati pretečom teorije grafova jeste rad Leonarda Ojlera pod nazivom „Sedam mostova Kenigsberga“, objavljen 1736. godine [32].

Fransis Gutri je 1852. godine izložio problem četiri boje, koji postavlja pitanje da li je moguće obojiti zemlje na geografskoj karti sa samo četiri boje, a da se ne pojave dve susedne zemlje obojene istom bojom. Ovaj problem su rešili tek 1976. godine Kenet Apel i Wolfgang Haken. Postavljanje ovog problema se smatra početkom teorije grafova. Rešavanjem ovog problema otkrivene su mnoge teoreme i postavljeni mnogi teoretski pojmovi i koncepti [33].

6.1 OSNOVNI POJMOVI

Graf je apstraktni matematički objekat.

Grafovi su sastavljeni od čvorova i grana koje ih povezuju.

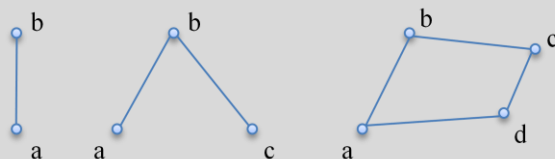


Sl. 21. Skupovi čvorova i grana

Skup čvorova se najčešće obeležava sa V , a skup grana sa E .

Graf je zadat ako su poznata dva skupa, skup čvorova V i skup grana E .

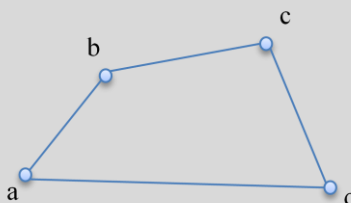
Definicija: Graf $G = (V, E)$ je uređeni par koji se sastoji od skupa čvorova V i skupa grana $E \subseteq \binom{V}{2}$.
Primer: • Čvorovi mogu biti gradovi, a grane putevi između njih; • Čvorovi mogu biti računari, a načini komunikacija između njih grane.
Primer: • Dat je skup $V = \{a, b\}$ i $E = \{\{a, b\}\}$; • Dat je skup $V = \{a, b, c\}$ i $E = \{\{a, b\}, \{b, c\}\}$; • Dat je skup $V = \{a, b, c, d\}$ i $E = \{\{a, b\}, \{b, c\}, \{a, d\}, \{c, d\}\}$.



Sl. 22. Skupovi čvorova i grana

Grafovi su relacione strukture koje se jednostavno mogu predstaviti crtežima.

Primer:



Sl. 23. Primer crteža grafa

Graf sa slike može da se napiše kao relacija: $G = \{\{a, b, c, d\}, \{a, b\}, \{b, c\}, \{c, d\}\}$.

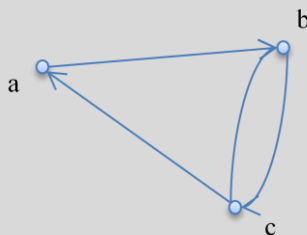
- Dve grane su susedne ako imaju isti čvor;
- Čvorovi jedne grane nazivaju se krajevi;
- Grana koja spaja čvor sa samim sobom naziva se petljom;
- Graf koji nema nijednu petlju naziva se prostim grafom;
- Neorijentisani graf $G = (V, E)$ je uređen skup parova, čvorova i grana gde je $E \subseteq \binom{V}{2} \cup V$, sledi graf može da ima i petlje.

Napomena: Prost graf je ustvari neorijentisana putanja bez petlji;

- Orijetisani graf ili digraf $G = (V, E)$ je uređen skup parova čvorova i grana gde je $E \subseteq V \times V$. Orijetisana grana $v = (a, b)$ ima početni čvor u a i krajnji u b ;

Primer:

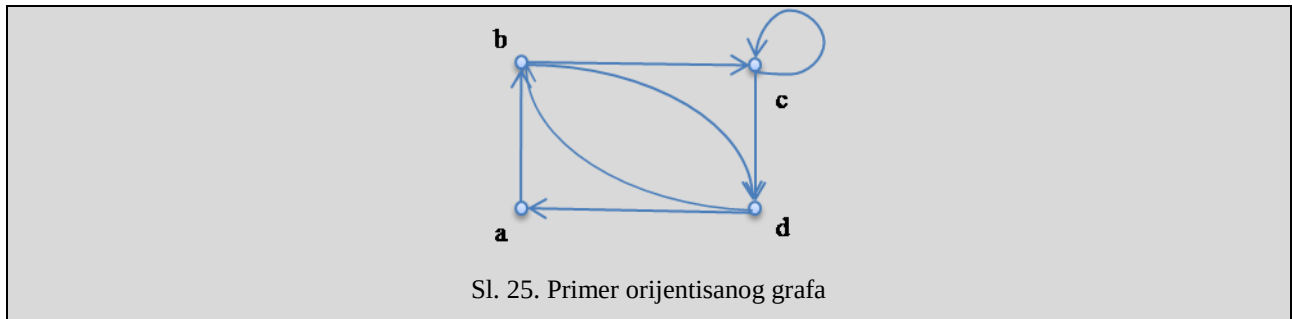
- Digraf koji sadrži skup $V = \{a, b, c\}$ i skup $E = \{(a, b), (b, c), (c, b), (c, a)\}$.



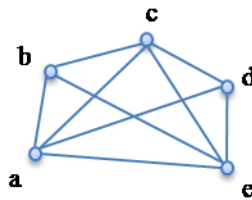
Sl. 24. Primer orijentisanog grafa

Primer:

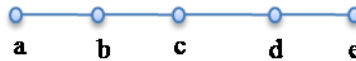
- Digraf koji sadrži skup $V = \{a, b, c, d\}$ i skup $E = \{(a, b), (b, c), (c, c), (b, d), (d, b), (c, d), (d, a)\}$.



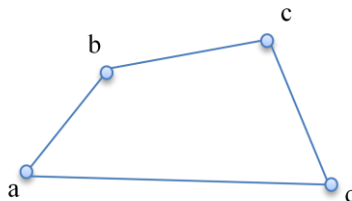
- Multigraf je graf kod koga između dva čvora a i b postoji više od jedne grane, koje polaze iz a i završavaju u b ;
- Kompletan ili potpun graf je onaj graf kod koga su svaka dva čvora povezana granom;



- Putanja je niz grana koje su međusobno povezane;

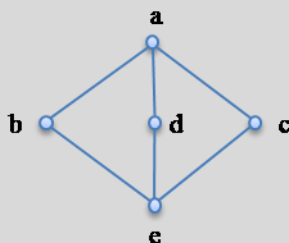


- Elementarna putanja je putanja koja kroz svaki čvor grafa prolazi najviše jedanput;
- Ciklus je graf koji se dobija od putanje, dodavanjem grane koja spaja krajeve putanje. Ciklus se često naziva i konturom;



- Prazan graf je graf sa čvorovima koji ne sadrži ni jednu granu;
- Stepen grafa je broj grana grafa koje imaju kraj u jednom čvoru. Čvor stepena 1 naziva se izolovani čvor ili list;

Primer:



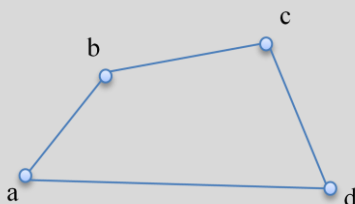
Sl. 29. Primer stepena grafa

- U grafu na slici čvorovi a i c su susedni, kao i grane ab , ad i ac .
- Čvorovi a i e nisu susedni, kao ni grane ac i be .
- Čvorovi b, c, d su stepena 2, a čvorovi a i e stepena 3.

- Graf je regularan ako su svi čvorovi istog stepena.

Primer:

- Regularan graf (svi čvorovi su stepena 2).



Sl. 30. Primer regularnog grafa

Kao što je istaknuto u prethodnom odeljku višestruki tripleti mogu da se povezuju pomoću istog subjekta i objekta u različitim tripletima, spajanje ovih lanaca odnosa formira usmereni graf.

Spektralna teorija grafova se koristi za rešavanje velikog broja problema u računarskim naukama, numeričkoj matematici, fizici, biologiji, hemiji i mnogim drugim naučnim i tehničkim disciplinama. Glavni problemi analize grafa su pronalaženje spektra, tj eigenvalues (sopstvene vrednosti) i eigenvectors (sopstveni vektori) za dati graf, kao i karakteristični i minimalni polinomi. Nijedan od ovih zadataka nije lak i veoma se retko mogu naći rešenja u zatvorenoj formi. Kada se izračunavaju eigenvalues, tj nule odgovarajućeg karakterističnog polinoma, često se koriste numeričke metode. Takođe, veoma značajnu ulogu igraju nejednakosti koje uključuju eigenvalues ili karakteristične polinome [90].

Kako se svaka računarska mreža može predstaviti grafom gde su čvorovi računari a grane veze (konekcije) između računara, iz određene matrice susedstva datog grafa se mogu izračunati eigenvalues kao veoma interesantni parametri za rešavanje različitih problema u aplikacijama. Kao na primer, maksimalna eigenvalue λ matrice susedstva igra važnu ulogu u modelovanju širenja virusa u računarskim mrežama. Što je manja maksimalna eigenvalue, veća je robusnost mreže u odnosu na širenje virusa. A što je veća maksimalna eigenvalue računarska mreža je pogodna za širenje znanja.

7. STANDARDI

7.1 PRISTUP KOJI KORISTI SLOJEVE

Razvoj semantičkog weba se kreće u koracima, svaki korak gradi sloj na vrhu prethodnog. Pragmatično opravdanje za ovakav pristup je jednostavnije postizanje konsenzusa u malim koracima. Obično postoji nekoliko istraživačkih grupa koje se kreću u različitim pravcima. Ovo takmičenje u idejama je glavna pokretačka sila za naučni napredak. Međutim, iz inženjerske perspektive postoji potreba za standardizacijom. Tako da ako se većina istraživača slaže sa određenim pitanjima i ne slaže sa drugim, ima smisla da se napravi tačka sporazuma. Na ovaj način čak i ako ambiciozna istraživanja propadnu, ostaće barem delimično pozitivni rezultati.

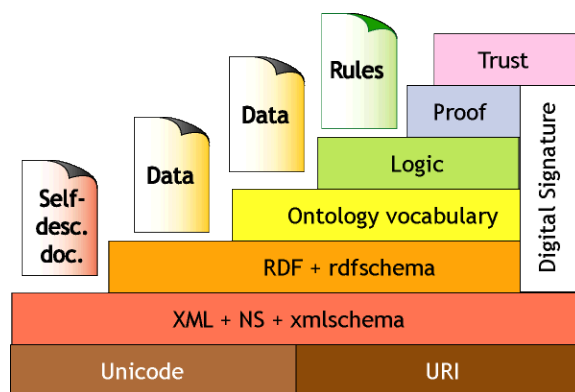
Kada se uspostave standardi, biće prihvaćeni od strane mnogo više grupa i kompanija, za razliku od čekanja da vide da li će alternativna istraživanja na kraju postići uspeh. Priroda semantičkog weba je takva da kompanije i krajnji korisnici moraju da izrade alate, dodaju sadržaj i koriste taj sadržaj. Ne može da se čeka do punog razvoja vizije semantičkog weba. Možda je potrebno još deset godina da bi se realizovao u punom obimu (kao što se danas predviđa).

Treba slediti dva principa u izgradnji jednog sloja semantičkog weba na vrhu prethodnog:

- Silazna kompatibilnost. Agenti treba da budu u potpunosti svesni sloja na nižem nivou i da budu u mogućnosti da interpretiraju i koriste napisanu informaciju. Na primer, agenti svesni semantike OWL-a mogu da preuzmu punu prednost informacija napisanih u RDF-u ili RDF Schema-i;
- Uzlazno delimično razumevanje. Dizajn treba da bude takav da agenti budu u potpunosti svesni sloja na višem nivou i da budu u mogućnosti da makar parcijalno iskoriste informaciju. Na primer, agent svestan samo RDF i RDF Schema semantike može parcijalno da interpretira znanje zapisano u OWL-u bez obzira na elemente koji idu dalje od RDF-a i RDF Schema-e. Naravno nije potrebno da svi alati podržavaju ovu funkcionalnost, poenta je da ova opcija treba da bude omogućena.

Dok su ove ideje teorijski privlačne i dok se koriste kao vodeći principi za razvoj semantičkog weba, njihova realizacija u praksi se ispostavila kao teška, i neki kompromisi moraju da se naprave.

Slika 31 Predstavlja “layer cake” koji opisuje glavne slojeve dizajna i vizije semantičkog weba.



Sl. 31. “layer cake” semantičkog weba [34]

Na dnu se nalazi XML, jezik koji dozvoljava pisanje strukturiranog web dokumenta sa rečnikom definisanim od strane korisnika. XML je naročito pogodan za slanje dokumenata preko weba.

RDF je osnovni model podataka, kao model entitet-odnos za pisanje jednostavnih izjava o web objektima (resursima). RDF model ne zavisi od XML-a, ali RDF ima sintaksu zasnovanu na XML-u. Zato je lociran iznad XML sloja.

RDF Schema omogućava modelovanje primitiva za organizovanje web objekata u hijerarhiju. Ključne primitive su klase i svojstva, odnosi podklase i podsvojstva, i delokrug i niz ograničenja. RDF Schema je bazirana na RDF-u.

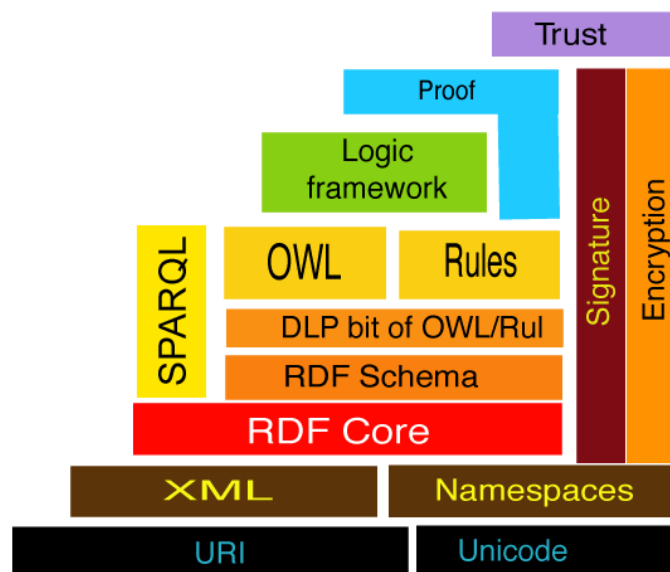
RDF Schema može da se posmatra kao primitivni jezik za pisanje ontologija. Ali postoji potreba za moćnijim jezicima za ontologiju koji proširuju RDF Schema-u i omogućavaju predstavljanje kompleksnijih odnosa između web objekata.

Logic sloj se koristi za poboljšanje ontologijskih jezika i dozvoljava pisanje deklarativnog znanja specifičnog za aplikaciju.

Proof sloj se odnosi na aktuelni deduktivni proces, kao i predstavljanje dokaza u web jezicima (iz nižeg nivoa) i dokaz valjanosti.

Konačno, Trust sloj će se pojaviti kroz korišćenje digitalnih potpisa i druge vrste znanja na osnovu preporuka od strane pouzdanih agenata, rejting i sertifikovanih agencija i tela potrošača. Ponekad izraz “Web poverenja” govori da će poverenje biti organizovano na isti distribuirani i haotičan način kao WWW. Kako se nalazi na vrhu piramide, poverenje na visokom nivou je ključni koncept. Web će postići svoj puni potencijal kada korisnici imaju poverenja u njegovo funkcionisanje (sigurnost) i kvalitet informacija koji pruža.

Sledeća slika prikazuje alternativni stek slojeva koji uzima u obzir trenutni razvoj. Glavne razlike u odnosu na klasični stek su:



Sl. 32. Alternativni “layer cake” semantičkog weba [35]

- Sloj ontologije se instancira sa dve alternative. Trenutni standardni web jezik ontologije OWL i jezik zasnovan na pravilima. Tako se pojavljuje alternativni tok u

razvoju semantičkog weba;

- DLP (**D**escription **L**ogic **P**rograms) je ukrštanje OWL-a i Horn logike, i služi kao zajednički osnov.

Arhitektura semantičkog weba nije statična već se stalno menja i razvija.

7.2 RDF

Sada kada postoji razumevanje semantike i kako je moguće izgraditi jednostavan triple, potrebno je predstaviti svet formata podataka i semantičkih standarda. Na početku je potrebno da se odgovori na par pitanja:

- Koji predikat (glagol) treba da se koristi?
- Kako da se sazna da li je vrednost podatka koji je broj, float ili string?
- Zašto je „Ridley Scott“ predstavljen kao `ridley_scott`, a ne drugačije?
- Da li su zarezom odvojeni tripleti zaista najbolji način za skladištenje i razmenu podataka?

Ova pitanja se odnose na održavanje i razmenu velikog broja semantičkih podataka. U ovom poglavlju će biti objašnjeno nekoliko koncepata koji mogu da pomognu u rešavanju ovih problema:

- URIs i strong keys, kako bi bili sigurni da se priča o istoj stvari svuda;
- RDF serijalizacije, koji se to parseri za svaki popularan jezik koriste za standardan način predstavljanja i razmenu semantičkih podataka;
- SPARQL jezik, standardan način za pretraživanje semantičkih podataka.

Upotreba semantičkog web formata ide daleko izvan ovih koncepata, ali oni su jedini koji su stvarno potrebni za izgradnju semantičkih aplikacija [36].

7.2.1 ŠTA JE RDF?

Strukturiranje podataka u grafove je dobra ideja. Ali kada postoje podaci učitani u graf kako omogućiti razmenu i dostupnost drugim korisnicima? Odgovor je W3C Resource Description Framework (RDF). RDF omogućava standardni način za izražavanje grafova podataka i njihovu razmenu sa drugim korisnicima, i što je još važnije, mašinama. RDF je W3C „preporuka“ i postoji velika kolekcija alata i servisa koji su nastali oko RDF-a.

Istraživanja RDF-a počinju 1990. kada je Tim Berners-Lee napisao predlog koji je doveo do razvoja WWW. U prvobitnom predlogu je bilo različitih vrsta veza između dokumenata, što je omogućilo računarima da automatski shvate hipertekst. Ispisane veze nisu bile uključene u prvu HTML specifikaciju, ali se ideja pojavila iz Meta Content Framework-a (MFC). Sistem za opisivanje metapodataka i organizaciju weba je razvio Ramanathan Guha dok je radio u Apple-u i Netscape-u kasnih 1990.-tih, sa XML reprezentacijom koju je razvio Tim Bray. W3C je razmatrao generičku reprezentaciju metapodataka, i mnoge ideje pronađene u MFC-u su ugrađene u prvu RDF W3C preporuku iz 1999. Od tada su standardi revidirani, a današnji softveri i alati odražavaju ova poboljšanja.

7.2.2 *RDF MODEL PODATAKA*

RDF je jezik za izražavanje modela podataka pomoću iskaza izraženih kao triple. Svaki iskaz se sastoji od subjekta, predikta i objekta. RDF dodaje nekoliko važnih koncepata koji ovaj model čine mnogo preciznijim i robusnijim. Ovi dodaci igraju važnu ulogu u otklanjanju nejasnoća pri slanju semantičkih podataka između mašina koje nemaju nikakva međusobna saznanja.

URI kao jaki ključ

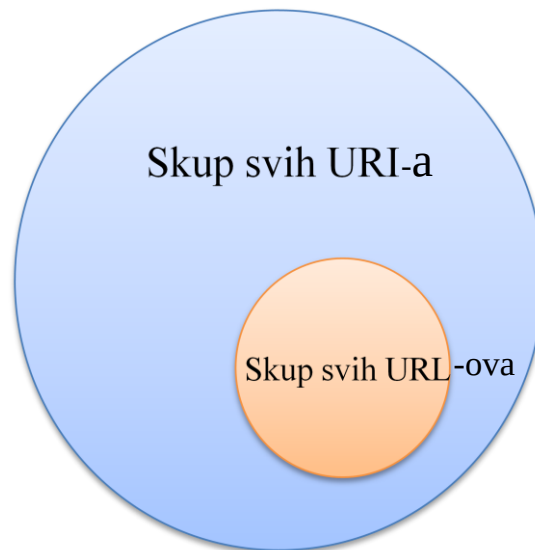
U poglavlju 4. prilikom spajanja više tabela kod relacionih podataka potrebno je bilo da se nađe zajednički identifikator između dve tabele koji dozvoljava mapiranje tabela. Ovi zajednički identifikatori se nazivaju ključevi i oni su osnov za pridruživanje relacionih podataka.

Slično, u grafu strukture podataka, potrebno je dodeliti jedinstveni identifikator za svaki čvor, tako da se odnosi na njih kroz sve triple koji opisuju njihove relacije. Do sada su korišćeni stringovi za obeležavanje čvorova u grafovima, i pažljivo su korišćena jedinstvena imena gde god je postojala šansa preklapanja.

Za velike aplikacije i tamo gde se koordinira sa više skupova podataka, teže je garantovati jedinstven i dosledan identifikator za svaki čvor. Primer bi mogao biti: ako bi se pokušalo objedinjavanje baza podataka starih klasičnih filmova sa bazom podataka novijih filmova. Baza podataka klasičnih filmova bi imala zapis za Harrison Forda, zvezdu nemih klasika kao što je „Oh, Lady, Lady“. Ako bi ID ovog glumca bio harrison_ford, u objedinjenoj bazi podataka bi otkrili da je glumac koji je glumio Han Sola i Indijana Džonsa poginuo u saobraćajnoj nesreći 1957.

Resursi

Da bi se izbegla ova vrsta nejasnoće, RDF konceptualizuje sve u univerzumu kao resurs. Resurs je jednostavno sve što može da se identifikuje sa Univerzal Resource Identifier (URI). Uglavnom su svi upoznati sa URL-ovima (Universal Resource Locators), ili stringovima koji preciziraju kako se preuzimaju web stranice. URL je podskup URI-a koji identifikuje odakle mogu da se preuzmu digitalne informacije. URL definiše gde je moguće pronaći specifične informacije, a takođe pruža jedinstveni identifikator informacije. URI-i generalizuju ovaj koncept.



Sl. 33. Dijagram koji pokazuje odnos URL-ova i URI-a.

S obzirom da URI može da identifikuje sve kao resurs, subjekat i objekat u RDF iskazu mogu da budu resursi, a predikat u RDF iskazu je uvek resurs. URI resursa predstavljenog u RDF iskazu se naziva URI reference (URIref) za taj čvor u grafu. Upotreba fraze „URI reference“ može lako da izgubi smisao. Čvor u grafu nije stvar koju URI identifikuje, već je URI identifikator nečega što je samo predstavljeno kao čvor u grafu. Tako se kaže da čvor u grafu ima referencu koja je identifikovana URI-jem.

URI omogućuje distribuiranim organizacijama kreiranje imena za stvari za koje su zainteresovani, moguće je izbeći problem da dve grupe izaberu isto ime za različite entitete. URI su jednostavni stringovi, sastavljeni od naziva šeme praćeni dvotačkom, dve kose crte (://) i šema-specifičnim identifikatorom. Šema identifikuje protokol URI, dok se šema-specifičan identifikator koristi od strane protokola šeme da jedinstveno identifikuje resurs. Uglavnom svi URI koriste „http“ ili „https“ šemu za identifikaciju. Na osnovu iskustva sa URL-om, šema-specifični deo ovih identifikatora obično poprima formu naziv hosta, opcioni port identifikator i hijerarhijsku putanju. Ako se organizacije drže kreiranja identifikatora koristeći nazive hostova koji su u njihovoj kontroli, onda je nemoguće da dve organizacije izgrade iste identifikatore za dve različite stvari. Isto tako, organizacija može da podeli sopstveno kreiranje identifikatora na različite nazive hostova i/ili različite hijerarhijske putanje.

URI se smatraju jakim identifikatorima iz razloga što jedinstveno identifikuju resurse. Oni uvek predstavljaju istu stvar bez obzira na kontekst u kojem se nalaze.

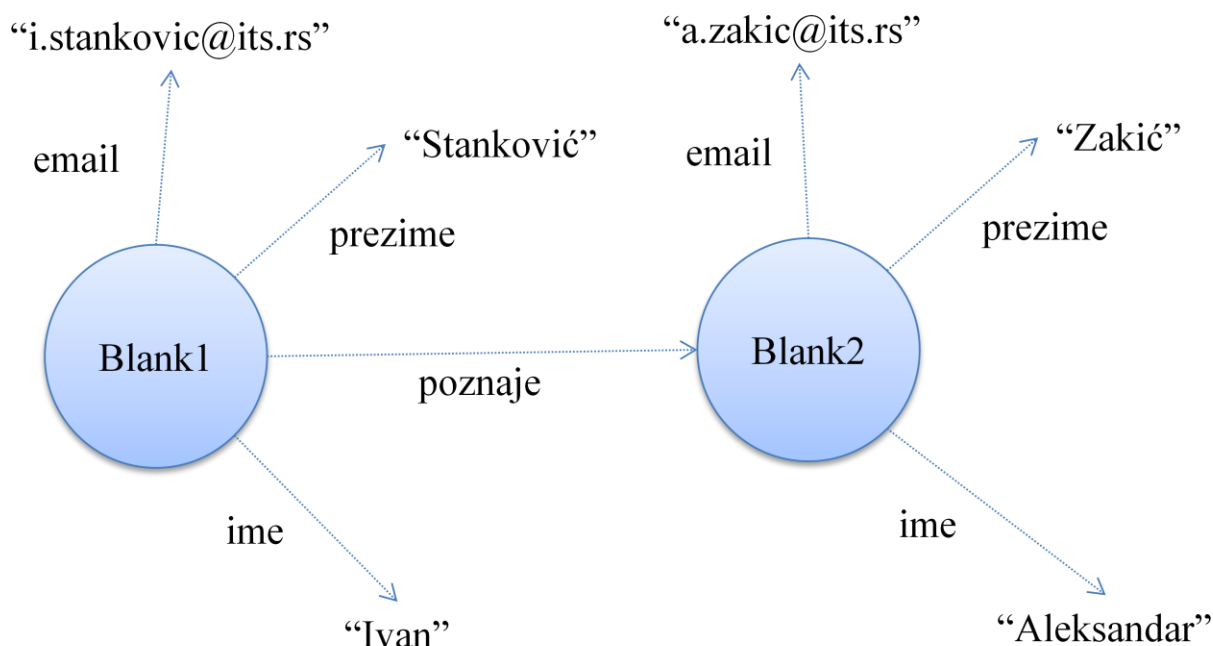
Važno je da se napomene da URI nije URL (iako je svaki URL URI). Praktično govoreći, ovo znači da ne bi trebalo pretpostaviti da će URI proizvoditi bilo kakve informacije ako se upiše identifikator u web pretraživač. Ali kreiranje digitalnih informacija o resursu dostupnom na adresi identifikatora se smatra dobrom praksom.

Uobičajeno je u RDF-u skraćivanje URI-a dodeljivanjem prostora imena osnovi URI-a i pisanje samo karakterističnog dela identifikatora. Na primer, rdf se često koristi kao nadimak osnove URI-a: <http://www.w3.org/1999/02/22-rdfsyntax-ns#>, dozvoljavajući prediktu kao što je <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> da se skрати kao rdf:type.

Prazni čvorovi

U svim diskusijama o URIsima i resursima, se ne koristi „Svi RDF subjekti su resursi“. Ovo je zato što postoje situacije gde nije poznat URI resursa na koje treba da se referencira ili gde ne postoji identifikator. Za ovakve situacije RDF obezbeđuje „anonimne“ ili prazne (blank) čvorove.

Prazni čvorovi su čvorovi grafa koji predstavljaju subjekat (ili objekat) za koji je potrebna tvrdnja, ali nije moguć pravi URI. Na primer mnogi API socijalnih mreža ne koriste „jake“ identifikatore za svoje članove. Umesto toga socijalne mreže koriste prazne čvorove da predstavljaju članove, i činjenice o članovima su povezane na prazan čvor.



Sl. 34. Prazni čvorovi u grafu socijalne mreže

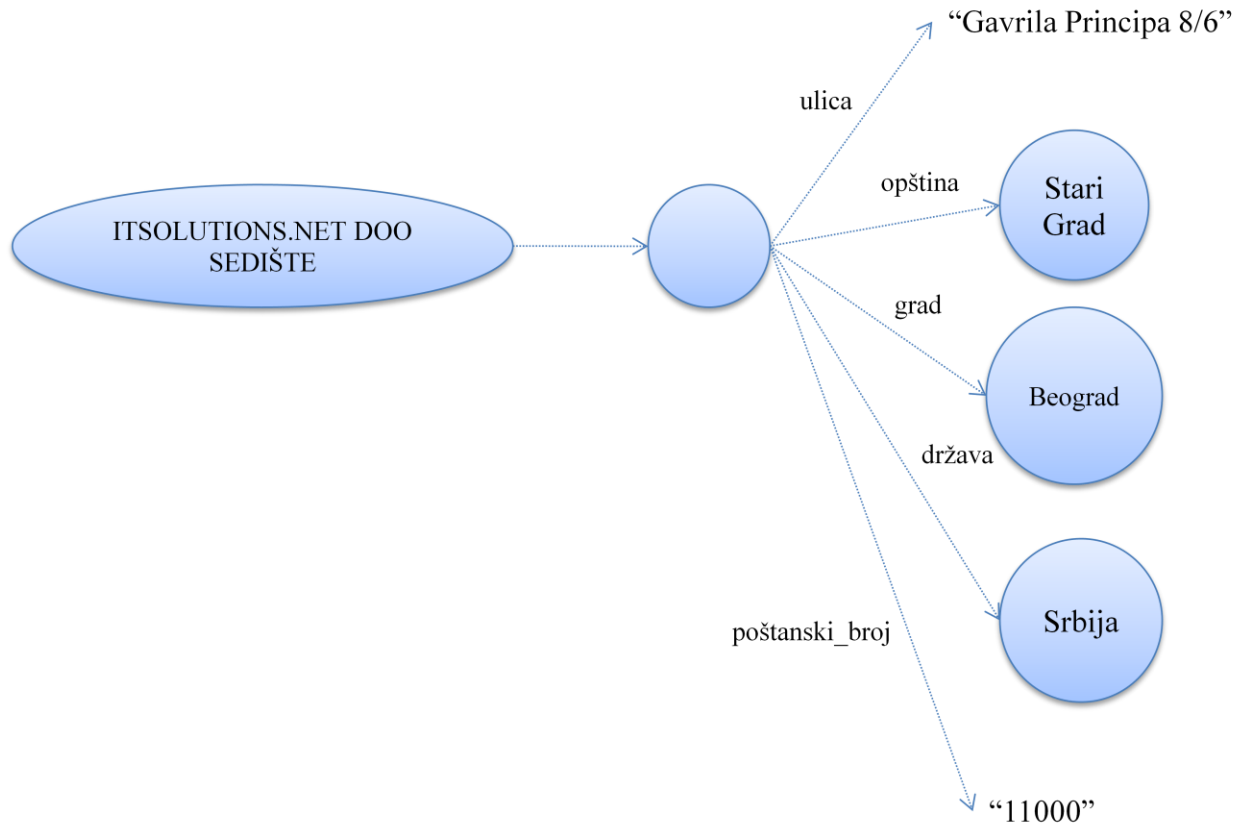
U reprezentaciji tripla, ID praznog čvora se ispisuje `_:id`, gde je id proizvoljan, graf-specifičan lokalni identifikator. Većina RDF API-a rešava ovo izdavanjem internih ID-a za čvor koji je jedino validan u lokalnom grafu i ne može da se koristi kao jaki ključ između grafova. Ovaj graf može da se napiše u triple izraz formatu:

```
(_:is1, "email", "i.stankovic@its.rs")
(_:is1, "prezime", "Stanković")
(_:is1, "ime", "Ivan")
(_:is1, "poznaje", _:az1)
(_:az1, "prezime", "Zakić")
(_:az1, "ime", "Aleksandar")
(_:az1, "email", "a.zakic@its.rs")
```

Bez obzira što ni jedan čvor nema jak spoljni identifikator, iz konteksta podataka povezanih sa čvorom jasno se vidi da Ivan Stanković sa email adresom i.stankovic@its.rs poznaje Aleksandra Zakića sa email adresom a.zakic@its.rs.

Postoje situacije kada je korisno da se kreira model podataka koji koristi čvor u RDF grafu da jednostavno grupiše skup iskaza. Na primer, kada se opisuje zgrada, često je korisno da

se grupišu naziv ulice, opština, grad, poštanski broj i država kao celina koja predstavlja poštansku adresu zgrade. Da bi se ovo uradilo potreban je čvor u grafu koji predstavlja apstraktni koncept adrese zgrade. Iako možemo da obezbedimo URIref za ovu poštansku adresu, skoro nikada neće postojati potreba koja se odnosi na ovo grupisanje čvorova van konteksta trenutnog grafa.



Sl. 35. Korišćenje praznog čvora za modelovanje poštanske adrese

Literalne vrednosti

RDF koristi literalne vrednosti na isti način kao graf sa slike 35 da izrazi imena, datume i druge tipove vrednosti o subjektu. U RDF-u, literalne vrednosti mogu opciono da imaju jezik (primer: Engleski, Japanski) ili tip (primer: integer, boolean, string) povezan sa njim. Tip URI iz XML specifikacije šeme se često koristi da označi literalne tipove. Na primer, <http://www.w3.org/2001/XMLSchema#int> ili xsd:int za integer vrednosti. ISO 639 kodovi se koriste za određivanje jezika. Na primer, en za Engleski ili ja za Japanski.

7.2.3 FORMATI RDF SERIJALIZACIJE

Iako je model podataka koji koristi RDF veoma jednostavan, serijalizovana reprezentacija je komplikovanija kada se RDF graf snimi u fajl i pošalje preko mreže iz razloga što se koriste različite metode da se sačuva kompaktnost podatka kako bi i dalje bili čitljivi. Ovi mehanizmi generalno uzimaju oblik prečice koje identifikuju više referenci na čvor u grafu koristeći zajedničku ali složenu strukturu.

Briga o kompleksnosti serijalizovanih formata nije potrebna s obzirom da postoje RDF biblioteke otvorenog koda za skoro svaki moderan programski jezik.

Četiri najpopularnija formata za serijalizaciju su:

- N-Triples, najjednostavnija notacija;
- N3, kompakcija N-Triple formata;
- RDF/XML, jedan od najzastupljenijih formata;
- “RDF in attributes” (RDFa), koji može da se ugradi u druge formate serijalizacije kao što je XHTML.

Graf prijatelja

Najjednostavniji način da se uporede različiti formati je da se napravi jednostavan graf koji može da se iskoristi kroz primere različitih serijalizacija.

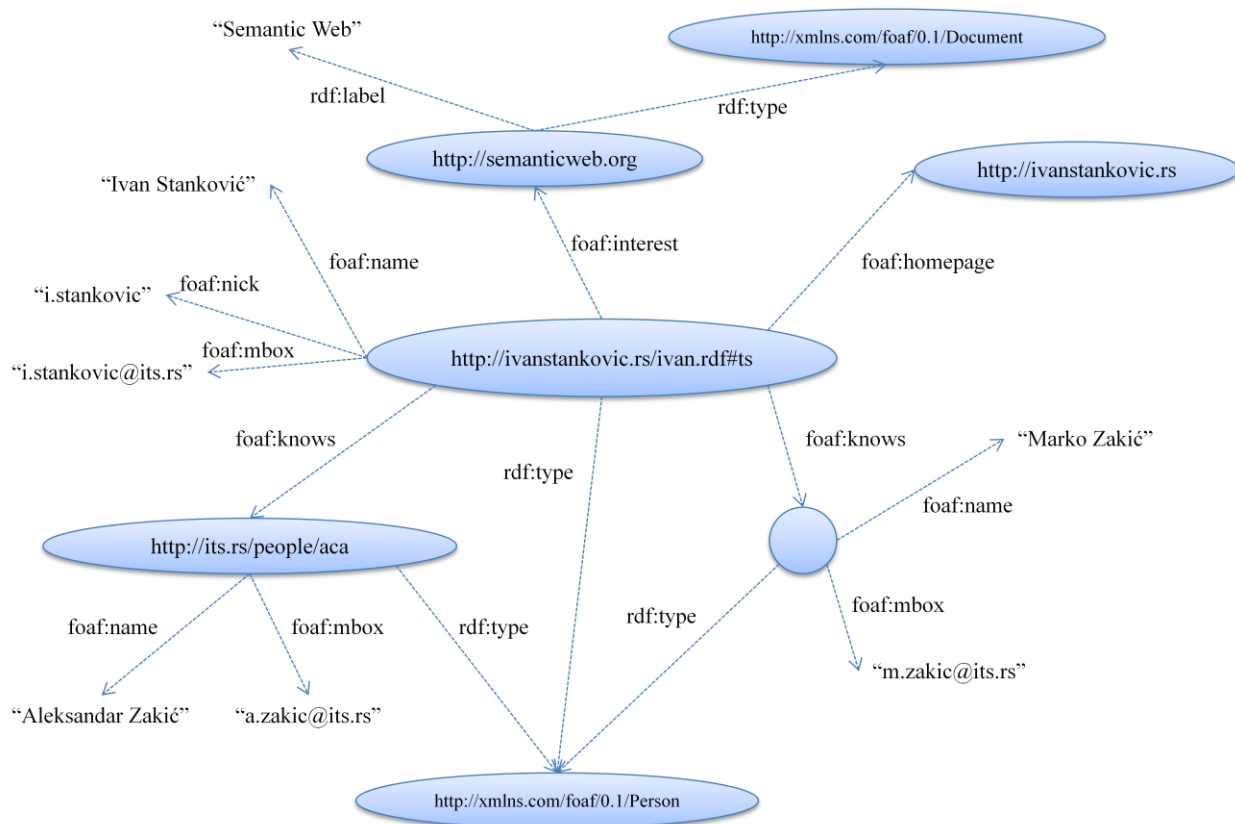
U ovom jednostavnom grafu biće modelovan mali deo Ivanove društvene sfere, a posebno kako on poznaje svoje kolege. U grafu neće biti samo informacija o ljudima koje Ivan poznaje, već i opisi drugih relacija koje mogu da se iskoriste za jedinstvenu identifikaciju Ivana. Ovo će da uključi naslovnu stranicu njegovog sajta, email adresu, interesovanja i druga imena koja bi mogao da koristi.

Ovi tragovi o identitetu su važni jer će pomoći u razlikovanju “datog Ivana” od brojnih drugih Ivana u svetu. Ljudska imena nisu ni malo jedinstvena, ali pružanjem kolekcije atributa o nekoj osobi moguće je određivanje interesovanja te osobe i obezbeđivanje jakog identifikatora (URI) koji može da se koristi u budućim interakcijama.

Kao što se nazire, mreža društvenih odnosa koje ljudi imaju međusobno jednostavno može da se prikaže kroz graf. Tako da nije iznenađenje što su grafovi prijatelja čitljivi od strane mašina koevoluirali sa RDF-om, kreirajući društvene grafove kao jedan od najdostupnijih RDF skupova podataka dostupnih na Internetu. Vremenom su odnosi izraženi u ovim društvenim grafovima postali uređena kolekcija dobro poznatih predikata, formirajući rečnik izražavanja poznat kao “Friend of a Friend” ili jednostavnije FOAF.

Jezgro FOAF rečnika, skup predikata, je usvojen i proširen za opisivanje niza uobičajenih “stvari” dostupnih na Internetu. Na primer, FOAF omogućava predikat za identifikaciju fotografija koje oslikavaju subjekat iskaza. Dok se FOAF primarno bavi ljudima, formalna definicija FOAF navodi da foaf:depiction predikat može da se koristi za grafičko oslikavanje bilo kog resursa (“stvari”) na svetu.

Slika 36 predstavlja mali deo Ivanovog društvenog sveta. Uz ovaj graf, biće prikazano kako se ovo znanje može prikazati koristeći različite oznake.



Sl. 36. Ivanov FOAF graf

N-Triples

N-Triple notacija je veoma jednostavna serijalizacija. W3C Core Working Group je koristila N-Triples da nedvosmisleno prikaže različite RDF test slučajeve modela podataka pri razvoju i ažuriranju RDF specifikacije. Ova jednostavnost čini N-Triple format korisnim za ručnu izradu skupova podataka za testiranje i debugovanje aplikacija.

Svaka izlazna vrednost u N-Triple formatu predstavlja jedan iskaz koji sadrži subjekat, predikat i objekat iza čega sledi tačka. Osim za prazne čvorove i literale, subjekat, predikat i objekat su izraženi kao apsolutni URI zatvoren u uglaste zagrade. Subjekti i objekti predstavljeni kao anonimni čvorovi su predstavljeni kao `_:name`, gde je name alfanumerički naziv čvora koji počinje slovom. Literalni objekti su stringovi pod navodnicima. Oni koriste obrnutu kosu crtu kako bi uključili navodnike, tabove, nove redove i samu obrnutu kosu crtu. Literalni string u N-Triple notaciji može opciono da precizira svoj jezik kada je praćen sa `@lang`, gde je lang ISO 639 jezički kod. Literalni mogu takođe da pruže informaciju o svom tipu podataka kada su praćeni sa `^^type`, gde je type obično XSD (XML Schema Definition) tip podataka.

Ekstenzija `.nt` se tipično koristi kada su N-Triples sačuvani u fajlu, i kada se prenose preko HTTP-a sa mime tipom `text/plain`. Oficijalni N-Triple format je dokumentovan na <http://www.w3.org/TR/rdf-testcases/#ntriples>.

Ivanov FOAF graf može da se predstavi u N-Triple formatu kao:

```

<http://ivanstankovic.rs/ivan.rdf#ts> <http://xmlns.com/foaf/0.1/homepage>
<http://ivanstankovic.rs/>.
<http://ivanstankovic.rs/ivan.rdf#ts> <http://xmlns.com/foaf/0.1/nick> "i.stankovic".
<http://ivanstankovic.rs/ivan.rdf#ts> <http://xmlns.com/foaf/0.1/name> "Ivan Stanković".
<http://ivanstankovic.rs/ivan.rdf#ts> <http://xmlns.com/foaf/0.1/mbox>
<mailto:i.stankovic@its.rs>.
<http://ivanstankovic.rs/ivan.rdf#ts> <http://xmlns.com/foaf/0.1/interest>
<http://semanticweb.org>.
<http://ivanstankovic.rs/ivan.rdf#ts> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://xmlns.com/foaf/0.1/Person>.
<http://ivanstankovic.rs/ivan.rdf#ts> <http://xmlns.com/foaf/0.1/knows> _:marko .
<http://ivanstankovic.rs/ivan.rdf#ts> <http://xmlns.com/foaf/0.1/knows>
<http://its.rs/people/aca>.
_:marko <http://xmlns.com/foaf/0.1/name> "Marko Zakić".
_:marko <http://xmlns.com/foaf/0.1/mbox> <mailto:m.zakic@its.rs>.
_:marko <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://xmlns.com/foaf/0.1/Person>.
<http://its.rs/people/aca> <http://xmlns.com/foaf/0.1/name> "Aleksandar Zakić".
<http://its.rs/people/aca> <http://xmlns.com/foaf/0.1/mbox>
<mailto:a.zakic@its.rs>.
<http://its.rs/people/aca> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://xmlns.com/foaf/0.1/Person>.
<http://semanticweb.org> <http://www.w3.org/2000/01/rdf-schema#label>
"Semantic Web".
<http://semanticweb.org> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://xmlns.com/foaf/0.1/Document>.

```

N3

Dok je N-Triple konceptualno veoma jednostavan, u izlaznoj vrednosti postoji mnogo ponavljanja. Potrebno je dodatno vreme za prenos i analizu suvišnih informacija. To nije problem dok god se radi sa malim količinama podataka, ali dodatne informacije su obavezne kada sa radi sa većom količinom podataka. Dodavanjem nekoliko dodatnih struktura, N3 kondenzuje mnogo ponavljanja u N-Triple formatu.

U RDF grafu, svaka veza između čvorova predstavlja triple. Pošto svaki čvor može da učestvuje u velikom broju veza, moguće je značajno smanjenje broja karaktera koji se koriste u N-Triple ako se koristi simbol za skraćivanje koji predstavlja čvor koji se ponavlja. Moguće je ići i dalje ako se prepozna da mnogi URI-i koji se koriste u specifičnom modelu često dolaze iz srodnih URI-a. Na isti način na koji XML obezbeđuje mehanizam imenskog prostora za generisanje kratkih imena (Qualified Name - qnames) za čvorove, N3 dozvoljava da se definiše URI prefix i identifikuje entitet URI-a u odnosu na skup prefiksa deklariranih na početku dokumenta. Iskaz:

```
@prefix itsp: <http://its.rs/people/>
```

dozvoljava da se skрати apsolutni URI za Acu iz <http://its.rs/people/aca> u itsp:aca.

Pošto je svaki čvor u RDF grafu potencijalni subjekat, isti subjekat može da se ponavlja više puta u N-Triple izlaznoj vrednosti. N3 smanjuje ova ponavljanja dozvoljavajući kombinaciju više iskaza o istom subjektu koristeći tačku i zarez (;) posle prvog iskaza tako da se u drugim iskazima navodi samo predikat i objekat za isti subjekat. Sledeći iskaz govori da Aca poznaje

Ivana, i da je Acina email adresa a.zakic@its.rs (itsp:aca subjekat je naveden samo jednom):

```
itsp:aca foaf:knows <http://ivanstankovic.rs/ivan.rdf#ts>;  
foaf:mbox "a.zakic@its.rs".
```

N3 takođe pruža „prečicu“ koja dozvoljava predstavljanje grupe iskaza koje imaju zajednički anonimni subjekat (prazan čvor) bez potrebe da se odredi interni naziv za prazan čvor. Kao što je diskutovano ranije, poštanske adrese se često modeluju kao prazan čvor za držanje svih komponenti adrese zajedno. W3C je definisao rečnik za predstavljanje elemenata podataka u vCard interchange formatu koji uključuje predikate za modelovanje adresa ulice. Na primer, adresa ITSOLUTIONS.NET DOO može da se napiše kao:

```
[ <http://www.w3.org/2006/vcard/ns#street-address> "Gavrila Principa 8/6";  
<http://www.w3.org/2006/vcard/ns#locality> "Beograd";  
<http://www.w3.org/2006/vcard/ns#postal-code> "11000";  
<http://www.w3.org/2006/vcard/ns#country-name> "Srbija"  
].
```

Važno je da se eksplicitno navede da je entitet određenog tipa. N3 omogućava da se koriste slova kao predikat za predstavljanje RDF tipova veza prikazanih kao URI <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>

Još jedan predikat za koji N3 omogućava prečicu je <http://www.w3.org/2002/07/owl#sameAs>. OWL (Web Ontology Language) je rečnik za definisanje preciznih odnosa između elemenata modela. Čak i kada modeli ne koriste preciznost OWL-a, često se sreće owl:sameAs predikat da izrazi da se dva URI-a odnose na isti entitet. Predikat sameAs se često koristi kada autori N3 određuju simbol = kao skraćeni izraz.

Kako je N-Triple podskup N3, bilo koja biblioteka sposobna za čitanje N3 može takođe da pročita N-Triple. Ivanov FOAF graf će N3 da pročita na sledeći način:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.  
@prefix itsp: <http://its.rs/people/>.  
@prefix ivan: <http://ivanstankovic.rs/ivan.rdf#>.
```

```
ivan:ts a foaf:Person;  
foaf:homepage <http://ivanstankovic.rs/>;  
foaf:interest <http://semanticweb.org>;  
foaf:knows itsp:aca,  
[ a foaf:Person;  
  foaf:mbox <mailto:m.zakic@its.rs>;  
  foaf:name "Marko Zakić";  
foaf:mbox <mailto:i.stankovic@its.rs>;  
foaf:name "Ivan Stanković";  
foaf:nick "i.stankovic".
```

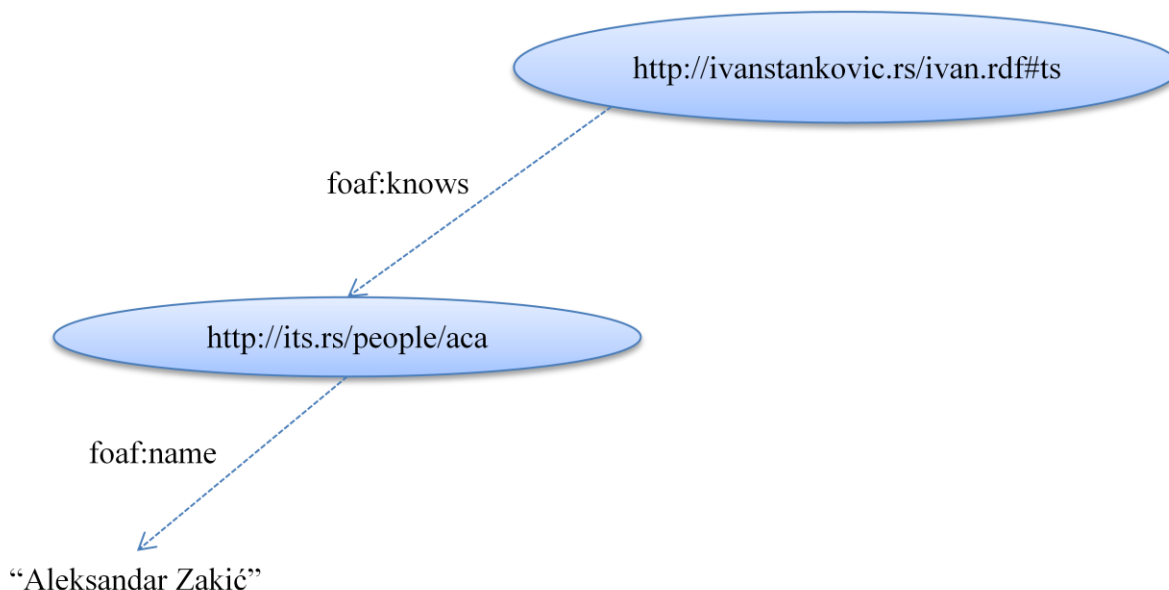
```
<http://semanticweb.org> a foaf:Document;  
rdfs:label "Semantic Web".
```

```
itsp:aca a foaf:Person;  
foaf:mbox <mailto:a.zakic@its.rs>;  
foaf:name "Aleksandar Zakić".
```

RDF/XML

RDF/XML je originalna W3C preporuka za RDF koja pokriva i opis RDF-a kao modela podataka i XML-a kao izraz RDF modela. Zbog ovoga korisnici se često prema RDF/XML-u odnose kao prema RDF-u, ali je veoma važno da se prepozna da je ovo samo jedna od mogućnosti predstavljanja RDF grafa. RDF/XML se često kritikuje kao težak za čitanje zbog svih skraćenih struktura koje pruža, ali i dalje je jedan od najčešće korišćenih formata.

Konceptualno, RDF/XML je izgrađen od niza manjih opisa od kojih svaki prati putanju kroz RDF graf. Ove putanje su opisane koristeći čvorove (subjekte) i veze (predikate) koji ih povezuju sa drugim čvorovima (objektima). Ovaj niz čvorova “čvor, veza, čvor” se ponavlja formirajući “trakastu” strukturu. Kako svaki čvor ima jaki identifikator, moguće je spojiti manje opise zajedno kako bi se “naučila” struktura većeg RDF grafa.



Sl. 37. Traka Ivanovog FOAF grafa

Ako postoji više od jedne putanje opisane u RDF/XML dokumentu, svi opisi moraju da budu “deca” jednog RDF elementa. Ako postoji opisana samo jedna putanja, rdf:RDF element može da se izostavi. Kao i kod drugih XML dokumenata, element najvišeg nivoa se najčešće koristi za definisanje drugih XML prostora imena koji se koriste kroz dokument:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
```

Putanje su uvek opisane početnim čvorom grafa koristeći rdf:Description element. URI referenca čvora može da se navede u opisu elementa sa rdf:about atributom. Za prazne čvorove, lokalni identifikator (validan samo u kontekstu tekućeg dokumenta) može da se navede koristeći rdf:NodeID atribut. Veze predikata se navode kao “deca” elementi rdf:Description čvora koji će imati svoju “decu” predstavljenu u grafu čvorova. Jednostavna traka koja predstavlja Acu kao Ivanovog prijatelja bi izgledala ovako:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">

<rdf:Description rdf:About="http://ivanstankovic.rs/ivan.rdf#ts">
<foaf:knows>
<rdf:Description rdf:About="http://its.rs/people/aca">
<foaf:name>Aleksandar Zakić</foaf:name>
</rdf:Description>
</foaf:knows>
</rdf:Description>

</rdf:RDF>
```

Literalni objekti mogu da se navedu kao tekst elementa ili kao atribut `rdf:Description` elementa. Ako bi se proširio primer na dodavanje novih informacija o drugom Ivanovom prijatelju, onda bi to izgledalo ovako:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">

<rdf:Description rdf:About="http://ivanstankovic.rs/ivan.rdf#ts">
<foaf:knows>
<rdf:Description rdf:About="http://its.rs/people/aca">
<foaf:name>Aleksandar Zakić</foaf:name>
</rdf:Description>
</foaf:knows>

<foaf:knows>
<rdf:Description foaf:mbox="m.zakic@its.rs"/>
</foaf:knows>
</rdf:Description>

</rdf:RDF>
```

Iako je ovo jasno definisan opis odnosa Ivana, Ace i Marka, još uvek nedostaje `rdf:type` informacija koja navodi da su Ivan, Aca i Marko ljudi. Kao i u drugim RDF serijalizacijama, RDF/XML obezbeđuje prečicu za ovaj iskaz, što omogućava da se `rdf:Description` element zameni elementom koji predstavlja `rdf:type` čvora. Tako je redosled elemenata:

```
<rdf:Description rdf:about="http://ivanstankovic.rs/ivan.rdf#ts">
<rdf:type>
<foaf:Person>
```

prikazano u `rdf:Description` elementu u formi:

```
<foaf:Person rdf:about="http://ivanstankovic.rs/ivan.rdf#ts">
```

FOAF graf predstavljen u N-Triple i N3 sada može da se predstavi u RDF/XML-u kao:

```
<rdf:RDF
  xmlns:foaf='http://xmlns.com/foaf/0.1/'
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:rdfs='http://www.w3.org/2000/01/rdf-schema#'
```

```

<foaf:Person rdf:about="http://ivanstankovic.rs/ivan.rdf#ts">
<foaf:name>Ivan Stanković</foaf:name>
<foaf:homepage rdf:resource="http://ivanstankovic.rs/" />
<foaf:nick>i.stankovic</foaf:nick>
<foaf:mbox rdf:resource="mailto:i.stankovic@its.rs" />
<foaf:interest>
<foaf:Document rdf:about="http://semanticweb.org">
<rdfs:label>Semantic Web</rdfs:label>
</foaf:Document>
</foaf:interest>
<foaf:knows>
<foaf:Person rdf:about="http://its.rs/people/aca">
<foaf:name>Aleksandar Zakić</foaf:name>
<foaf:mbox rdf:resource="mailto:a.zakic@its.rs" />
</foaf:Person>
</foaf:knows>

<foaf:knows>
<foaf:Person>
<foaf:name>Marko Zakić</foaf:name>
<foaf:mbox rdf:resource="mailto:m.zakic@its.rs" />
</foaf:Person>
</foaf:knows>
</foaf:Person>
</rdf:RDF>

```

Ovo nisu jedine skraćene RDF/XML strukture, ali bi trebalo da budu dovoljne za čitanje većine RDF/XML fajlova.

RDFa

RDFa nije čista serijalizacija RDF formata, već je način obeležavanja XHTML web stranica sa RDF podacima. Ideja korišćenja RDFa je da se sadržaj objavi samo jednom mešanjem čitljivog sadržaja od strane korisnika i mašina zajedno. Ovo je slično filozofiji Microformata, jednostavniji, više ad-hok pristup dodavanja semantičkih anotacija u XHTML sadržaj.

RDFa koristi mali skup XML atributa koji su dodati postojećim XHTML tagovima sadržaja kako bi se precizirala semantika koja opisuje prikazane informacije. Ovi atributi semantičko značenje postojećih XHTML sadržaja čine jasnijim. Osnovni model procesiranja je takav da je subjekat tripla, u stvari subjekat URI-a identifikovan na većem nivou XHTML elementa u DOM strukturi stabla, a predikat i objekat iskazi su “deca” tog subjekta.

Umesto korišćenja URI-a za opis subjekta, predikta i objekta, mnogi RDFa atributi koriste Compact URI (CURIEs) kako bi smanjili broj oznaka. CURIEs funkcionišu kao XML Qualified Names (u stvari QNames su podskup CURIEs), tako da sve što važi za XML QNames (na primer, foaf:nick u stvari znači <http://xmlns.com/foaf/0.1/nick>) važi i za CURIEs. Ali CURIEs su pogodniji za sadržaj localpart od prefix:localpart izraza.

Izgradnja QName zabranjuje kose crte (/) u localpart-u, tako da zahteva odvojenu deklaraciju XML imenskog prostora za svaki QName koristeći različite delove hijerarhijske putanje. CURIEs dozvoljava iskazima kao što su example:cow i example:places/barn da koriste xmlns deklaraciju kao što je <http://example.org/farm> za generisanje celog URI-a

<http://example.org/farm/cow> i <http://example.org/farm/places/barn> respektivno.

CURIEs takođe dozvoljavaju da localpart počne brojem. Ovo znači da je moguće definisanje XML imenskog prostora kao:

```
xmlns:amazonisbn="http://www.amazon.com/exec/obidos/ASIN/"
```

i pozivanje na knjigu The Art of War po ISBN-u sa CURIE-m:

```
amazonisbn:9780385292160
```

CURIEs su pogodni sa prediktima jer mogu da naprave jednu xmlns deklaraciju za svaki rečnik koji se koristi i da brzo izgrade CURIEs za bilo koje svojstvo u rečniku. Međutim, nisu pogodni kada je potrebno koristiti veliki niz subjekata i objekata (s obzirom da je potrebno da se napravi xmlns deklaracija za svaki jedinstven bazni URI). Da bi razrešio ovaj problem, RDFa dozvoljava korišćenje celih URI-a za nekoliko atributa subjekta i objekta. S obzirom da ceo URI koristi dve tačke (:) da razdvoji protokol šeme od hijerarhijskog dela URI-a, parseri mogu pogrešno da procene šta je http:, da li je to http: kao u <http://example.org/cow> ili je u pitanju CURIE gde je http prefix nekog imenskog prostora.

Da bi se izbegla ova zbrka, RDFa definiše “safe CURIE” koji čini da CURIE bude jasan kada se koriste iskazi odvajanja sa dve tačke u odnosu na identifikator protokola u URI-u. Za izgradnju sigurnog CURIE potrebno je da CURIE bude u uglastim zagradama kao u:

```
[example:place/barn]
```

Atribut za definisanje RDF subjekta je:

- **about** – URI (ili sigurni CURIE) koristi subjekat u RDF triple-u. Po podrazumevanoj vrednosti, osnova URI-a za stranicu je bazni URI za sve iskaze. Atribut **about** dozvoljava da se iskazi koriste gde osnovni URI nije subjekat.

Atributi za definisanje RDF predikta su:

- **rel** – CURIE koji izražava odnose između dva resursa;
- **property** – CURIE koji izražava odnos između resursa i literala;
- **rev** – CURIE koji izražava obrnuti odnos između dva resursa;

Atributi za definisanje RDF objekta su:

- **content** – string koji predstavlja literal RDF objekta;
- **href** – URI resurs koji izražava RDF objekat kao inline klik;
- **src** – URI resurs koji izražava RDF objekat kao inline ugrađenu stavku;
- **resource** – URI (ili sigurni CURIE) koji izražava RDF objekat kada objekat nije vidljiv na stranici.

RDF takođe omogućava i specijalne attribute za određivanje tipova podataka i kreiranje rdf:type izjava:

- **datatype** – tip podataka literal;
- **typeof** – tip subjekta.

Važno je napomenuti da atributi koji se koriste za postavljanje predikata zavise od tipa objekta u RDF iskazu. Ako je objekat literal, onda je predikat određen property atributom. Ako je objekat resurs, onda se koriste rel i rev atributi. Iz razloga što je XHTML jezik za označavanje za prikaz koji je čitljiv za korisnika, nije pogodan za prikaz podataka u subjekat-predikat-objekat redosledu RDF iskaza. Da bi se rešila ova situacija RDFa obezbeđuje rev atribut za definisanje predikta, koji takođe ukazuje da je redosled iskaza obrnut (objekat-predikat-subjekat).

Minimalno je moguće da se navede RDF triple u jednom elementu za označavanje. U sledećem primeru, objekat je literal, tako da koristi atribut property da bi naveo predikat:

```
<span xmlns:foaf="http://xmlns.com/foaf/0.1/"
about="http://ivanstankovic.rs/ivan.rdf#ts"
property="foaf:nick"
content="i.stankovic" />
```

Kada je objekat iskaza resurs, koristi se rel atribut da bi se naveo predikat:

```
<span xmlns:foaf="http://xmlns.com/foaf/0.1/"
about="http://ivanstankovic.rs/ivan.rdf#ts"
rel="http://xmlns.com/foaf/0.1/homepage"
href="http://ivanstankovic.rs" />
```

Takođe može da se koristi XHTML element za prikaz delova strukture. Na primer, moguće je napraviti sledeći iskaz u vezi Ivanovog nadimka prilikom prikaza stringa "i.stankovic":

```
Ivanov nadimak je: <span xmlns:foaf="http://xmlns.com/foaf/0.1/"
about="http://ivanstankovic.rs/ivan.rdf#ts"
property="http://xmlns.com/foaf/0.1/nick">i.stankovic</span>
```

FOAF graf sad može da se predstavi kao fragment XHTML-a anotiran sa RDFa-om. XML atributi i tekst koji RDFa parseri sakupe iz XHTML su boldovani. Anotacija može da se napravi u bilo kom tagu i moguće je korišćenje postojećih XHTML atributa i teksta koji se takođe koriste za XHTML sadržaj čitljiv za korisnike. Svi podaci su dostupni na jednom mestu i koncept je nedvosmisleno opisan koristeći jak URIrefs:

```
<div xmlns:foaf="http://xmlns.com/foaf/0.1/"
about="http://ivanstankovic.rs/ivan.rdf#ts" typeof="foaf:Person">
<p>
Ime: <span property="foaf:name">Ivan Stanković</span><br/>
Nadimak: <span property="foaf:nick">i.stankovic</span><br/>
Interesovanje: <a rel="foaf:interest" href="http://semanticweb.org">
<span property="rdfs:label">Semantic Web</span></a>
Homepage: <a rel="foaf:homepage"
href="http://ivanstankovic.rs/">IvanStanković</a>
</p>

Priatelji:<br/>
<ul rel="foaf:knows">
<li about="http://its.rs/people/aca"
typeof="foaf:Person" property="foaf:name">Aleksandar Zakić</li>
<li typeof="foaf:Person">
<span property="foaf:name">Marko Zakić</span><br/>
```

```
Email: <a rel="foaf:mbox" href="mailto:m.zakic@its.rs">
m.zakic@its.rs</a><br/>
</li>
</ul>
</div>
```

XHTML je dizajniran da bude proširiv i da omogući dodavanje novih atributa u elemente za označavanje, RDFa se koristi za anotaciju u XHTML-u. U praksi RDFa dobro funkcioniše sa HTML-om iako oznake nisu validne u HTML 4.

Količina RDFa dostupnog na webu ubrzano raste. Broj značajnih sajtova koji koriste mogućnosti RDFa izlazne vrednosti raste (uključujući MySpace), iako se to ne vidi jer RDFa ne menja HTML renderovanje.

RDFa je “preporuka” W3C-a. Postoji još sličnih rešenja ugradnje RDF-a u HTML. Jedan pokušaj je eRDF (“embeddable RDF”) odakle je nastao RDFa. eRDF nije primenjiv jer nije u potpunosti podržan od RDF alata.

7.3 SPARQL

Kao što SQL obezbeđuje relativno standardan jezik upita u sistemima relacionih baza podataka, SPARQL omogućava standardizovan jezik upita za RDF grafove. SPARQL (Simple Protocol and RDF Query Language) ima mogućnost filtriranja rezultata i izgradnje novih grafova na osnovu upita. SPARQL upiti pokušavaju da uporede uzorke u grafu i povežu „wildcard” promenljive kako on pronađe rešenja [36].

U ovom delu rada se vrše upiti na malom grafu podataka o filmovima, predstavljениh u N3 formatu. Skup iskaza je deo većeg skupa tripleta koji predstavljaju filmove snimljene između 2000. i 2008. godine. Izvor ovih podataka je Freebase, velika otvorena semantička baza podataka besplatna za korišćenje.

```
@prefix fb: <http://rdf.freebase.com/ns/> .

<http://rdf.freebase.com/ns/en.hollywood_homicide>
  <http://rdf.freebase.com/ns/film.film.directed_by>
    <http://rdf.freebase.com/ns/en.ron_shelton> ;
<http://rdf.freebase.com/ns/film.film.starring>
  <http://rdf.freebase.com/ns/en.harrison_ford> ,
  <http://rdf.freebase.com/ns/en.kurupt> ,
  <http://rdf.freebase.com/ns/en.robert_wagner> ;
<http://rdf.freebase.com/ns/film.film.initial_release_date> "2003" .

<http://rdf.freebase.com/ns/en.k_19_the_widowmaker>
  <http://rdf.freebase.com/ns/film.film.directed_by>
    <http://rdf.freebase.com/ns/en.kathryn_bigelow> ;
<http://rdf.freebase.com/ns/film.film.starring>
  <http://rdf.freebase.com/ns/en.harrison_ford> ,
  <http://rdf.freebase.com/ns/en.joss_ackland> ;
<http://rdf.freebase.com/ns/film.film.initial_release_date> "2002" .

<http://rdf.freebase.com/ns/en.dark_blue>
  <http://rdf.freebase.com/ns/film.film.directed_by>
```

```

    <http://rdf.freebase.com/ns/en.ron_shelton> ;
<http://rdf.freebase.com/ns/film.film.starring>
    <http://rdf.freebase.com/ns/en.kurupt> ,
    <http://rdf.freebase.com/ns/en.kurt_russell> ,
    <http://rdf.freebase.com/ns/en.dash_mihok> .

<http://rdf.freebase.com/ns/en.the_weight_of_water_2002>
    <http://rdf.freebase.com/ns/film.film.directed_by>
    <http://rdf.freebase.com/ns/en.kathryn_bigelow> ;
<http://rdf.freebase.com/ns/film.film.starring>
    <http://rdf.freebase.com/ns/en.sean_penn> ,
    <http://rdf.freebase.com/ns/en.elizabeth_hurley> ;
<http://rdf.freebase.com/ns/film.film.initial_release_date> "2002" .

<http://rdf.freebase.com/ns/en.becoming_dick>
    <http://rdf.freebase.com/ns/film.film.directed_by>
    <http://rdf.freebase.com/ns/en.bob_saget> ;
<http://rdf.freebase.com/ns/film.film.starring>
    <http://rdf.freebase.com/ns/en.robert_wagner> ,
    <http://rdf.freebase.com/ns/en.bob_saget> ;
<http://rdf.freebase.com/ns/film.film.initial_release_date> "2000" .

<http://rdf.freebase.com/ns/en.body_of_lies>
    <http://rdf.freebase.com/ns/film.film.directed_by>
    <http://rdf.freebase.com/ns/en.ridley_scott> ;
<http://rdf.freebase.com/ns/film.film.starring>
    <http://rdf.freebase.com/ns/en.russell_crowe> ,
    <http://rdf.freebase.com/ns/en.mark_strong> ;
<http://rdf.freebase.com/ns/film.film.initial_release_date> "2008" .

<http://rdf.freebase.com/ns/en.kurt_russell>
    <http://rdf.freebase.com/ns/type.object.name> "Kurt Russell" .
<http://rdf.freebase.com/ns/en.dash_mihok>
    <http://rdf.freebase.com/ns/type.object.name> "Dash Mihok" .
<http://rdf.freebase.com/ns/en.sean_penn>
    <http://rdf.freebase.com/ns/type.object.name> "Sean Penn" .
<http://rdf.freebase.com/ns/en.elizabeth_hurley>
    <http://rdf.freebase.com/ns/type.object.name> "Elizabeth Hurley" .
<http://rdf.freebase.com/ns/en.kathryn_bigelow>
    <http://rdf.freebase.com/ns/type.object.name> "Kathryn Bigelow" .
<http://rdf.freebase.com/ns/en.bob_saget>
    <http://rdf.freebase.com/ns/type.object.name> "Bob Saget" .
<http://rdf.freebase.com/ns/en.ridley_scott>
    <http://rdf.freebase.com/ns/type.object.name> "Ridley Scott" .
<http://rdf.freebase.com/ns/en.russell_crowe>
    <http://rdf.freebase.com/ns/type.object.name> "Russell Crowe" .
<http://rdf.freebase.com/ns/en.mark_strong>
    <http://rdf.freebase.com/ns/type.object.name> "Mark Strong" .
<http://rdf.freebase.com/ns/en.ron_shelton>
    <http://rdf.freebase.com/ns/type.object.name> "Ron Shelton" .
<http://rdf.freebase.com/ns/en.harrison_ford>

```

```

<http://rdf.freebase.com/ns/type.object.name> "Harrison Ford" .
<http://rdf.freebase.com/ns/en.robert_wagner>
  <http://rdf.freebase.com/ns/type.object.name> "Robert Wagner" .
<http://rdf.freebase.com/ns/en.kurupt>
  <http://rdf.freebase.com/ns/type.object.name> "Kurupt" .

```

SPARQL omogućava četiri forme upita: SELECT, CONSTRUCT, ASK i DESCRIBE. Svi upiti pokušavaju da nađu rešenje za uzorak grafa i dele sličnu konstrukciju. Iako je SPARQL preporuka W3C, mnoge semantičke platforme podržavaju samo nekoliko SPARQL upit formata. Međutim svaka “ozbiljna” semantička platforma podržava SELECT formu SPARQL upita.

7.3.1 SELECT FORMA UPITA

Kao i kod svih SPARQL upita, SELECT upit može da počne blokom deklaracije PREFIX koja daje stenografske identifikatore za URI imenski prostor koji može da se koristi kroz upit. Da bi se skratio Freebase URI potrebno je da se doda sledeći iskaz:

```
PREFIX fb: <http://rdf.freebase.com/ns/>
```

U početnom delu upita je moguća definicija BASE URI-a na koji su povezani svi relativni URI-i. BASE deklaracija omogućava korišćenje relativnog URI-a <b006ww0v.rdf> za određivanje Pete Tong BBC programa koji ima apsolutni URI:

```

<http://www.bbc.co.uk/programmes/b006ww0v.rdf>,
BASE <http://www.bbc.co.uk/programmes/>

```

Iako je moguće više PREFIX deklaracija, BASE deklaracija je moguća samo jednom.

SELECT upit dozvoljava identifikaciju podskupa varijabli korišćenih u uzorku grafa koje su povezane u vraćenim rešenjima. SELECT klauzula je praćena sa WHERE klauzulom koja određuje uzorak grafa za procesiranje kao kolekciju tripleta. Varijable u triple uzorku su identifikovane string karakteristikama koji počinju sa znakom pitanja (?) ili dolarom (\$) (nema razlike između ova dva identifikatora varijabli).

Sledeći primer prikazuje upit koji postavlja pitanje koji se režiser pojavljuje u svom filmu:

```

PREFIX fb:<http://rdf.freebase.com/ns/>

SELECT ?who ?film
WHERE{
  ?film fb:film.film.directed_by ?who .
  ?film fb:film.film.starring ?who .
}

```

Ovaj upit proizvodi sledeći rezultat:

who	Film
fb:en.bob_saget	fb:en.becoming_dick

Tabela 6. Rezultat pretrage

7.3.2 OPTIONAL I FILTER OGRANIČENJA

Ponekad je potrebno iskoristiti informacije koje su dostupne, a ponekad se ignorišu kada nisu dostupne. Ovo se često dešava kada postoje nepotpune informacije o resursu, kao što je na primer nedostatak datuma objavljivanja filma u bazi filmova. SPARQL pruža klauzulu OPTIONAL koja omogućava da se koristi informacija u uzorku grafa ako je dostupna, ali ne

eleminiše rešenje ako nedostaje. Sledeći primer vraća spisak svih filmova Rona Sheltona, uključujući i datum objavljivanja ako postoji:

```
PREFIX fb: <http://rdf.freebase.com/ns/>

SELECT ?film ?reldate
WHERE {
    ?film fb:film.film.directed_by fb:en.ron_shelton .
    OPTIONAL { ?film fb:film.film.initial_release_date ?reldate .}
}
```

Kada su informacije dostupne, klauzula OPTIONAL vezuje ?reldate za rešenje koje proizvodi zahtevani deo uzorka grafa:

film	Reldate
fb:en.dark_blue	
fb:en.hollywood_homicide	2003

Tabela 7. Rezultat pretrage

Uzorci grafa su korisni za određivanje da li određeni skup veza postoji u grafu, ali često je potrebno ograničenje rešenja baziranog na specifičnosti subjekta, predikta i objekta. SPARQL pruža klauzulu FILTER koja omogućava da se navedu dodatna ograničenja na rešenje vezivanja. FILTER ograničenje koristi mali skup operatora, od kojih su mnogi izvedeni iz XPath 2.0, koji omogućavaju testiranje promenljivih u uzorku grafa za specifične uslove. Kada FILTER vrati false (ili error - grešku), rešenje koje se razmatra uklanjanje neželjene elemente.

Na primer, dok je klauzula OPTIONAL korisna za rukovanje podacima koji nedostaju, bound FILTER operator omogućava pronalaženje subjekata koji ne potvrđuju određeni odnos. Sledeći primer koristi FILTER da pronađe filmove koje je režirao Ron Shelton koji nemaju datum objavljivanja:

```
PREFIX fb: <http://rdf.freebase.com/ns/>

SELECT ?film
WHERE {
    ?film fb:film.film.directed_by fb:en.ron_shelton .
    OPTIONAL { ?film fb:film.film.initial_release_date ?reldate .}
    FILTER (!bound(?reldate))
}
```

Rezultat govori da je skup podataka koji nemaju datum objavljivanja za film Dark Blue fb:en.dark_blue.

SPARQL filter operatori takođe dozvoljavaju postavljanje uslova za kvalitet vrednosti vezanih varijabli. Na primer, često je korisno znati da li string literal sledi specifičan uzorak. Za ovu vrstu upoređivanja stringova, SPARQL pruža regex operator koji uzima vezanu varijablu, regex uzorak, i opcioni skup oznaka koje omogućavaju ignorisanje slučajeva različitosti između uzorka i vrednosti varijable.

Sledeći primer postavlja pitanje koji glumac sadrži string “russell” (case insensitive) u svom imenu:

```
PREFIX fb:<http://rdf.freebase.com/ns/>
```

```
SELECT distinct ?who ?film
```

```
WHERE {
```

```
    ?film fb:film.film.starring ?star .
```

```
    ?star fb:type.object.name ?who .
```

```
    FILTER regex(?who, "russell", "i")
```

```
}
```

Rezultat je Russell Crowe i Kurt Russell. SPARQL regex operator je definisan da bude isti kao i u XPath 2.0 fn:matches operator koji omogućava određivanje složenih regex uzoraka. Na primer, moguće je limitiranje rezultata na Russell Crowe zahtevajući da “russell” bude na početku stringa:

```
FILTER regex(?who, "^russell", "i")
```

SPARQL pruža širok spektar operatora poređenja za filtre dozvoljavajući podelu rezultata. Na primer, uz pomoć filtera znaka nejednakosti da pronade filmove objavljene posle 2002:

```
PREFIX fb:<http://rdf.freebase.com/ns/>
```

```
SELECT ?film ?when
```

```
WHERE {
```

```
    ?film fb:film.film.initial_release_date ?when .
```

```
    FILTER (?when > "2002")
```

```
}
```

Ovaj upit vraća sledeći rezultat:

Film	When
fb:en.hollywood_homicide	2003
fb:en.body_of_lies	2008

Tabela 8. Rezultat pretrage

7.3.3 VIŠESTRUKI UZORCI GRAFA

Do sada su svi primeri bili primenjivani na jedan uzorak grafa, ali SPARQL dozvoljava da se navede više uzoraka grafova u upitu koristeći zagrade za grupisanje tripleta u odvojene uzorke. Bez modifikatora, svi uzorci se procenjuju zajedno da proizvedu rešenje (kao da su svi tripleti u istoj grupi). Kada su uzorci grupisani, klauzula FILTER se odnosi na uzorak grafa sa kojim je grupisan.

U sledećem primeru, traži se režiser sa imenom od tri slova i glumac čije ime počinje sa “B”. Pošto oba ograničenja moraju da se zadovolje, rešenje mora da bude neko ko je i glumac i režiser. U skupu podataka koji se koristi u primerima postoji samo jedan režiser koji je takođe i glumac (Bob Saget), tako da su kandidati za skup rešenja prilično ograničeni za početak. Bob Saget je režiser sa imenom od tri slova, i glumac čije ime počinje na “B”, tako da je on rešenje upita.

```
PREFIX fb:<http://rdf.freebase.com/ns/>
```

```
SELECT ?name
```

```
WHERE {
```

```
    {
```

```
        ?film fb:film.film.directed_by ?person .
```

```

        ?person fb:type.object.name ?name
        filter regex(?name, "^... ", "i")
    }
    {
        ?film fb:film.film.starring ?actor .
        ?actor fb:type.object.name ?name
        filter regex(?name, "^b", "i")
    }
}

```

Oba uzorka grafa se procenjuju zajedno i proizvode Bob Saget kao rešenje.

Korišćenjem UNION ključne (rezervisane) reči, svaki uzorak u upitu može da se procenjuje samostalno i da se njihova rešenja sjedine. Ovo može da bude korisno kada je više rešenja podjednako korisno.

```

PREFIX fb:<http://rdf.freebase.com/ns/>
SELECT ?name
WHERE {
    {
        ?film fb:film.film.directed_by ?director .
        ?director fb:type.object.name ?name
        filter regex(?name, "^... ", "i")
    }
    UNION
    {
        ?film fb:film.film.starring ?actor .
        ?actor fb:type.object.name ?name
        filter regex(?name, "^b", "i")
    }
}

```

U ovom slučaju ne postoji uslov da jedno ime ispinjava oba modela, tako da su režiser Ron Shelton i Bob Saget rešenja.

Često je korisno da se odluči da li su dva resursa ista kada se razmatra rešenje. U sledećem primeru se pretražuje režiser koji je radio sa određenim glumcem, u ovom slučaju Harrisonom Fordom. Upit onda “pita” ko je još glumio u tim filmovima. Onda nalazi i druge filmove u kojima su režiser i drugi glumci radili zajedno. U primeru treba da se isključe filmovi u kojima je Harrison Ford radio sa režiserom i drugim glumcima, potrebno je samo rešenje o drugim filmovima u kojima su režiser i drugi glumci radili zajedno. Da bi se ovo uradilo, navodi se FILTER ograničenje gde ?othermovie i ?movie ne mogu da budu isti.

```

PREFIX fb: <http://rdf.freebase.com/ns/>

SELECT ?othermovie ?director ?costar
WHERE {
    ?movie fb:film.film.starring fb:en.harrison_ford .
    ?movie fb:film.film.directed_by ?director .
    ?movie fb:film.film.starring ?costar .
    ?othermovie fb:film.film.directed_by ?director .
    ?othermovie fb:film.film.starring ?costar .
}

```

```
} FILTER (?othermovie != ?movie)
```

Ovaj upit govori da su Ron Shelton i Kurupt radili na filmu sa Harrison Fordom, i da su radili zajedno na drugom filmu.

7.3.4 CONSTRUCT FORMA UPITA

U mnogim situacijama je korisno dobiti spisak povezanih varijabli iz upita, ali postoje situacije u kojima je potrebno da se izgradi novi graf na osnovu skupa rešenja. Iako je moguće da se napiše malo koda koji konvertuje povezano rešenje u triplete i dodaje u graf, SPARQL pruža formu upita koja ovo radi direktno. Novi graf je konstruisan iz šablona tripleta zadatim u klauzuli CONSTRUCT koja menja SELECT klauzulu. Klauzule WHERE i FILTER rade na isti način kao u SELECT formi. U sledećem primeru su kreirani tripleti koji ukazuju ko je bio zaposlen koje godine:

```
PREFIX fb:<http://rdf.freebase.com/ns/>

CONSTRUCT {
  ?who <http://employment.history/was_employed_in> ?year
}
WHERE {
  {
    ?film fb:film.film.starring ?who .
    ?film fb:film.film.initial_release_date ?year .
  }
  UNION
  {
    ?film fb:film.film.directed_by ?who .
    ?film fb:film.film.initial_release_date ?year .
  }
}
```

7.3.5 ASK I DESCRIBE FORME UPITA

SPARQL pruža jednostavnu formu ASK koja testira da li se uzorak nalazi u grafu. ASK zamenjuje WHERE ključnu reč, i vraća se jednostavan boolean rezultat koji odgovara da li postoji rešenje za uzorak u grafu. Sledeći upit pita da li se Bob Saget i Harrison Ford pojavljuju u istom filmu:

```
PREFIX fb:<http://rdf.freebase.com/ns/>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>

ASK {
  ?film fb:film.film.starring fb:en.bob_saget .
  ?film fb:film.film.starring fb:en.harrison_ford .
}
```

DESCRIBE je potencijalno moćna forma upita koja vraća korisne informacije koje servis ima o resursu. Rezultati su “jaka reakcija” na implementaciju servisa upita. U teoriji izdavanje DESCRIBE upita pomaže razumevanju konteksta vraćenog resursa.

U najjednostavnijoj formi, moguće je pitati sistem upita da opiše sve što zna o specifičnom resursu:

```
DESCRIBE <http://rdf.freebase.com/ns/en.harrison_ford>
```

Korišćenje DESCRIBE je isto kao i korišćenje SELECT, ali umesto vraćenog skupa rešenja, sistem pokušava da pruži informaciju o resursima vraćenim u rešenju. U sledećem primeru upit bi trebalo da opiše režisere koji su snimili filmove u 2003:

```
PREFIX fb:<http://rdf.freebase.com/ns/>

DESCRIBE ?director
WHERE {
    ?film fb:film.film.initial_release_date "2003" .
    ?film fb:film.film.directed_by ?director .
}
```

7.3.6 KORISNI MODIFIKATORI UPITA

SPARQL je “bogat” interfejs upita koji omogućava veliki broj opcionih modifikatora koji su veoma korisni u razvoju aplikacija. Na primer, SPARQL podržava jednostavnu formu straničenja rezultata koristeći OFFSET, LIMIT i ORDER BY. Da bi straničenje radilo prvo je potrebno da se uvede red u rešenje pomoću modifikatora ORDER BY. ORDER BY uzima listu vezanih varijabli, sortira rešenje po prvoj varijabli, zatim sortira rezultat rešenja redom koristeći sledeće varijable ako su precizirane.

```
PREFIX fb:<http://rdf.freebase.com/ns/>

SELECT ?name ?year
WHERE{
    ?movie fb:film.film.initial_release_date ?year .
    ?movie fb:film.film.starring ?actor .
    ?actor fb:type.object.name ?name .
} ORDER BY ?year ?name
```

Ovaj upit vraća listu glumaca sortiranih po godini kada su radili na filmu, pa onda po imenu:

name	Year
Bob Saget	2000
Robert Wagner	2000
Elizabeth Hurley	2002
Harrison Ford	2002
Joss Ackland	2002
Sean Penn	2002
Harrison Ford	2003
Kurupt	2003
Robert Wagner	2003
Mark Strong	2008
Russell Crowe	2008

Tabela 9. Rezultat pretrage

Redosledom u rešenju, moguće je kreiranje stranica određene veličine pomoću LIMIT ključne reči. OFFSET ključna reč onda može da se iskoristi da bi označila tačku u rešenju od koje treba da počne sledeće preuzimanje. Ako broj stranice počinje od 1, da bi se dobili rezultati za stranicu N, navodi se OFFSET koji je (N-1)*LIMIT.

```
PREFIX fb:<http://rdf.freebase.com/ns/>
```

```
SELECT ?film ?year
```

```
WHERE {
```

```
    ?film fb:film.film.initial_release_date ?year.
```

```
}
```

```
ORDER BY ?year LIMIT 2 OFFSET 0
```

Ovaj upit vraća listu filmova od najstarijeg do najnovijeg, sa limitom od 2 filma, sa početkom od prvog filma.

Ovde je prikazan SPARQL 1.0, koji podržava samo operacije za čitanje grafa. Iako je ovo ograničenje, moguće je da se dozvoli nepouzdanom sistemu (sa nekim ograničenjima na pristupnim funkcijama) pristup da vrši upite bez “straha” od menjanja grafa. Omogućavanje sirovog interfejsa upita na udaljeno skladište podataka je “moćan” arhitektonski dizajn.

SPARQL 1.1 je u procesu standardizacije i sadržaće mnoge potrebne mogućnosti:

- **Aggregates:** mogućnost grupisanja rezultata pretrage i izračunavanje agregatnih vrednosti (count, min, max, avg, sum, itd.);
- **Projected expressions:** mogućnost da rezultat pretrage sadrži vrednosti izvedene iz konstanti, poziva funkcija, ili drugih izraza u SELECT listi;
- **Sub-queries:** mogućnost da upit bude ugnježđen u okviru drugog upita;
- **Negation:** sadrži dva operatora negacije: NOT EXISTS i MINUS;
- **Update:** update jezik za RDF;
- **Property paths:** upit proizvoljne dužine putanje grafa preko sintakse nalik regularnim izrazima;
- **Query Federation:** mogućnost podele jednog upita i slanje delova različitim SPARQL endpoints (krajnje tačke), a zatim kombinovanje rezultata iz svakog;
- **Service Description:** rečnik i mehanizam za otkrivanje koji opisuje mogućnosti SPARQL krajnje tačke;
- **Entailment Regimes:** definiše uslove pod kojim SPARQL upiti mogu da se koriste za zaključke pod RDF, RDF Schema, OWL i RIF entailment (dodela).

Popularni triple stores već imaju implementirane neke ili sve mogućnosti SPARQL 1.1.

8. ONTOLOGIJA

Do sad je prikazano kako se predstavlja i prenosi znanje formalno na mašinski čitljiv način. Tokom istraživanja tripleta i grafova korišćeni su postojeći skupovi predikata ili su jednostavno izmišljeni novi koristeći intuiciju i razmenjena iskustva o subjektima i objektima.

Ovaj neformalni pristup modelovanja podataka je radio dobro, i često je dovoljan za male, nezavisne projekte. Moguće je pisanje programa koji odgovaraju na predikte kao što su foaf:knows ili fb:film.film.starring na pouzdan način, i na osnovu akcija tih programa moguće je konstatovati da program “zna šta znači glumiti u filmu”. Ali definicija “filma” nije najbolja. Da li skup filmova uključuje značenje napravljen-za-TV? Da li je film uključen u skup filmova podobnih za Oskara? Ova pitanja možda deluju precizno, ali kako projekat raste i kako se informacije distribuiraju, precizan način predstavljanja i prenosa ovog razumevanja postaje sve značajniji.

Izrazi sastavljeni od subjekta, predikta i objekta su korisna i fleksibilna metoda za predstavljanje znanja. Diskusije o entitetima i odnosima subjekta i objekta, i predikati koji opisuju kako se ovi entiteti odnose jedan prema drugom su tradicionalno bili u nadležnosti filozofije. Još od Aristotela, filozofi su bili veoma “zabrinuti” znajući šta postoji, i kako da to opišu. I kada se misli da se zna da nešto postoji, sledeći korak je da se pronade mesto za to među svim drugim stvarima. U domenu filozofije, ovaj napor se naziva ontologija.

Semantičkim softverskim sistemima je potreban model sveta iz koga znanje sa kojim rade ima smisla, sledi dizajneri sistema su takođe ontolozi. Ali dok filozofi pokušavaju da srede redosled univerzuma, dizajneri sistema treba samo da se brinu o kreiranju redosleda i opisa relacija stvari važnih za aplikaciju.

Izgradnja modela sveta nije ništa novo za sistem dizajnere, bez obzira da li se kreiraju šeme baze podataka ili navode relacije objekata. Ali semantičkim sistemima, potrebno je izraziti ove modele na taj način da sistemi distribuirani preko weba mogu da čitaju modele i da precizno shvate kako se koriste.

Da li je važno za spisak filmova da sadrži značenje “napravljen-za-TV?”. Verovatno ne. Iako se film Sarah T. – Portrait of a Teenage Alcoholic možda nije prikazivao u bioskopima, film je produciran kao i svaki drugi bioskopski film, a ljudi koji su radili na njemu su bili poznati glumci i reditelji. Osim ako ne postoji neka specifična funkcija u aplikaciji koja treba da tretira bioskopske filmove drugačije, nema potrebe za predstavljanjem ove razlike u modelu.

Kako da se zna da se misli na istu stvar o “filmu”? Znaće se kroz akciju. Ako se predloži da svi filmovi mogu da budu nominovani za Oskara, i ako je Sarah T. na listi, znaće se da se ne misli o istoj stvari o “filmu”.

Ontologija pruža precizan rečnik sa kojim se predstavlja znanje. Ovaj rečnik dozvoljava da se odredi koji će entiteti biti predstavljeni, kako mogu da se grupišu i koje relacije ih povezuju.

Rečnik može da se posmatra kao društveni ugovor između dobavljača podataka i korisnika podataka. Što je ontologija preciznija, to je veći potencijal razumevanja kako podaci mogu da se koriste. Ali, ako je ontologija previše kompleksna, uključujući kategorije i odnose koje korisnik podataka ne treba, onda postaje konfuzna, komplikovana i teška za korišćenje, održavanje i proširivanje.

Kako se zna da li je ontologija ona “prava”? Prava je ako su korisnici i sistemi koji je

koriste u mogućnosti da uspešno sarađuju. Ovo je isti standard koji se odnosi na računarski softver. Da li program radi? Da li dobijamo pravi odgovor? Ako je tako, onda je uspešan.

U ovom poglavlju će biti razmotreno kako napraviti potpuniji model relacija i kako ih predstaviti u samom RDF-u [37].

8.1 MODELI SU PODACI

S obzirom da je RDF meta model, ontologija može da se izrazi kao RDF triple i da se čuva u grafu zajedno sa podacima koje opisuje. Važno je da se shvati da su ovi semantički modeli jednostavno još jedan oblik podataka koji nema unutrašnje ponašanje. Zanimljive stvari se dešavaju samo kada softver koji razume semantički model radi sa podacima.

Ontologije omogućavaju da se izraze formalna pravila za zaključivanje. Kada softver čita ontologije, treba da ima sve neophodne informacije da izvuče iste zaključke iz podataka. Kada su akcije softvera u skladu sa pravilima izraženim u ontologiji, kaže se da je napravljen ontološki čin.

U objektno-orientisanom (OO) programiranju je “lako zaboraviti” sa koliko pravila sistem rukuje. Na primer, podklasa u OO sistemu sadrži članove svog “roditelja” (metode i promenljive), jer je negde u sistemu softver izjavio da ima pristup članovima “roditelj” u skladu sa onim što je OO klasa.

Po dizajnu, semantički sistemi ostavljaju mogućnost izgradnje ili pozivanja neophodnog softvera zbog zaključivanja na osnovu modela. Proizvodnja konzistentnih akcija na osnovu podataka je bukvalno ono što se podrazumeva pod “semantičko programiranje”. Postoji mnogo semantičkih platformi i alata koji mogu da pomognu u razvoju obezbeđujući osnovne sposobnosti zaključivanja za uobičajene ontologije.

8.2 MODELOVANJE PODATAKA

Do sada je urađeno mnogo modelovanja podataka, viđeni su mnogi primeri o tome kako da se predstave odnosi između filmova i glumaca, restorana, kuhinje i lokacije. U svakom slučaju se govorilo o različitim klasama stvari i koja svojstva ove stvari imaju. Na primer, filmovi imaju imena, datum objavljivanja i glumce, dok restorani imaju kuhinje i lokacije. Klase se koriste za definisanje karakteristika grupe stvari, kao što su filmovi i glumci, i za određivanje odnose prema drugim klasama.

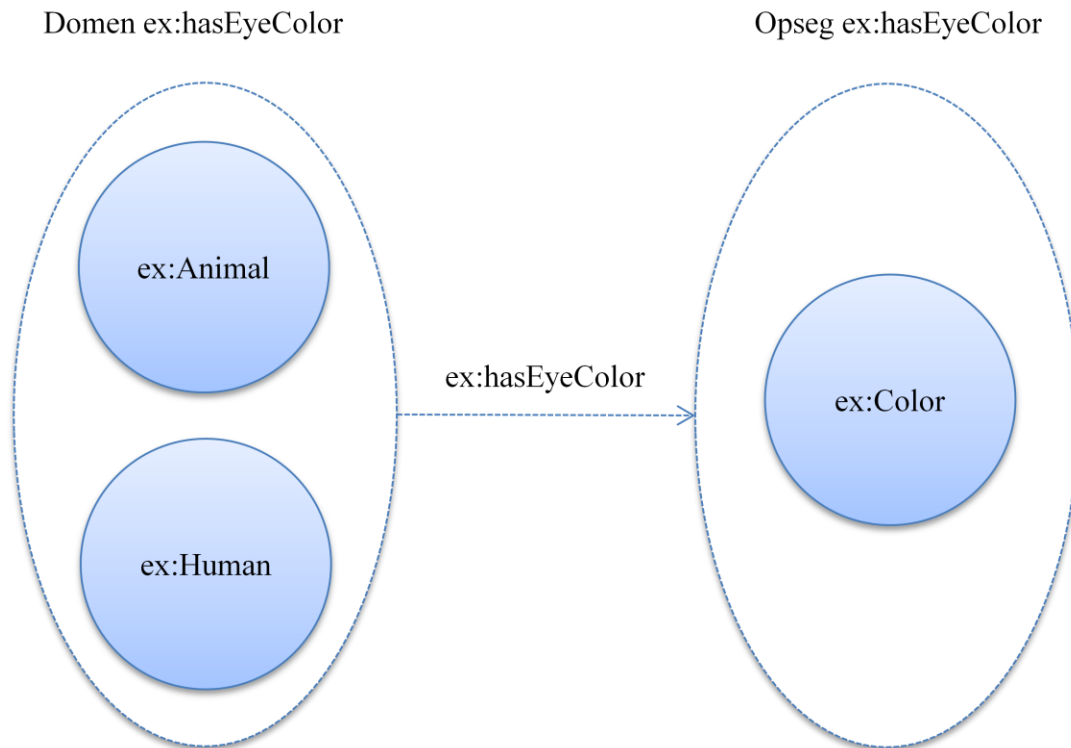
8.2.1 KLASA I SVOJSTVA

Kao objektno-orientisani model, semantički model koristi termin klasa da opiše grupu entiteta. U većini OO sistema se kaže da je objekat instanca klase zato što je realizovana iz definicije klase koja služi kao prototip ili nacrt. Jednom kada se instanca kreira, ona sadrži sve članove klase.

Semantički podaci se fokusiraju na odnose između entiteta. Semantički model je orijentisan na svojstva a ne na entitet ili objekat. Tako se semantički entiteti ne “rađaju” u klasi, već shvataju da su članovi klase zbog svojih svojstava. Ovo može da izgleda kao “suptilna” razlika, ali ovaj fokus na vezama daje semantičkom modelu veliku fleksibilnost.

Svojstva su u objektno-orientisanim sistemima definisana kao deo klase. Kada se zna iz koje je klase objekat kreiran, zna se koja svojstva sadrži. U semantičkim sistemima se svojstva definišu nezavisno. Definicija svojstva može opciono da ukaže koji tip resursa ima svojstvo.

Kolekcija tipova koji koriste svojstvo se naziva domen svojstva. Definicija svojstva može da ukaže koje tipove vrednosti svojstvo može da preuzme, predstavljajući opseg svojstva.



Sl. 38. Domen i opseg svojstva koje izražava boju očiju

Kada svojstvo ne ukazuje na svoj domen, ne može da se zaključi ništa o resursima koje opisuje jer mogu da ga koriste svi tipovi resursa. S druge strane, ako svojstvo definiše tip kao svoj domen, tada je moguće da se zaključi da je sve što je opisano svojstvom tip domena. Štaviše, ako je svojstvo definisalo nekoliko tipova kao svoj domen, može da se zaključi da je resurs opisan svojstvom svih tipova domena.

Slično tome, ako svojstvo ne odredi opseg, onda ne može da se zaključi ništa o vrednosti svojstva. Ali ako svojstvo određuje jedan ili više tipova kao svoj opseg onda može da se zaključi da vrednost svojstva mogu da budu svi tipovi određeni opsegom svojstva.

Razlika između semantičkog i OO pristupa definiciji klase i svojstva može da se istakne kroz jednostavan model. U trivijalnom modelu osobe koja pruža informaciju o boji očiju u OO sistemu moguća je definicija Animal klase sa podklasom Human koja ima promenljivu koja predstavlja boju očiju. Kada se konstruiše instanca Human klase koja predstavlja Aleksandra, instanca će da sadrži promenljivu za čuvanje njegove boje očiju, koja bi mogla da bude "braon". Zna se da je Aleksandar i "životinja" i "čovjek" zbog objekta koji ga predstavlja konstruisanom iz Human klase.

U semantičkom modelu (uz korišćenje example.org domena, kome je dat prefix ex) moglo bi da se deklarise svojstvo hasEyeColor sa domenom Animal i Human i sa opsegom Color:

```
ex:hasEyeColor rdf:type rdf:Property .
ex:hasEyeColor rdfs:domain ex:Animal .
```

```
ex:hasEyeColor rdfs:domain ex:Human .  
ex:hasEyeColor rdfs:range ex:Color .
```

Sledećim iskazom može da se predstavi da Aleksandar ima braon oči:

```
<http://its.rs/people/aca> ex:hasEyeColor <http://rdf.freebase.com/ns/en.brown> .
```

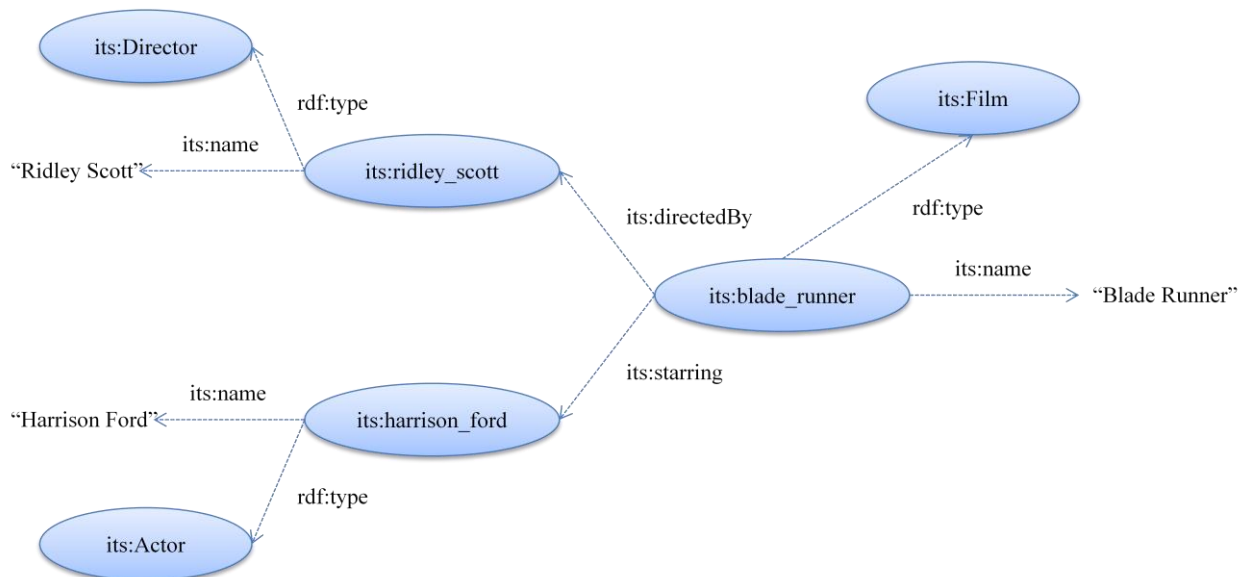
Na osnovu ovog iskaza i definicije hasEyeColor može da se zaključi da je Aleksandar i životinja i čovek i da je braon boja, iako nije bilo drugog iskaza o Aleksandru ili braon boji.

Semantičke klase su definisane svojstvima. Takođe, moguće je definisati klasu u odnosu na vrednost svojstva. Na primer, moguće je definisanje klase koja se sastoji od ljudi koji imaju braon oči. Nije potrebno da se potvrdi da je Aleksandar član ove klase jer njegovo članstvo može da se zaključi na osnovu tvrdnje da ima braon oči.

Naravno, uvek je moguć eksplicitni iskaz da je resurs član klase. Ovo se dešava veoma često tako da je većina RDF serijalizacija razvila stenografije za to izražavanje. Iako se često potvrđuju članovi klase, važno je da se razume da članstvo klase može da se zaključi kroz nekoliko mehanizama u okviru semantičkih modela.

8.2.2 MODELOVANJE FILMOVA

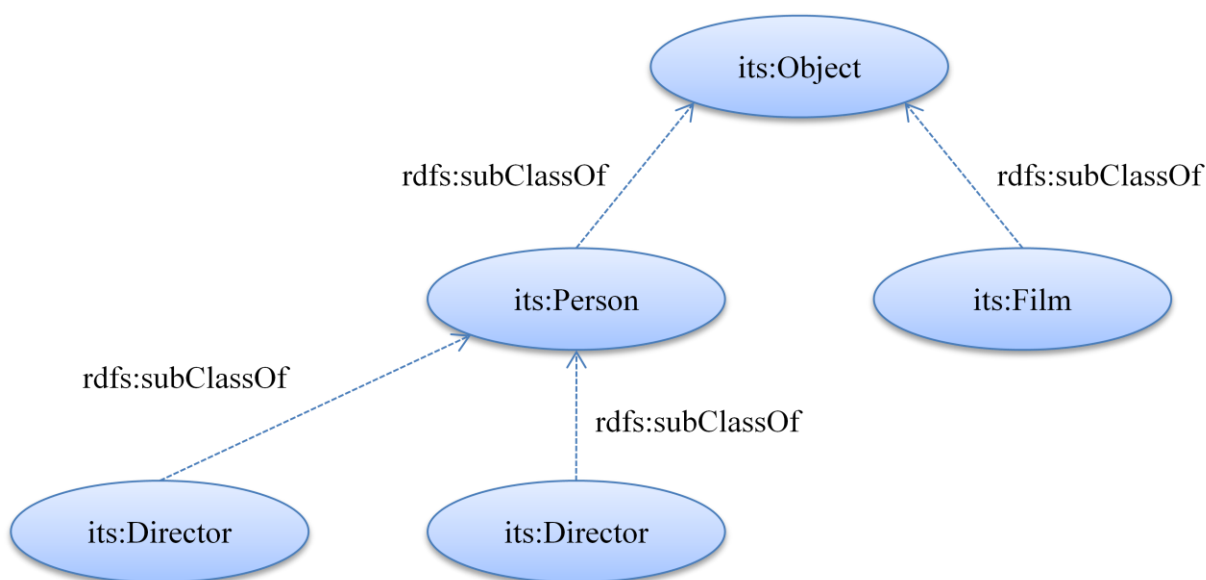
U sledećem primeru, biće izgrađen model podataka za skup podataka o filmovima. U tom skupu podataka biće tri različite stvari koje su povezane, filmovi, glumci i reditelji, i za svaku od njih će biti kreirane klase. Glumci imaju svoj performans (izvode “ulogu”) u filmu, režiseri režiraju film, a sve ima ime. Slika 39 prikazuje graf za film Blade Runner i pokazuje da je Ridley Scott režiser, a Harrison Ford glumac. Za ID i tipove će se koristiti “its” imenski prostor, što odgovara <http://www.its.rs/film#>. Koristi se rdf:type svojstvo koje pokazuje koji tip klase je nešto. Biće predstavljeno mnogo rečnika koji koriste RDF, RDFS i OWL.



Sl. 39. Graf koji opisuje film Blade Runner

Kao što se vidi na slici 39 klasa its:Film koristi svojstva its:starring i its:directedBy, i sve tri klase koriste svojstvo its:name. Sada je potrebno kreirati šemu koja opisuje klase i svojstva. Na početku je potrebno definisati jednostavnu hijerarhiju klasa. Da bi se ponovo upotrebilo svojstvo name kroz više klasa, biće “prikačeno” za klasu its:Object, a zatim će druge klase biti

podklase za dati objekat. Takođe će biti dodata `its:People` klasa koja je superklasa za `its:Actor` i `its:Director`, tako da će kasnije biti moguće dodavanje svojstva zajedničkih za ljude kao što su datumi rođenja i porodični odnosi. Koristiće se `rdfs:subClassOf` svojstvo za povezivanje podklasa sa superklasama.



Sl. 40. Hijerarhija klasa za film ontologiju

Pored hijerarhije klasa, potrebno je deklarirati opseg i domen svojstva koji se koriste. Kao što je pomenuto ranije, opseg je izvor tipova svojstva, a domen je odredište tipova svojstva. Slika 40 prikazuje triplete za definisanje svojstva i tipova za podatke o filmu. Deklarisani su `its:Object`, `its:Film` i `its:Director` kao tip `owl:Class`, `its:starring` i `its:directedBy` kao tip `owl:ObjectProperty`, i `its:name` kao tip `owl:DatatypeProperty`. To je zato što je `its:name` literalna vrednost, a druga svojstva su resursi. Takođe su definisani domen i opsezi. Svojstvo `its:starring` ima domen `its:Film` i opseg `its:Actor`, `its:directedBy` ima domen `its:Film` i opseg `its:Director`.

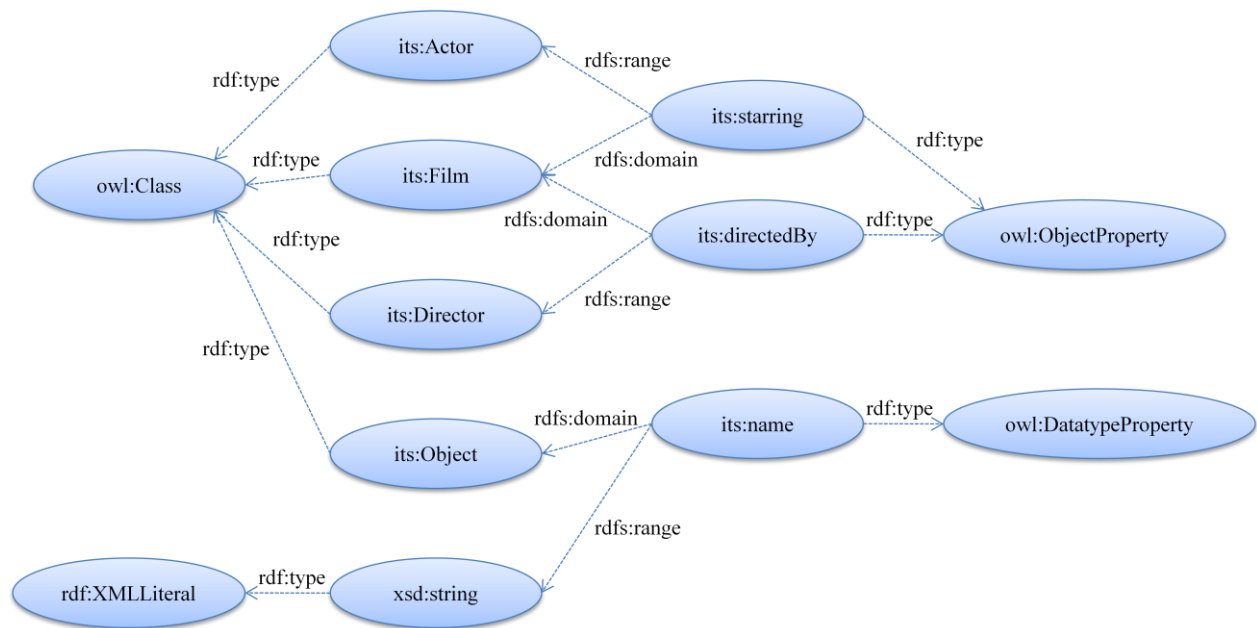
Svojstvo `its:name` je definisano na drugačiji način jer `name` ukazuje na string vrednost. Opseg `name-a` je `xsd:string`, koji je XML literalni tip podataka. Svi XML literalni tipovi podataka kao što su `float`, `integer`, `string` i `dates` mogu da se koriste u OWL-u dok god je svojstvo koje ih koristi tip `owl:DatatypeProperty`.

Imenovanje Svojstva i Klasa u Ontologijama

Kada se ime svojstva ili klase sastoji od više reči, postignut je konsenzus za korišćenje camel-casing. Ovo znači da umesto `directed_by` ili `directedby` treba da se koristi `directedBy` za ime “directed by” svojstva. Prvo slovo imena klase je obično veliko slovo, a prvo slovo svojstva je malo slovo.

Ne postoji konsenzus za druge instance resursa jer su ovi identifikatori često zavisni od sistema. Međutim, kratki i čitljivi identifikatori su uvek bolji od čisto mašinski generisanih kodova.

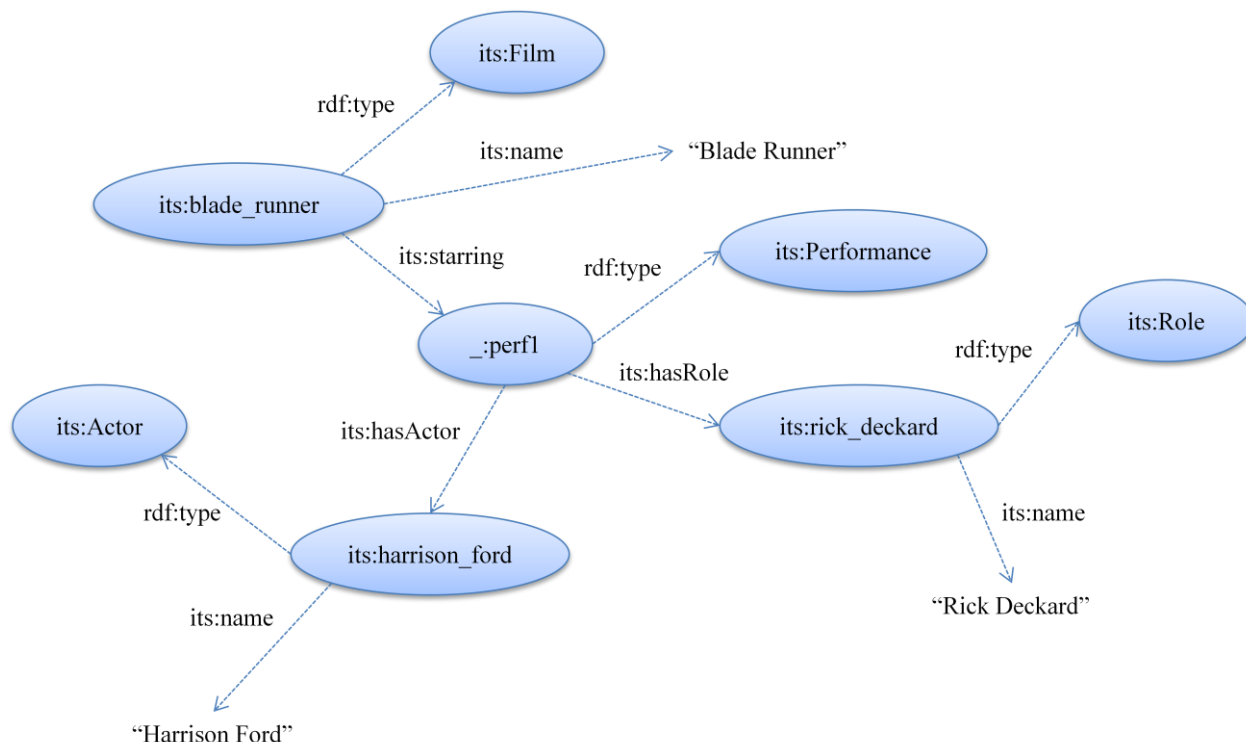
Na grafu na slici 41 se vidi da definisanje klasa i svojstava može postati brzo veoma složeno. Ovo je mesto gde alati i API mogu da budu od velike pomoći.



Sl. 41. Definicije svojstava za film ontologiju

Uobičajeni problem kod modelovanja podataka se javlja kada je potrebno da se daju iskazi o relacijama. Primer za to u skupu podataka filmova bi bio dodavanje uloge koju je glumac igrao u filmu. Dakle, umesto da se kaže da je Harrison Ford glumio u filmu Blade Runner, treba da se kaže da je Harrison Ford imao ulogu Rick Deckarda u filmu Blade Runner. Nije moguće jednostavno dodavanje svojstva uloga, jer uloga nije svojstvo glumca ili filma, to je svojstvo performansa Harrison Forda u Blade Runner.

Proces kreiranja subjekat-predikat-objekat iskaza u subjektu se naziva “reifikacija” u RDF. RDF standard ima rečnik za reifikaciju tripleta, ali je preterano opširan i komplikovan i ne sarađuje dobro sa postojećim RDF API-em ili sa OWL-om ili RDF Schema (RDFS). Umesto toga, jednostavnije je imati `its:starring` svojstvo pokazivač na klasu `its:Performance` koja ima `its:actor` i `its:role` svojstva. Sa ovom jednostavnom promenom sada je moguće imati više iskaza o performansi glumca. Slika 42 prikazuje novu šemu.



Sl. 42. Klasa performans u film ontologiji

Klasa `its:Performance` se razlikuje od `its:Actor`, `its:Film` i `its:Director` jer umesto da opisuje primarni koncept ili stvar u podacima, ona opisuje relaciju između dve stvari. Bez `its:Actor`, `its:Film` ili `its:Role` vezane na nju, specifična `its:Performance` klasa je besmislena, ona zavisi od stvari koje povezuje da bi izrazila značenje. U RDF je uobičajeno korišćenje praznih čvorova za reifikaciju relacija, jer oni ne mogu da imaju suštinski ID koji dolazi iz podatka koji su modelovani.

Obrazac reifikacije relacija je uobičajen u semantičkom modelovanju podataka. Ona ima veću efikasnost po cenu veće složenosti, nije potrebno reifikovati sve relacije, ali slobodno može da se koristi kada podaci to zahtevaju.

8.3 OWL

Web Ontology Language (OWL) je RDF jezik razvijen od strane W3C za definisanje klasa i svojstava, kao i za omogućavanje rasuđivanja i zaključivanja preko relacija. OWL je razvijen kao ekstenzija RDFS-a, ranije, jednostavnije šeme rečnika, i baziran je na mnogo prethodnog rada u razvoju jezika ontologija. OWL je trenutni W3C standard za definisanje semantičkih web šema. Alati i API koji imaju podršku za OWL se brzo razvijaju.

OWL je složen jezik sa mnogo kompleksnih delova. Sam OWL je podeljen u tri pod-jezika povećane složenosti i ekspresivnosti: OWL-Lite (najjednostavniji), OWL DL i OWL Full. Biće predstavljeno sasvim dovoljno OWL-Lite za definisanje tipova i svojstva, jer su to najkorisnije aplikacije OWL.

Ceo rečnik OWL koristi URI u RDF, RDFS i OWL imenski prostor, i takođe koristi XML Schema literalne definicije. Najvažnije klase za kreiranje ontologija su:

- **owl:Thing** – Klasa svih stvari u OWL. Sve klase su implicitno podklase ove klase, i sve pojedinačne OWL instance su implicitno instance owl:Thing. Upotreba ove klase je slična superklasi Object u nekim objektno-orijentisanim programskim jezicima. Svojstva sa opsegom ili domenom rdf:Resource mogu da se koriste u bilo kojoj instanci;
- **owl:Class** – Klasa RDF resursa koji su klase. Sve klase su instance rdfs:Class;
- **owl:DatatypeProperty** – Klasa svih svojstava koja imaju opseg koji je literalan i instance rdfs:Datatype;
- **owl:ObjectProperty** – Klasa svih svojstava koja imaju opseg koji je instanca owl:Class;
- **rdf:XMLLiteral** – Klasa svih XML literalnih vrednosti definisanih u XML Schema specifikaciji. Generalno, većina ontologija koristi isključivo XML Schema literalne vrednosti, jer su veoma dobro definisane i razumljive. Ovo je podklasa rdfs:Literal i instanca rdfs:Datatype.

Najvažnija svojstva za kreiranje ontologija su:

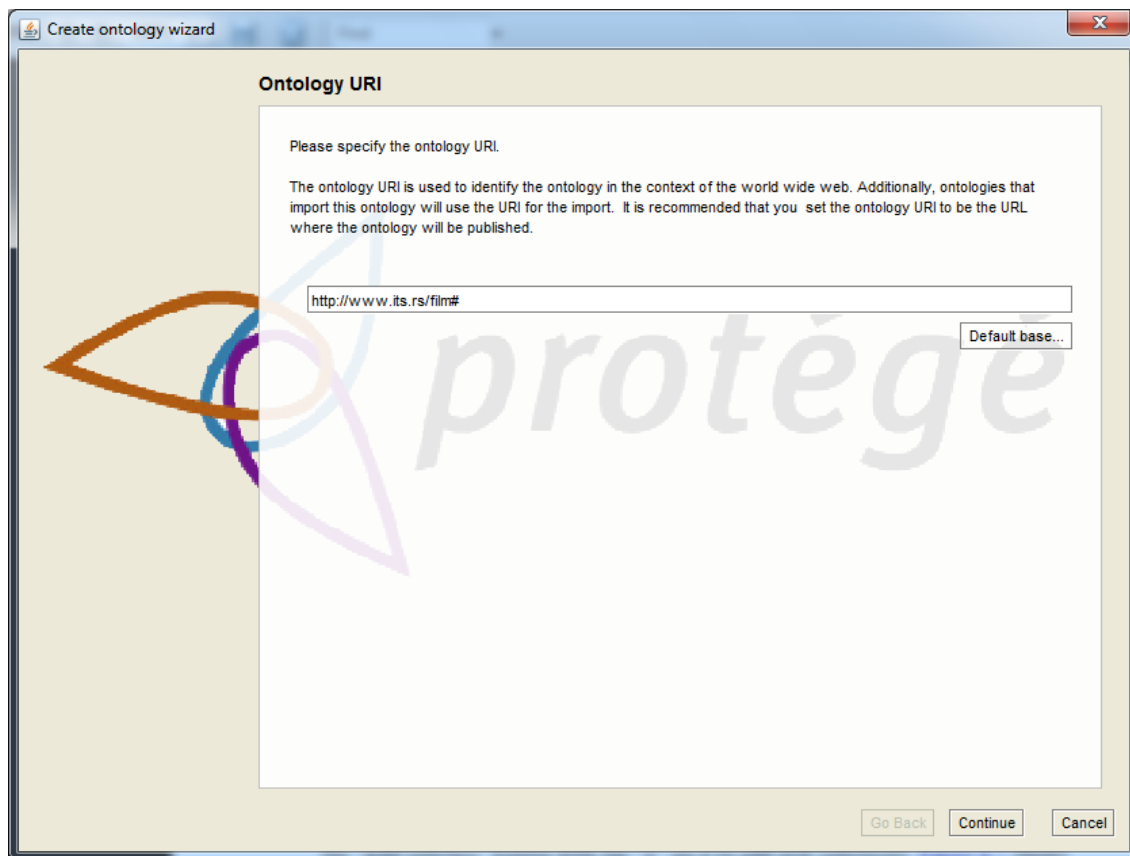
- **rdf:type** – Tip resursa. Ovo svojstvo određuje da je resurs instanca određene RDFS klase;
- **rdfs:subClassOf** – Određuje da je klasa, podklasa druge klase, dakle sve instance podklase su takođe instance superklase;
- **rdfs:domain** – Određuje da svojstvo ima domen u određenoj klasi. U praksi, ovo znači da kada se određeno svojstvo koristi u triplu, subjekat će uvek biti instanca klase određene rdfs:domain svojstvom u ontologiji;
- **rdfs:range** – Određuje da svojstvo ima opseg određene klase. U praksi, ovo znači da kada se određeno svojstvo koristi u triplu, subjekat će uvek biti instanca klase određene rdfs:range svojstvom u ontologiji. Ovo može da se koristi u određivanju literalnog opsega svojstva takođe.

8.4 PROTÉGÉ

Razvoj velikih OWL ontologija koristeći samo triplete je težak. Postoji veliki broj alata za izgradnju i održavanje velikih OWL ontologija. Protégé je alat za OWL koji je “bogat” funkcijama. Protégé je JAVA GUI otvorenog koda razvijen na Stanford Univerzitetu, izdanja su dostupna na <http://protege.stanford.edu/> [38].

8.4.1 KREIRANJE NOVE ONTOLOGIJE

Kada se pokrene Protégé, otvara se meni gde je prva stavka kreiranje nove OWL ontologije (Create new OWL ontology). Nakon izbora “Create new OWL ontology”, potrebno je da se navede bazni URI sa kojim će svi RDF URI da budu povezani. Za film ontologiju će bazni URI biti <http://www.its.rs/film#>.

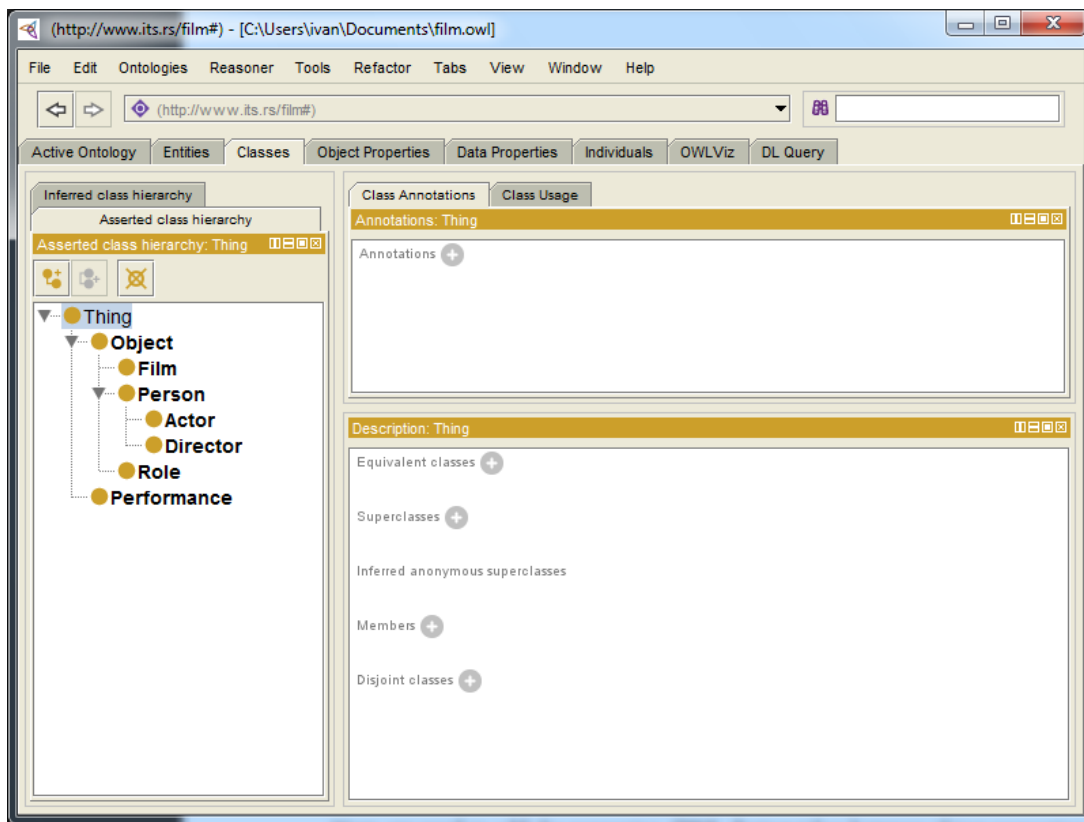


Sl. 43. Kreiranje ontologije u Protégéu

Sledeći korak je izbor fizičke lokacije na kojoj će fajlovi projekta biti snimljeni. Klikom na Finish se otvara editor ontologije.

8.4.2 IZMENA ONTOLOGIJE

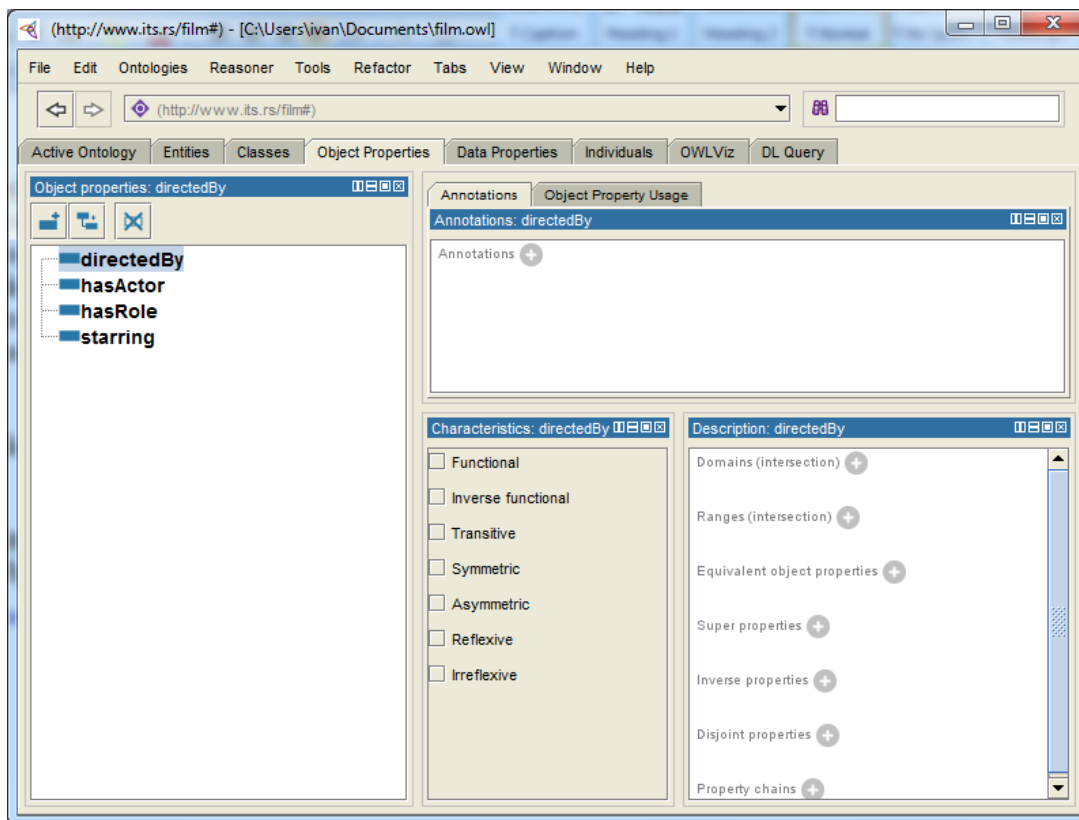
Pri izgradnji hijerarhije klasa prvi korak je izbor taba “Classes”, a zatim izbor klase Thing. Ovo je root klasa u OWL i sve klase su njene podklase. Klikom na “Add subclass” dugme sa “+” na njemu dodaje nove podklase. Slika 44 prikazuje hijerarhiju film ontologije, izgrađenu dodavanjem pojedinačnih podklasa.



Sl. 44. Kreiranje hijerarhije klasa u Protégéu

Sledeći korak je dodavanje svojstva. OWL pravi razliku između svojstva objekta koja upućuju na objekte i na svojstva tipova podataka koja upućuju na literalne vrednosti. Svojstvo se dodaje izborom “Object Properties” taba, a zatim klikom na “Add property” dugme sa “+” na njemu. Nakon dodavanja svojstva, moguć je izbor domena i opsega klikom na “+” dugme pored “Domains” i “Ranges”. U domen i opseg dijalozima može da se upiše naziv klase ili da se izabere klasa iz hijerarhije.

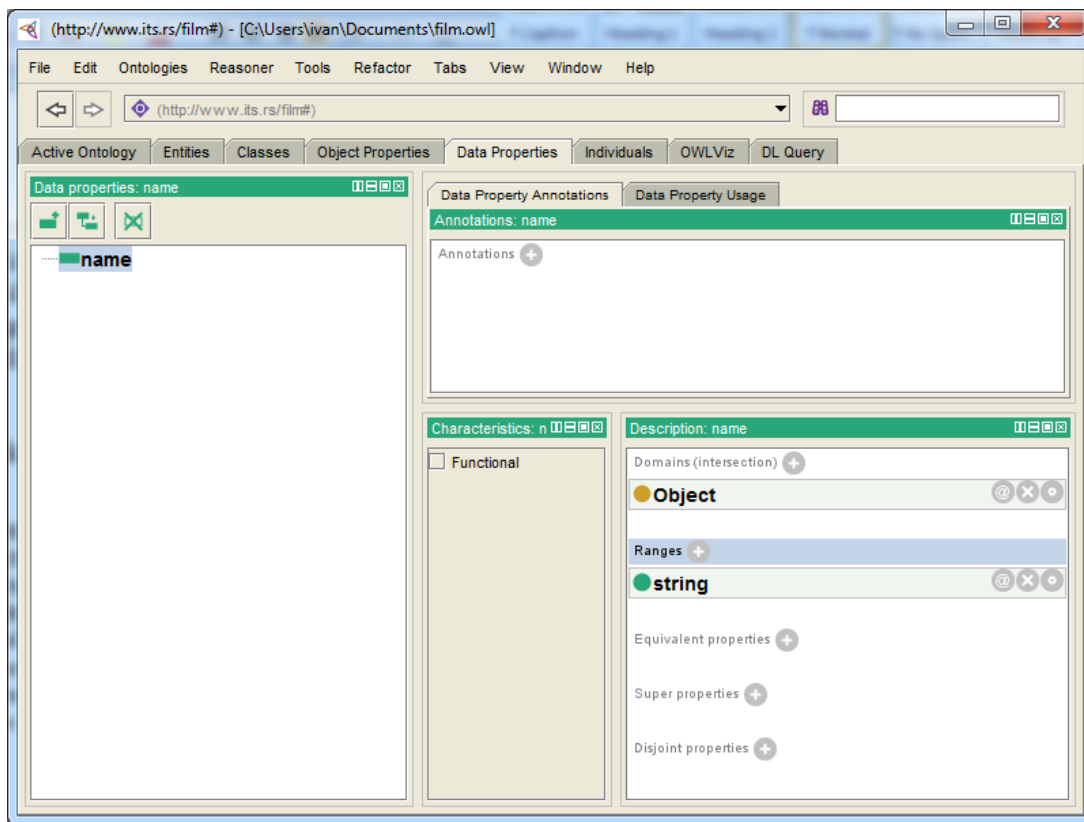
Slika 45 prikazuje popunjena svojstva. Takođe su uneti domen i opseg za svako svojstvo.



Sl. 45. Kreiranje svojstva objekta u Protégéu

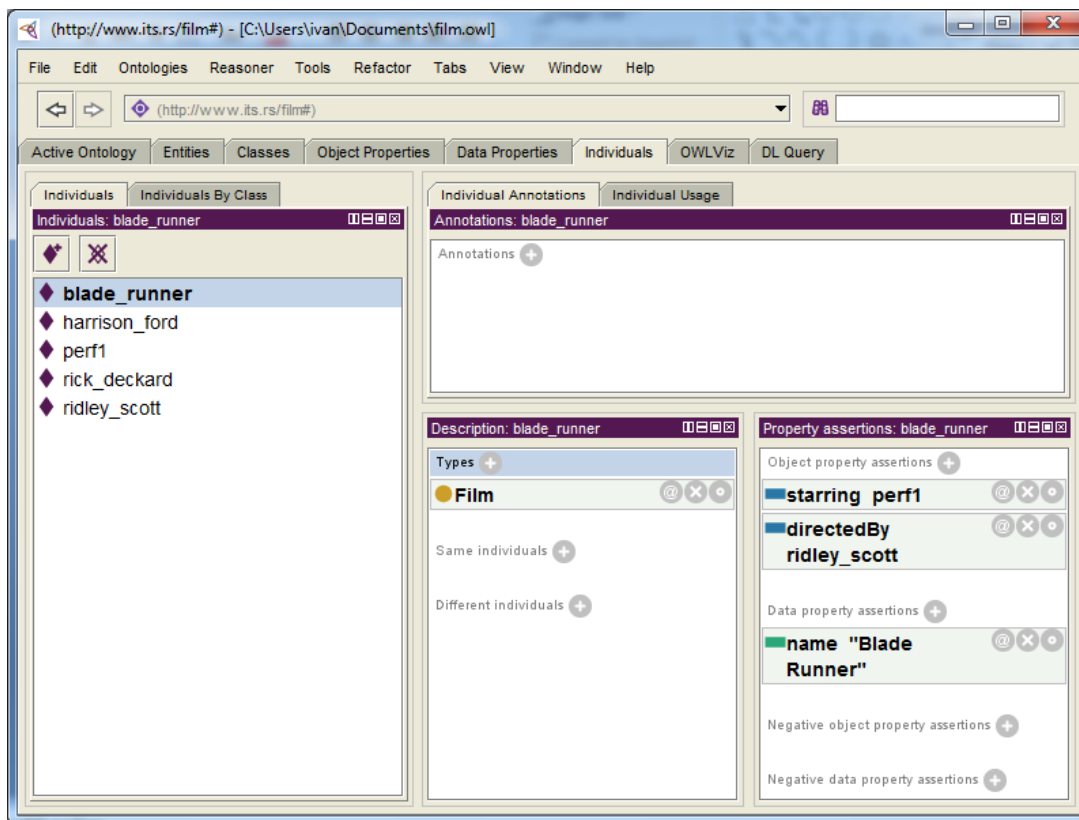
Konačno, potrebno je da se opiše svojstvo name. To je svojstvo tip podatka, tako da je potrebno da se izabere “Data Properties” tab, a zatim klikne na “Add property” dugme sa “+” na njemu. Domen se dodaje klikom na dugme “+” pored “Domains” i izborom klase. Opseg se dodaje klikom na dugme “+” pored “Ranges” i izborom u ovom slučaju “string” literalnog tipa iz liste tipova podataka poznatih Protégé.

Slika 46 prikazuje popunjeno svojstvo.



Sl. 46. Kreiranje svojstva podataka u Protégéu

Kada se definiše ontologija, Protégé podržava kreiranje instanci klasa. Uopšteno govoreći, u većini slučajeva učitavanje podataka će ići preko skripte ili programa umesto ručno, ali “Individuals” tab omogućava popunjavanje podataka, što je zgodno za testiranje ontologije. Za primer film ontologije je potrebno da se unese vrednost za tip i svojstvo za svaku instancu. Slika 47 prikazuje podatke za film Blade Runner.



Sl. 47. Kreiranje pojedinačne instance u Protégéu

OWL ontologija se u Protégéu snima u RDF/XML formatu.

U Protégéu pored osnovnih postoje i opcione informacije koje mogu da se dodele klasama i svojstvima. Ovi modifikatori dozvoljavaju uklanjanje nejasnoća o tome kako su stvari povezane.

8.4.3 FUNKCIONALNA I INVERZNO FUNKCIONALNA SVOJSTVA

Označavanje svojstva kao funkcionalno znači da postoji samo jedan objekat za dati subjekat sa svojstvom. Na primer, u film ontologiji, `has_actor` i `has_role` bi bila funkcionalna svojstva jer za svaki pojedinačni performans može postojati samo jedan glumac i samo jedna uloga. Označavanje svojstva kao inverzno funkcionalno znači da postoji samo jedan subjekat za dati predikat ili objekat. Na primer, u film ontologiji, `starring` bi mogao da se označi kao inverzno funkcionalan jer za svaki performans može da bude u samo jednom filmu.

Klase `owl:FunctionalProperty` i `owl:InverseFunctionalProperty` su podklase `owl:ObjectProperty`, tako da je deklarisanje svojstva kao funkcionalno ili inverzno funkcionalno moguće navođenjem tipa svojstva korišćenjem `rdf:type`.

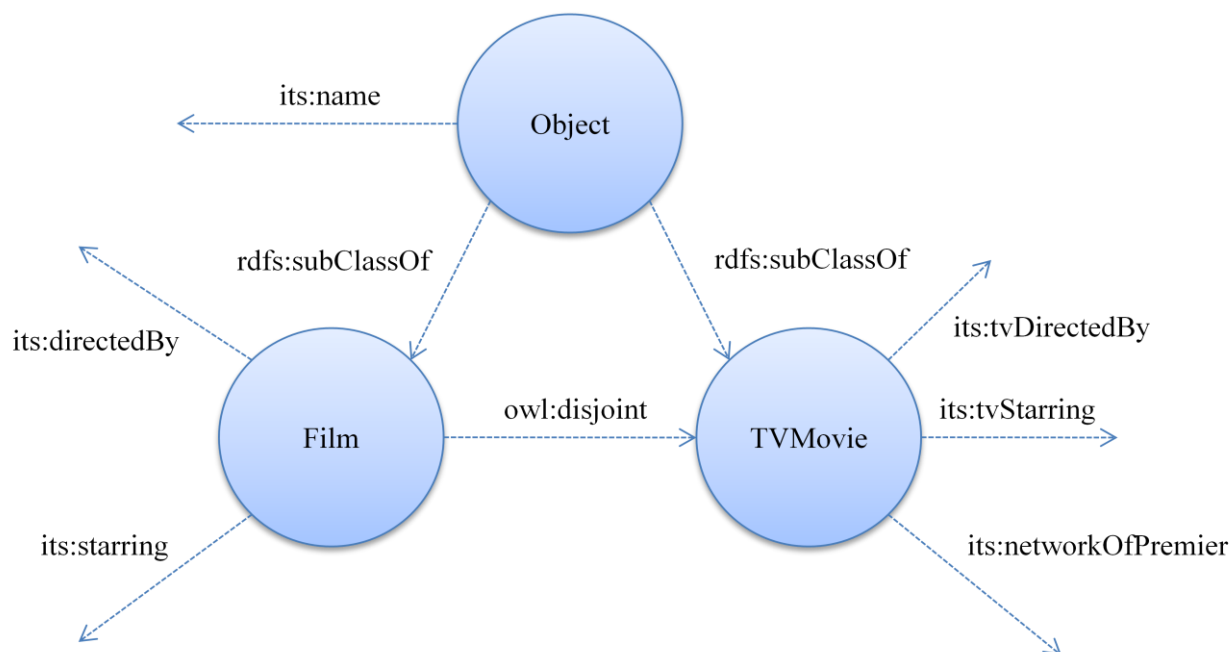
8.4.4 INVERZNA SVOJSTVA

RDF iskazi podrazumevaju smer relacije koji ide iz domena subjekta prema opsegu objekta. Tako se kaže da su filmovi u režiji režisera, ali obrnuto, režiseri režiraju filmove nije posebno obuhvaćeno modelom. Umesto modelovanja nezavisnog drugog skupa tvrdnji koji istom merom vraćaju film-režiser relaciju, OWL dozvoljava iskaz:

```
<http://www.its.rs/film#directed> owl:inverseOf  
<http://www.its.rs/film#directedBy> .
```

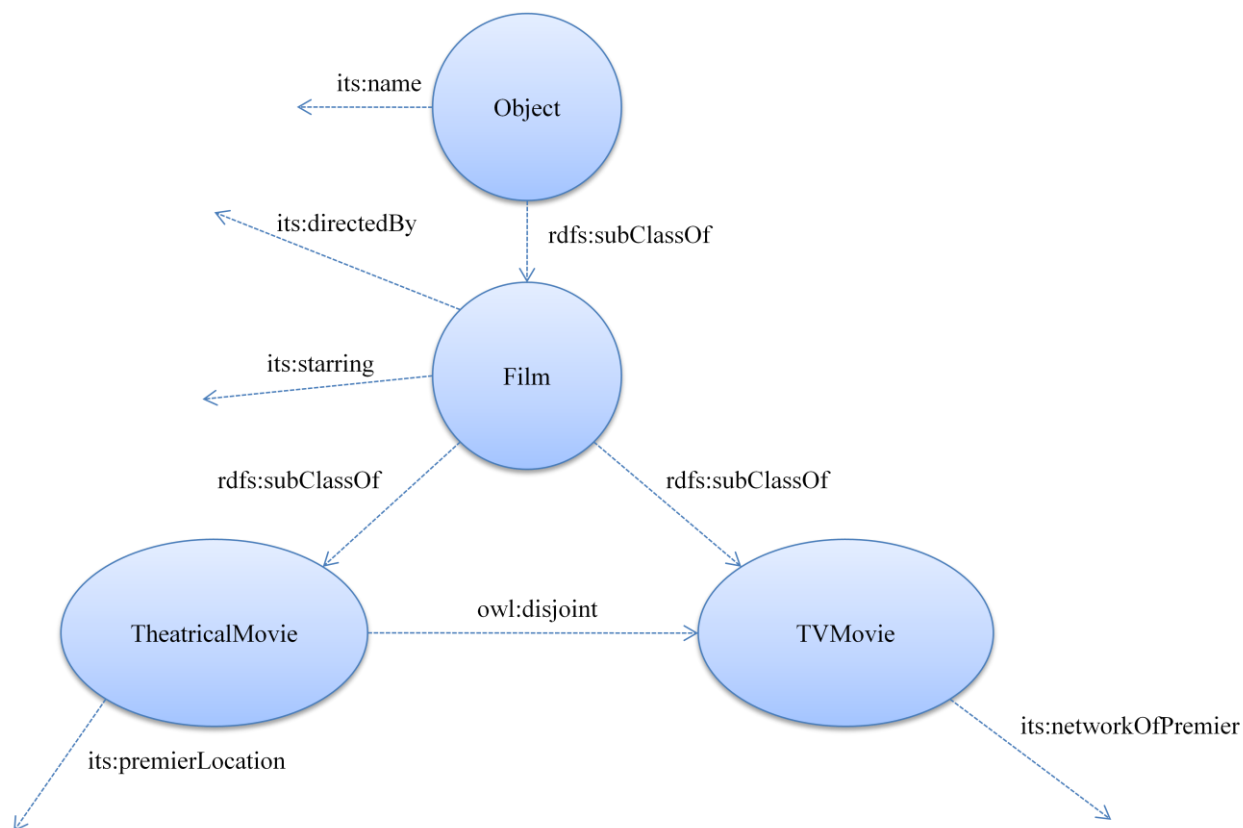
8.4.5 DISJOINT KLAZE

Članovi klase koji su razdvojeni od druge klase ne moraju nužno da budu članovi druge klase. Na primer, ako bi trebala da se napravi razlika između napravljen-za-TV od bioskopskog filma, mogla bi da se definiše TVMovie klasa koja je razdvojena od Film klase. To bi sprečilo TVMovie klasu da bude podklasa Film klase. Ovo nije dobar izbor, jer bi onda trebalo da se definišu svojstva kao što je režiser i performans odvojeno za TVMovie klasu.



Sl. 48. Loš model koji se bavi napravljen-za-TV filmove

Korisnije bi bilo da se definiše TVMovie klasa kao podklasa Film klase i da se kreira nova klasa TheatricalMovie za predstavljanje bioskopskih filmova. TheatricalMovie klasa bi takođe bila podklasa Film klase, ali bi bila razdvojena od TVMovie klase.



Sl. 49. Bolji model koji se bavi napravljen-za-TV filmove

Kada se razvija ontologija (ili model), eksplicitno se biraju opisi aspekta koji je važan za aplikaciju. Odluka o tome šta da se isključi iz ontologije je veoma važna, kao i odluka šta da se uključi. Dobra ontologija pruža dovoljno informacija, ali nikad više nego što je potrebno.

Važno je biti pragmatičan kada se razvijaju ontologije, one se ne završavaju u sebi, već su jednostavno okruženje za izraze. Ako niko ne kodira znanje pomoću ontologije, ili ako nije napisana nijedna aplikacija koja radi nešto korisno sa znanjem, vrednost ontologije je nula.

U poglavlju 7 su vršeni upiti sa podacima o filmovima bez ikakvog znanja o ontologiji. Zbog jednostavnosti, podaci o filmovima nisu imali nijednu `rdf:type` tvrdnju, a ipak su SPARQL upiti radili odlično. Iako se to ne smatra dobrom praksom, ideja je da za mnoge svrhe, jednostavno razumevanje podataka, bez obzira na značajan ontološki model, može biti “dovoljno dobro”.

Takođe je zanimljivo istaći da su podaci instance i definicije u semantičkim sistemima predstavljene na isti način i često postoje u istom skladištu. Metapodaci su podaci. Ovo znači da je moguće ispitivanje metapodataka na isti način kao i podataka instance, takođe je moguće izvršenje upita istovremeno. Umesto odvojenih rečnika podataka ili DDL koji su izgrađeni, održavanja i korišćenja odvojeno od samih podataka, opis podataka je deo samih podataka. Ovo je kritičan deo onoga što čini semantičke podatke prenosivim i ponovo iskoristljivim.

8.5 OWL-S

8.5.1 *SERVISI U SEMANTIČKOM WEBU*

Semantički web se razvija ubrzano. Uskoro će biti moguć pristup web resursima na osnovu sadržaja, a ne samo na osnovu ključnih reči. Semantički web bi trebao da omogući veći pristup ne samo sadržaju, već i servisima na webu. Korisnici i softverski agenti bi trebalo da budu u stanju da otkriju, aktiviraju, sastavljaju i kontrolišu web resurse i nude određene usluge i imaju određena svojstva, i treba da budu u mogućnosti da to urade sa visokim stepenom automatizacije ako to žele. Moćni alati bi trebalo da budu omogućeni opisom servisa kroz njegov ciklus. OWL-S je ontologija servisa koja čini ove funkcionalnosti mogućim.

Jednostavni ili “atomic” (nedeljivi) servisi su oni gde se web-dostupni računarski program, senzor ili uređaj aktiviraju zahtevom, obavljaju zadatak, i eventualno proizvode jedan odgovor tražiocu usluge. U atomic servisima ne postoji interakcija između korisnika i servisa. Na primer, servis koji vraća poštanski broj kada mu se zada adresa, spada u ovu kategoriju. Nasuprot njima, kompleksni ili “composite” servisi su sastavljeni od više primitivnih servisa i mogu da zahtevaju proširenu interakciju ili razgovor između tražioca i skupa servisa koji se koriste. Jedna interakcija sa www.amazon.com za kupovinu knjige je ovakva, korisnik pretražuje knjige po raznim kriterijumima, možda čita recenzije, može ili ne mora da odluči da kupi knjigu, i daje informacije o kreditnoj kartici i poštansku adresu. OWL-S je namenjen da podrži obe kategorije servisa. Kompleksni servisi uključuju mnogo elemenata ontologije [39].

8.5.2 *OWL-S ONTOLOGIJA*

Razvoj OWL-S ontologije ima za cilj da omogući sledeće zadatke:

- **Automatsko otkriće web servisa** – sa razvojem semantičkog weba, mnogi web servisi će biti dostupni na webu, izvršavajući najrazličitije zadatke. OWL-S će da pomogne softverskim agentima da otkriju web servis koji će da ispuni specifične potrebe u okviru kvalitetnih ograničenja bez potrebe za intervencijom korisnika;
- **Automatsko aktiviranje web servisa** – generalno, potrebno je da se napiše specifičan program za aktiviranje web servisa uz pomoć WSDL opisa. OWL-S će da stvori mogućnost softverskim agentima da automatski pročitaju opis web servisa, njegove ulazne i izlazne vrednosti i da aktiviraju servis;
- **Automatsko sastavljanje i razmenu podataka između web servisa** – na webu gde su mnogi servisi na raspolaganju, biće moguće izvršenje kompleksnih zadataka, koji uključuju koordinisano aktiviranje različitih web servisa. OWL-S će da omogući automatsko izvršenje ovih zadataka.

OWL-S ontologija se sastoji iz tri dela:

- **Service profile** se koristi da opiše šta servis radi. Ova informacija je primarno namenjena za korisničko čitanje, i sadrži naziv servisa i opis, ograničenja na primenljivost i kvalitet usluge, izdavača i kontakt informacije;
- **Process model** opisuje kako klijent može da komunicira sa servisom. Ovaj opis sadrži skup ulaznih i izlaznih vrednosti, preduslove i rezultate izvršenja servisa;
- **Service grounding** precizira detalje koji su potrebni klijentu za komunikaciju sa servisom, kao što su komunikacioni protokoli, formati poruka, brojevi portova itd.

8.5.3 OWL-S i WSDL

OWL-S zahteva dodatni opis za punu specifikaciju groundinga, koja se najčešće koristi kod WSDL. Iako oba jezika ciljaju na različite nivoe specifikacije, postoji ukrštanje između njih:

- OWL-S atomic proces odgovara WSDL operaciji;
- Ulazne i izlazne vrednosti OWL-S atomic procesa odgovaraju WSDL porukama;
- Tipovi ulaznih i izlaznih vrednosti OWL-S atomic procesa odgovaraju WSDL apstraktnim tipovima.

8.6 ONTOLOGIJE SIGURNOSTI

Sigurnost i privatnost informacija igra sve značajniju ulogu u svetu u kojem je WWW centar poslovanja, industrije, finansija, edukacije i e-uprave. Tokom poslednje decenije došlo je do značajnog progressa u oblasti sigurnosti, u rasponu od teorijskih osnova i modela, preko jezika, metoda i alata, do razvoja aplikacija i sigurnosnih sistema. Dobro uspostavljena sigurnosna praksa biće od velikog značaja za uspeh semantičkog weba i potrebe skladištenja, održavanja i obrade informacija izvedenih na način koji zadovoljava sigurnosne uslove kao što su autentifikacija, autorizacija, integracija podataka i privatnost.

Predloženo je više sigurnosnih ontologija koje će omogućiti visok nivo sigurnosti web resursa, servisa i agenata. Primeri u ovom radu imaju za cilj da predstavljaju sloj apstrakcije na vrhu raznih postojećih sigurnosnih standarda kao što su XML Signature, WS-Security i drugi. Predloženo je više sigurnosnih ontologija koje su dizajnirane da predstavljaju dobro poznate sigurnosne tehnike u pogledu njihovih karakteristika, kao što su mehanizmi podržani uverenjem, notacijom itd. Osim toga, predložena je ontologija za notaciju privatnosti. Ilustrovano je kako ontologije mogu da se koriste da pruže sigurnosne oznake za Semantičke Web Servise (SWSs) i agente, dati su detalji o sigurnosnim mogućnostima i potrebama servisa i agenata. Ontologije su proširene da zadovolje kompleksnije sigurnosne politike od onih koje mogu da se izraze sa osnovnim sigurnosnim i politika za privatnost [40].

Na osnovu opisa ovih ontologija, razvijeni su algoritmi za identifikaciju podudaranja parova SWSs i agenata koji zadovoljavaju uslove i politike.

8.6.1 ONTOLOGIJA UVERENJA

Proces utvrđivanja i verifikacije identiteta uputioca zahteva u web aplikaciji, tzv autentifikacija, je često osnova odluke da li je pristup dozvoljen ili ne. Autentifikacija je zasnovana na nekom znaku, nazvanom uverenje, koje uputilac zahteva zna ili ima. Uverenja su jedna od raznih poznatih autentifikacionih tehnika kao što su korisničko ime-lozinka prijava, javni i privatni ključevi ili sertifikati. Cilj ove ontologije je mogućnost određivanja ograničenja za kontrolu pristupa web stranici ili web servisu koji koristi autentifikaciju kao uslov za ovlašćeni pristup.

Postoji razlika između "SimpleCredential" i "ComposedCredential" klase. "Cookie, Login, Key, Certificate, BioMetric i OneTimePassword" su podklase "SimpleCredential" klase koja je klasa najvišeg nivoa. Sve podklase su prazni skupovi. "Public Key" i "Symmetric Key" su razdvojene podklase "Key" klase. "Certificate" klasa je specijalizovana za "X509Certificate", i dalje za specifične klase X509 sertifikate u XML Signature.

Definisane su neke od najčešće korišćenih klasa koje su sastavljene od uverenja kao što su

“IDCard” i “SmartCard”. Na primer, smart kartice mogu da sadrže podatke kao što su ključevi, biometrijski šabloni ili PIN-ovi za autentifikaciju. Sastavljena uverenja često sadrže jednostavna uverenja, kao što su modelovana u ontologiji svojstvom “credential”. Date su razne specijalizovane identifikacione kartice. “SimpleCredential” je takođe podklasa “ComposedCredential” klase. Ontologija je proširiva tako da može da dozvoli još klasa uverenja ili svojstava.

Svojstva i druga ograničenja su definisana u ontologiji. Na primer, “LoginWithPassphrase” klasa ima dva svojstva “loginName” i “passphrase”, oba string tipa. Koristi se OWL “restriction” koncept za izražavanje kardinalnosti ovih svojstava za “Login” klasu ograničenu na jedan. Ovo znači da svako “LoginWithPassphrase” uverenje dolazi uz tačno jedno korisničko ime i jednu lozinku. Za “Certificate” klasu je definisano “object” svojstvo, koje je svojstvo OWL klase sa range tipom. Svojstvo “assoc” je povezano sa svakim podatkom sertifikata, kao što je ime izdavaoca, serijski broj itd. U specijalnom slučaju XML Signature sertifikata, “object” svojstvo je ograničeno na tip “X509Data”. Ova klasa je definisana da bude ekvivalentna X509 elementu XML Signature definicije. Na ovaj način je ontologija standardizovana.

8.6.2 ONTOLOGIJA SIGURNOSNIH MEHANIZAMA

Predložena je ontologija koja omogućava interfejs visokog nivoa apstrakcije između raznih sigurnosnih standarda i notacija. Definisano je nekoliko svojstava za klasu najvišeg nivoa “SecurityMechanism”. Na primer, ontologija definiše objekat svojstvo “syntax” koje ima klasu “Syntax” kao opseg, drugo svojstvo “relSecNotation” ima klasu “SecurityNotation” kao opseg, i svojstvo “reqCredential” ima klasu “Credential” kao opseg. Postoje razne instance za definisane klase. Na primer, instance “Syntax” klase su “ASCII, DAML+OIL, OWL, XML, MIME“, sigurnosne notacije su “Authentication, Authorization, AccessControl, DataIntegrity, Confidentiality, Privacy, ExposureControl, Anonymity, Negotiation, Policy, KeyDistribution“, i “X.509” je instanca “KeyFormat” klase. “XML-DSIG” je instanca “Signature” klase i “OpenPGP-Enc” je tip “Encryption” klase. Specifični protokoli kao što su “X-KRSS, X-KISS, Kerberos” ili “SOAP” su definisani kao instance odgovarajućih protokol podklasa koje zadovoljavaju određene restrikcije.

Klase za ograničenje su klase u kojima je ograničen opseg na jedno od svojstava objekta “reqCredential, syntax, relSecNotation”. Klase za ograničenje se koriste kao “šabloni” za određene sigurnosne oznake “Authentication”. Dakle, ako web servis ima “AuthenticationSubClass” kao jedan od svojih zahteva, onda to znači da servis zahteva od korisnika da se autentifikuje.

Moguće je definisanje drugih klasa za ograničenje. Na primer, moguće je definisanje klase gde je jedan od zahteva uverenja X.509 ili gde je korišćena sintaksa XML.

Korišćeno je definisanje podklasa koje karakterizuju protokole. Na primer, XKMS je definisan kao protokol koji je presek različitih klasa, uključujući klase za ograničenje za autentifikaciju, distribuciju ključeva i XML. Ovo znači da je XKMS protokol koji zahteva autentifikaciju, a takođe je i protokol za distribuciju ključeva koji koristi XML sintaksu. Izabrano je definisanje klasa za ograničenje da bi predstavili sigurnosne zahteve, zato što želimo da iskoristimo nasleđene karakteristike Description Logic (DL), na kojima se zasniva OWL. Konkretno, to omogućava korišćenje sveobuhvatnog rasuđivanja. Korišćenjem ove tehnike moguće je odlučiti da li su sigurnosne mogućnosti i zahtevi SWSs i agenata identični ili imaju

odgovarajući nasleđen odnos. Više detalja kako se koristi osnovna ontologija za označavanje servisa i agenata je dato u “Proširenje sigurnosnih servisa”.

8.6.3 ONTOLOGIJA “PROŠIRENJE SIGURNOSNIH SERVISA”

Sigurnosne ontologije “Uverenja” i “Sigurnosni mehanizmi” mogu da se koriste za obeležavanje sigurnosnih svojstva SWSs. Definisana je “SecurityMechanism” klasa kao podklasa “ServiceParameter”. Definisana su dva nova svojstva za web servise, “securityRequirement” i “securityCapability” tipa “SecurityMechanism” klase.

8.6.4 ONTOLOGIJA “PROŠIRENJA SIGURNOSNIH AGENATA”

Analogno proširenju sigurnosnih servisa definisana su svojstva “securityRequirement” i “securityCapability” sa opsegom “SecurityMechanism” klase kao proširenje sigurnosnih agenata.

8.6.5 ONTOLOGIJA PRIVATNOSTI

Zaštita lične privatnosti i privatnosti preduzeća u SWSs je veoma važna jer omogućava automatsku razmenu informacija kroz inteligentne agente bez kontrole ili sa veoma malo kontrole ljudi. Proces pristupa i delegiranja privatnih informacija bi trebalo strogo da se primenjuje bez obzira gde se informacije nalaze. Politika privatnosti obično definiše pravila ili koncepte za pristup i korišćenje privatnih informacija. Ova pravila i koncepti uključuju dozvole ili prava, i obavezu korišćenja privatnih informacija. Na primer, kompanija je dobila dozvolu pristupa klijentovim privatnim informacijama sa obavezom da ne daje informacije trećoj strani. Predloženo je da svaki agent i web servis imaju pridodate polise privatnosti koje se mogu razmeniti na početku interakcije, i za proveru da li agent ispunjava propisanu privatnost autoriteta servisa i obrnuto.

Razvijena je generička, jednostavna i laka za korišćenje ontologija za izražavanje polisa privatnosti, kao i protokol za podršku upoređivanja polisa privatnosti. Definisana je “Entity” klasa sa dva svojstva “hasResource” i “hasPolicy”. Polisa može da ima više “Rules”, kao što je prikazano sa “hasRule” svojstvom. Svaki “Rule” ima “Action”, kao što je definisano sa “onAction” svojstvom, i primenjuje se na “Resource”, kao što je navedeno sa “onResource” svojstvom. Ako nijedno “onResource” svojstvo nije navedeno, pravilo će se primenjivati na sve tipove resursa. Resurs se odnosi na informacije koje moraju biti zaštićene. Razlikuju se tri tipa pravila: Authorization, Capability i Obligation.

- **Authorization** precizira akcije koje dozvoljavaju ili zabranjuju drugima pristup resursima. “PositiveAuthorization” i “NegativeAuthorization” su dve podklase “Authorization” klase;
- **Capability** precizira akcije koje sam entitet može da obavlja u cilju zadovoljenja polise privatnosti drugog entiteta, ako je potrebno. “PositiveCapability” (šta entitet može da uradi) i “NegativeCapability” (šta entitet ne može da uradi) su dve podklase “Capability” klase. “PositiveIntentCapability” i “NegativeIntentCapability” su dve podklase “PositiveCapability” klase koji opisuju akcije koje entitet namerava da izvrši;
- **Obligation** precizira akcije koje će entitet sprovesti ili neće. “PositiveObligation” i “NegativeObligation” su dve podklase “Obligation” klase. Obaveze se obično tiču šifrovanog skladištenja podataka, ili potpisivanja/kriptovanja podataka prilikom prenosa. Bilo koja strana koja prihvata ili oglašava polisu je odgovorna za sprovođenje ovih obaveza. “NegativeObligation” precizira koja vrsta prikupljenih

podataka ne sme da bude šifrovana za sprovođenje ovih obaveza, ili podatke koji ne treba da budu potpisani ili šifrovani tokom prenosa podataka.

Neke od akcija klase su dodatno prerađene u podklase.

- “Local”, “ThirdParty” i “Forum” su podklase “Disclosure” akcione klase. One detaljno opisuju na koji način informacije mogu da budu otkrivene. “ThirdParty” klasa ima “Affiliation” i “NonAffiliation” podklase. “Affiliation” klasa se odnosi na ljude, kompanije i organizacije koje imaju partnerstvo sa određenim kompanijama ili organizacijama. “NonAffiliation” klasa se odnosi na javne entitete;
- “StorageEncryption” i “StorageDecryption” su podklase “Storage” akcione klase. Stranka može da zahteva da izabrani podaci budu skladišteni sa određenim kriptografskim svojstvima;
- “Send” i “Receive” su podklase “Transmission” klase. Za obe, “Send” i “Receive” klase su definisane podklase “TransmitEncryption”, “TransmitPlainText” i “TransmitSign”, tako da stranka može da odredi da li podaci koji se šalju ili primaju moraju da budu šifrovani ili potpisani ili ne moraju da imaju bilo kakav zahtev (primer plaintext).

8.6.6 ONTOLOGIJA KRIPTOGRAFSKOG OZNAČAVANJA INFORMACIJE OBJEKTA

Da bi se uhvatila šifrovana ili potpisana ulazna ili izlazna vrednost podataka servisa definisana je “InfObj” klasa (informacija objekta) i podklase “EncInfObj” (šifrovana informacija objekta) i “SigInfObj” (potpisana informacija objekta). “InfObj” klasa se koristi sa opseg ulaznih i izlaznih parametara OWL-S servisa. Informacije objekata imaju “baseObject” svojstvo koje opisuje tip ili strukturu informacije koja je šifrovana u njoj. Na primer, osnovni objekat i I/O parameter klase “EncInfObj” mogu da budu klasa kao SSN. Ovo svojstvo omogućava izvođenje znanja o vrsti razmenjenih podataka i može da se iskoristi za namenu kao što je razmišljanje da li parametar servisa odgovara zahtevima klijentovog I/O parametra. Osim toga, informacija objekta može da ima svojstvo “cryptoAlgUsed” koje se odnosi na specifičan kriptografski algoritam koji je korišćen za potpisivanje ili šifrovanje podataka.

8.6.7 KONTEKST-SADRŽAJ POVERENJE: ONTOLOŠKI PRISTUP

Generalno, kontekst informacije može da ima uticaj na aspekt poverenje-relevantnost pri komunikaciji. Kontekst informacije može da karakteriše situaciju bilo kog entiteta, da li je to osoba, mesto ili objekat, koji je relevantan za interakciju između korisnika i aplikacije.

9. POVERENJE

Na webu je dozvoljeno da bilo koji korisnik objavi informaciju, i da tu informaciju da na raspolaganje drugim korisnicima. Semantički web koristi ovaj sistem protokola kao osnovno sredstvo komunikacije, ali se sposobnost pretraživanja zasniva na povezanim, mašinski razumljivim podacima u dokumenatima. Jedna od vizija semantičkog weba je da će automatizovani agenti moći da pretražuju dostupne baze podataka i da obrađuju podatke na inteligentan način koristeći naprednije tehnike od onih koje se trenutno koriste u obradi podataka. Semantički web teži da korisnicima omogući efikasniji rad računara i aplikacija, obezbeđujući im informacije u odgovarajućem formatu.

Trenutno na webu, korisnik odlučuje da li je objavljena informacija verodostojna ili vredna poverenja. Ova odluka se kod kvalifikovanih i iskusnih korisnika donosi relativno lako. Za automatizovane agente, slepo poverenje u informacije koje se dobijaju sa weba može da dovede do nepreciznih i netačnih zaključaka. Korisnik agenta ne mora da dovodi u pitanje rezultate koje mu agent predstavlja [41], ili može teško da dijagnostikuje koji su podaci ili koraci rezonovanja izazvali agenta da prikaže netačne rezultate. Zbog toga je važno za agenta da bude u mogućnosti da proceni verodostojnost svih podataka koje je dobio sa weba kako bi utvrdio da li treba da koristi informaciju za dalju obradu ili ne. Odluka o tome da li je podatak verodostojan ili ne je teška za agenta jer nije u stanju da uzme u obzir kontekst informacije ili neku od heuristika koje korisnik koristi da ispita izvor i kvalitet predstavljene informacije.

Agenti će preovladavati na semantičkom webu. Njihova upotrebljivost će biti smanjena bez sredstava za automatsko ocenjivanje verodostojnosti informacija koje su prikupili. Agenti će morati da budu sposobni da odluče da li su podaci odgovarajući za dalju obradu. Ova sposobnost još uvek nije jasno definisana, i bitnija je od poverenja osobi ili sistemu koji je generisao podatke. Procena svakog dela podataka je nerealna. Praktično sredstvo za klasifikaciju je ocenjivanje autora (korisnika ili mašine) koji je objavio informaciju.

Poverenje je „verovanje“ da su podaci tačni, ili da ispunjavaju kriterijume korisnika.

9.1 LJUDSKO POVERENJE

Gambetta u “Can We Trust Trust”, na osnovu diskusija u prikupljenim radovima predstavlja pregled i daje zaključak zašto ljudi veruju jedni drugima i kako je to poverenje izgrađeno. Na osnovu toga predstavljena je potencijalna definicija poverenja [42]:

- Poverenje (ili simetrično nepoverenje) je određen nivo subjektivne verovatnoće sa kojom agent procenjuje da će drugi agent ili grupa agenata izvršiti određenu akciju, i pre nadgledanja te akcije i u kontekstu u kojem to utiče na njegovo izvršenje.

Ova definicija ima važne posledice:

- **Poverenje je subjektivno:** Abdul-Rahman i Hailes zaključuju da zbog subjektivnog poverenja, sistem poverenja baziran na “trusted authority” nije validan, iz razloga što je ono subjektivno za svakog agenta [5], [6];
- Poverenje je važno u akcijama koje ne mogu da se nadgledaju;
- “Nivo poverenja zavisi od toga kako akcije korisnika utiču na akcije agenta” [6]. Poverenje je relevantno samo u kontekstu u kojem utiče na poverljivost agenta, nivo (težina, trošak) na koji prekid poverenja utiče na agenta je važan;

- Poverenje je relevantno u situacijama gde su faktori nepoznavanje ili neizvesnost. Ako je poznat ishod akcije pre izvršenja, poverenje će biti apsolutno i implicitno. Neizvesnost ishoda je važan faktor uslova za poverenje. Iako nivo poverenja može da se izmeri raznim kriterijumima, neizvesnost i dalje postoji.

Fogg i Tseng predstavljaju različite poglede na poverenje, gde tvrde da “poverenje označava pozitivno uverenje o opažanju pouzdanosti, zavisnosti i uverenja u osobu, objekat ili proces” [41].

9.2 ZAŠTO LJUDI VERUJU

Gambetta iznosi brojne argumente da li je saradnja između ljudi poželjna ili potrebna (posebno među članovima zajednice), sa zaključkom da postoje situacije u kojima je potrebna i u kojima je potencijalno štetna. Takmičenje između ljudi može da ima efekat poboljšanja karakteristika individue ili grupe, i može da služi kao važna lična motivacija. Potpuno odsustvo saradnje može da ima štetni efekat na pojedinca ili grupu. Ako se zamisli na primer zajednica „pljačkaša i ubica“, iz perspektive člana te zajednice, neka saradnja je važna kako bi se osiguralo samo-očuvanje. Zajednica ne bi postojala bez te saradnje. Kombinacija takmičenja i saradnje dovodi do optimalnog rešenja za individue i grupe [42].

Dilema zatvorenika je klasičan problem u teoriji igara, gde je izbor dva uhapšena zatvorenika da li će da sarađuju štetan ili ne. Ako jedan zatvorenik odbija a drugi pristaje na saradnju, onaj koji odbija saradnju dobija najveću, dok drugi dobija najmanju kaznu. Ako obojica sarađuju, dobijaju prosečnu kaznu. Ako obojica odbiju saradnju dobiće najmanju kaznu.

Axelrod u „Razvoj kooperativnosti“ predstavlja ponavljanje dileme zatvorenika, varijacija standardne dileme zatvorenika gde se igra ponavlja [43]. Saradnja na dugoročnom planu donosi nagradu. Ova nagrada nije najveća ako i druga strana sarađuje. Zatvorenik mora da veruje da će i drugi zatvorenik da deluje u interesu oba zatvorenika, obojica moraju uvek da sarađuju. Ovo poverenje nije uvek jednosmerno. Zatvorenik A mora da veruje da zatvorenik B radi u interesu zatvorenika A. Zatvorenik B mora da veruje da zatvorenik A radi u interesu zatvorenika B. Recipročno ili zajedničko poverenje je važno u izgradnji saradnje.

U realnim interakcijama nije moguće predviđanje svih akcija druge strane, tako da nije moguća kompletna zaštita. Iz tog razloga je potreban određen nivo poverenja u interakciji sa drugima. Realne interakcije sadrže obaveze sa mogućnošću nadoknade (ili troškove) za greške. Ovo se manifestuje postojanjem ugovora koji nude nadoknade.

9.3 KRITERIJUMI ZA MERENJE POVERENJA

Poverenje je subjektivno i određuje se različito od strane ljudi. Gambetta zapaža da je znanje o ljudskoj psihologiji limitirano, i da nije moguće razumeti sve faktore koji utiču na uverenja osobe. Faktor koji utiče na poverenje jedne osobe prema drugoj osobi, ne može u potpunosti da se razume.

Priroda poverenja može da se razume u opštim uslovima. Gambetta zapaža da je poverenje (ili nepoverenje) rezultat saradnje, ili preduslov saradnje. Poverenje postoji između vere i poverljivosti. Verovanje kao uverenje se ne zasniva na prethodnom iskustvu saradnje. Ako je početna saradnja probna, bazirana na uverenju, ona postaje sigurnija jer poverenje u drugu

stranu raste kroz iskustvo.

Ljudi su takođe skloni “slepom poverenju”, ili verovanju (ili neverovanju) u osobu na nerazumno velikom stepenu, uprkos dokazima koji sugerišu suprotno.

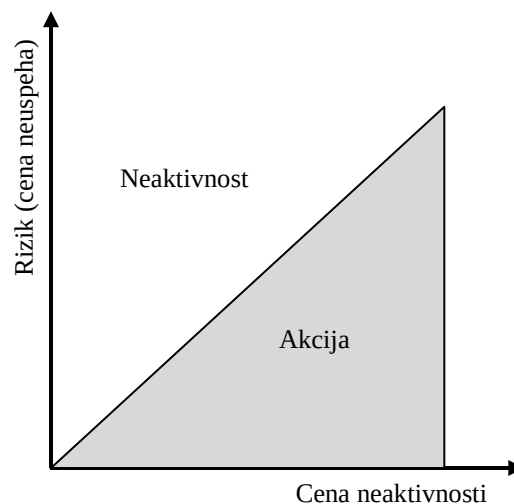
Procena troškova uključenih u nedostatak za “stranu od poverenja” može da omogući neke dokaze verovatnoće da li će ili ne stranka zahtevati izvođenje akcije. Ako “strana od poverenja” veštački nametne trošak stranki, to se naziva prinuda. Ako postoji prinuda, poverenje nije validno jer stranka ne deluje kompletno slobodnom voljom.

9.4 MODELOVANJE POVERENJA

Nivo poverenja može da se navede kao verovatnoća (p), gde verovatnoća može da se opiše kao uverenje da će agent uspešno i ispravno izvršiti određenu akciju. Pitanje je u kom trenutku može biti doneta odluka da li je neki subjekat dovoljno pouzdan da izvrši akciju. Ovo zavisi od procene nedostataka subjekta. Ako je procena niska, moguća je akcija i bez poverenja. Ako je procena visoka, potrebno je prisustvo velikog poverenja pre verovanja subjektu.

Moguće je postojanje procene da se ne izvrši akcija zbog nedostatka poverenja. Gambetta predstavlja primer izbora restorana za jelo. Restorani treba da se izaberu na osnovu mogućnosti da pripreme kvalitetnu hranu koja je hranljiva i ne postoji mogućnost izazivanja bolesti. Izborom restorana, gde su svi nepoverljivi, cena neodabira restorana može da bude veća od rizika odabira najpoverljivijeg restorana na osnovu doživljaja. Sledi, cena neaktivnosti nadmašuje uočeni rizik akcije.

Ako je subjekat nov, i ne postoji dostupan kriterijum na osnovu koga može da se oceni, teško je da se dođe do vrednosti p . Da li bi trebalo da bude 0.5, i na taj način označi 50% šanse očekivanog izvršenja, ili treba da bude 0 ili blizu 0, označavajući da subjekat ima malu šansu da se ponaša očekivano. Njegove akcije ne mogu da se predvide tako da bi vrednost p trebalo da bude blizu 0.5. Ako se posmatra prethodno ponašanje subjekta, visoka vrednost poverenja bi bila netačna jer subjekat može u 80% slučajeva da se ponaša korektno, dok u preostalih 20% može da se ponaša zlonamerno. Dodeljivanje numeričke vrednosti za poverenje je teško i vodi do komplikacija u modelovanju poverenja.



Sl. 50. Preduzimanje akcije (poverenje) kao rizika u fukciji cena neaktivnosti

9.5 POVERENJE, KREDIBILITET, POŠTENJE, POUZDANOST

Ljudi koji veruju osobi daju informacije na osnovu doživljaja pouzdanosti. Ovo opažanje može da bude bazirano na mnogo faktora kao što su: doživljaj kredibiliteta, poštenja i pouzdanosti. Mnoge od njih su ljudske osobine, opisane kao osobine “entiteti strasti”. Osobina kao što je pouzdanost može da se pripiše podacima ili računarskom sistemu.

Kredibilitet se definiše kao “reputacija koja utiče na sposobnost kojoj se veruje” [44], i na taj način je kvalitet koji može da se dodeli agentu ili podacima. Stranice sa lošom prezentacijom su često diskreditovane od strane korisnika koji surfuju webom, jer prezentacija daje korisniku osećaj da i informacija može biti lošeg kvaliteta [45]. Ovi faktori utiču na kredibilitet web strane. Kredibilitet je faktor koji doprinosi vrednosti verovatnoće p , opažen od strane koja ima poverenje. Poverenje nije svojstvo cilja, već uverenje koje sadrži strana koja ima poverenje. Kredibilitet je ogledalo poverenja – to je skup svojstava koji mogu da dovedu do poverenja. Kredibilitet je sličan verovanju [41].

“Doživljaj poštenja” je uverenje da agent nije zlonameran, i da će postupiti na namenjen način, i da neće namerno odstupiti od određene putanje. Ovo odstupanje može da dovede do smanjenja vrednosti p , ili može da se izrazi kao neodređeno odstupanje. Obeležićemo ovo odstupanje sa d .

Pouzdanost je slična poštenju, ali nije posledica zlobe. Nedostatak pouzdanosti je nemogućnost preciznog ispunjenja očekivanja. Pouzdanost doprinosi vrednosti p , ali takođe doprinosi i odstupanju d . Agent koji je pošten i kredibilan će dobiti veliku vrednost za p , a malu vrednost za d . Ako je agent nepouzdan, vrednost p će se smanjivati sa svakom greškom, a vrednost d će se povećavati. Fogg i Tseng definišu poverenje kao kombinaciju doživljaja pouzdanosti, zavisnosti i uverenja [41].

Vrednosti p i d su intuitivno određene. Ako je vrednost poverenja dobijena iz posmatranja, onda je vrednost za p 0.5. Posmatranje akcija može da dovede do tačne vrednosti 1 ili netačne 0 u vrednovanju poverenja. Korišćenje mere odstupanja je besmisleno ako su rezultati ravnomerni. U ovom slučaju standardno odstupanje će uvek biti 0.5.

9.6 SVOJSTVA POVERENJA

Kada se modeluje veza poverenja, treba da postoji svest o njegovim svojstvima. Veza poverenja ima: određeni nivo tranzitivnosti, kompozitnost, lična i asimetrična.

Tranzitivnost - Koncept tranzitivnosti poverenja je prirodan. Ako A veruje B , onda A treba da veruje u preporuke B o drugima. Golbeck tvrdi da osoba treba da ima dva uverenja o drugoj osobi. Poverenje u osobu, i poverenje o njenoj sposobnosti da preporučuje [46]. Na osnovu drugih radova koji sugerišu da se poverenje razlikuje u zavisnosti od oblasti znanja ovo je pogrešna dihotomija. Ako je sve grupisano u globalnu grupu poverenja, koje svojstvo preporuke zaslužuje svoju zasebnu kategoriju? Primena kod ovog problema, i rešenje između jednostavnosti jednog (ili nekoliko) ocenjivanja i preciznosti dobijene od velikog broja ocenjivanja zahtevaju ovu podelu. Golbeck tvrdi da ocenjivanje poverenja može da se predstavi jednim ocenjivanjem kako bi se pojednostavila interakcija korisnika. Golbeck takođe tvrdi da poverenje ne prikazuje “savršenu tranzitivnost”, već količinu poverenja širi duž lanca koji treba da se smanji.

Kompozitnost - Golbeck tvrdi da kada se dobije broj preporuka o pouzdanosti osobe, vrednost poverenja u tim preporukama bi trebalo da se sastavi u jedno uverenje o pouzdanosti

osobe.

Personalizacija - Važno svojstvo poverenja je lično mišljenje ili pogled. Iz ovog razloga ocenjivanje na mreži poverenja retko može biti dosledno.

Asimetrija - Nije neophodno da B veruje A , ako A veruje B . Poverenje B u A je nezavisno (u odnosu na motivaciju B prema A) od poverenja A u B .

9.7 KAKO LJUDI OCENJUJU KREDIBILITET WEB SAJTA

Doživljaj kredibiliteta informacije ili autora te informacije je važan faktor ocenjivanja pouzdanosti informacije (ili autora). U literaturi se predlaže da ljudi mogu da kažu jednu stvar o ocenjivanju kredibiliteta web sajta, a da akcije koje se izvode budu različite. Najvažniji faktor koji se koristi u ocenjivanju kredibiliteta web sajta je doživljaj kvaliteta ili profesionalnosti skice i dizajna [45], [47]. Pretpostavlja se da korisnici ocenjuju napor uloženi u prikupljanje i obradu sadržaja web sajta, na osnovu napora uloženog u druge oblasti a naročito u dizajn.

Fogg iznosi brojne sugestije za poboljšanje kredibiliteta web sajta, na osnovu rezultata studija kako ljudi ocenjuju kredibilitet web sajta. Ove sugestije su [48]:

- Olakšati proveru tačnosti informacije na sajtu;
- Prikazati da postoji realna organizacija iza sajta;
- Istaknuti stručnost organizacije u sadržaju i uslugama koje pružaju;
- Prikazati da pošteni i poverljivi ljudi stoje iza web sajta;
- Jednostavna mogućnost kontakta;
- Profesionalni izgled sajta (ili je odgovarajući za datu svrhu);
- Jednostavnost korišćenja i korisnost;
- Često osvežavanje sadržaja (ili prikaz redovnosti pregleda);
- Uzdržavanje od bilo kakvih promotivnih sadržaja (reklama, ponuda);
- Izbegavanje svih tipova grešaka, bez obzira na njihovu veličinu.

	2H2010	1H2010	2H2009	1H2009	2H2008
Phishing domain names	42,624	28,646	28,775	30,131	30,454
Attacks	67,677	48,244	126,697	55,698	56,959
TLDS used	183	177	173	171	170
IP-based phish (unique IPs)	2,318	2,018	2,031	3,563	2,809
Maliciously registrated domains	11,769	4,755	6,372	4,382	5,591
IDN domains	10	10	12	13	10

Tabela 10. Osnovna statistika – izveštaj Anti-Phishing Working Group [86]

Ove sugestije ne pominju faktore koje korisnici koriste prilikom odlučivanja da li je web sajt koji koriste, onaj koji očekuju. Sa povećanim brojem “phishing” napada kroz e-mail i web,

edukacija dovodi korisnike do veće svesti nedoslednosti u web stranicama, kao što su nepravilna imena domena ili nepodudaranje HTTPS sertifikata. Drugi faktori koji privlače pažnju korisnika su neuobičajena praksa ili elementi na web stranici koja je inače poznata. Ovi faktori utiču na poverenje korisnika kod autentičnosti ili tačnosti web stranice.

9.8 POVERENJE U DISTRIBUIRANIM SISTEMIMA

Abdul-Rahman i Hailes raspravljaju da je poverenje subjektivna mera i da zato mora da bude na neki način distribuirano [49]. Nijedan centralni “Trusted Authority” ne može da omogući perspektivu istovremeno podudarnu sa perspektivom svih korisnika, tako da poverenje mora da bude distribuirano. Ovo ne znači da treba da bude deo distribuiranog sistema, nego da merenje poverenja treba da se računa za svaku jedinstvenu perspektivu sistema.

Josang razmatra značenje poverenja sa mogućnošću primene na distribuirane sisteme. Fokus rada je na sigurnosti, ali je ideja primenjiva i na semantički web [50]. Jedan od važnih uvida koji je predstavljen je razdvajanje “strastvenih” i “racionalnih” entiteta. Rad opisuje kako se različito ocenjuje pouzdanost ovih entiteta, kao i poverenje kao pozitivan koncept, odnosno uverenje da entitet nije zlonameran. Ova rečenica ilustruje dva njegova argumenta. Prvi, poverenje nije svojstvo entiteta, već uverenje sadržano u drugom entitetu o tom entitetu. Drugo, činjenica da je poverenje uverenje znači da samo strastveni entiteti mogu da ocenjuju pouzdanost drugih entiteta.

Strastveni entiti su opisani kao bilo koji entitet koji je osoba, ili uključuje ljudsku interakciju. Oni se razlikuju od racionalnih entiteta jer na njih utiče veliki broj faktora, i nisu predvidivi. Entitet ne može da predvidi svoje ponašanje. Ova slobodna volja entiteta može da se definiše kao: poverenje kod strastvenog entiteta je uverenje da će njegovo ponašanje biti bez zlonamerne svrhe.

Racionalni entiteti se opisuju kao entiteti koji nisu strastveni – entiteti koji nemaju slobodnu volju. Entiteti ne mogu da donose odluke na svoju ruku, jer su odluke unapred definisane u bilo kojim okolnostima. Uverenje u ispravnost sistema je uverenje da mogu da odole manipulaciji koja može da promeni to ponašanje. Poverenje kod racionalnog entiteta je uverenje da se može odoleti zlonamernoj manipulaciji strastvenog entiteta.

Josang dalje tvrdi da samo strastveni entiteti mogu da ocenjuju poverenje. Ovo je bazirano na tvrdnji da se poverenje računa iz velikog broja izvora od kojih mnogima mašine ne mogu da pristupe ili nemaju pristup. Na primer, poverenje može da bude bazirano na iskustvu, znanju “poverljivog” entiteta, preporuci ili uverenju. Jednostavno, “strastveno ponašanje može biti procenjeno od strane drugih strastvenih entiteta”.

Abdul-Rahman i Hailes definišu tranzitivnost u mrežama poverenja u referenci [51]. Josang podrazumeva da automatizovani sistem širenja poverenja nije moguć zbog tvrdnje da samo strastveni entiteti mogu da ocenjuju pouzdanost. Abdul-Rahman i Hailes dozvoljavaju kondicionalnu tranzitivnost ako su sledeći uslovi ispunjeni:

- B eksplicitno informiše A o svom poverenju u C , kao “preporuka”;
- A veruje B preporuci, odnosno preporučeno poverenje postoji u sistemu;
- A ima mogućnost da oceni “kvalitet” B preporuke;
- Poverenje nije apsolutno, odnosno A može manje da veruje C od B , na osnovu

njegove preporuke.

Model poverenja baziran na ovim uslovima sadrži preporuke:

- Decentralizacija daje mogućnost svakom agentu da ostvari svoje pravo da rukovodi svojim poverenjem;
- Generalizacija poverenja: da se formira skladan model, ideja poverenja je generalizovana i svedena na kategoriju i vrednost poverenja;
- Eksplicitna izjava poverenja: ovo smanjuje dvosmislenost i analogno je preporukama sa zasebnim kategorijama;
- Protokol preporuka: specifičan protokol za razmenu informacija o poverenju olakšava razmenu informacija vezanih za poverenje.

9.9 POVERENJE U SISTEMU PREPORUKA

Sistemi preporuka su popularni, posebno kod online prodavnica, za preporuku koji bi proizvodi mogli da budu interesantni ili odgovarajući za korisnike [52]. Mnogi korisnici su upoznati sa preporukama koje Amazon.com obezbeđuje zasnovane na kriterijumima kao što su istorija pregleda i naručivanja. Dva osnovna tipa sistema preporuka preovlađuju: bazirani na sadržaju i sistem kolaborativnog filtriranja.

Sistemi bazirani na sadržaju, stavke (i sa njima povezani meta podaci) se upoređuju sa korisnikovim profilom kako bi pronašli stavke koje odgovaraju oblasti koju je korisnik precizirao ili stavke koje je korisnik ranije kupio. Najveći problem ovih sistema je tačno označavanje velikog broja stavki. Ovaj zadatak zahteva centralizovani sistem, jer ne postoji jednostavan način za rešavanje protivrečnosti, kao što je koji od opisa dva korisnika za isti sadržaj treba da se prihvati.

Kolaborativno filtriranje koristi sugestije upoređujući ocenjivanje stavki od strane korisnika sa drugim ocenjivanjem korisnika za istu stavku [53]. Na taj način se pomaže pronalaženje korisnika koji su “slični”. Sistem je onda u mogućnosti da preporuči stavku koju su “slični” korisnici ocenili visoko, onim korisnicima koji nisu ocenili tu stavku.

Ove tehnike nisu limitirane na online prodavnice. Druge uobičajene aplikacije su servisi za vesti i blog koji nude filtriranje na osnovu onoga šta drugi korisnici čitaju i ocenjuju visokom ocenom. Bloglines i last.fm su primeri ovoga. Bloglines [54] omogućava preporuke na osnovu korisnika koji imaju pretplate na slične RSS, dok last.fm [55] pruža preporuke na osnovu korisnika koji imaju sličan ukus za muziku. Ove aplikacije imaju za cilj da pomognu u prevazilaženju preopterećenja informacijama. Kolaborativno filtriranje je predmet tri slabosti identifikovanih kao [56]:

- **Razređeni podaci** – Ogroman broj stavki potencijalno postoji u sistemu koji koristi sistem preporuka, a korisnici pregledaju samo mali broj. Verovatnoća da se preklape pregledi dva korisnika je mala. Vrednost razređenih podataka može da se posmatra kao broj ćelija koje su prazne kada se kreira grid svih korisnika u odnosu na stavke koje su pregledane. Podaci koji su obrađeni u [56], [57] se sastoje od 42,920 korisnika, koji su ocenili 139,738 i pregledali 664,824 stavki. Razređenost ovih podataka je 99.99%. Ovo je dovelo do problema sa računanjem, računanje korelacije između svih korisnika, iako ne postoji preklapanje;

- **Hladan start** – Slično kao i kod problema razređenih podataka, novi korisnici koji imaju nizak broj ocena će dobiti loše preporuke od sistema preporuka na osnovu kolaborativnog filtriranja, jer će podudaranja biti loše generisana;
- **Jednostavno manipulisanje** – Sistemi kolaborativnog filtriranja su otvoreni za manipulisanje od strane zlonamernih korisnika, na primer „copy-profile“ napad [20]. Napadaču je potreban dupliran korisnikov profil ocenjivanja, i onda dodaje stavku ili stavke koje želi da budu preporučene. Ovakav napad je moguć ako je profil ocenjivanja javno dostupan, i ako je moguće jednostavno kreiranje novih naloga.

Sve ove slabosti mogu da se ograniče kroz dodavanje mehanizma poverljivog filtriranja. Umesto da novi korisnici moraju da pregledaju mnogo stavki kako bi izbegli problem hladnog starta, dovoljno je da odrede nekoliko poverljivih prijatelja sa sličnim ukusom i da se njihovi profili koriste kao osnova za preporuke. U ovom slučaju treba da se razmotre pitanja privatnosti, jer je korisnik u mogućnosti da vidi koje stavke su visoko ocenje od strane prijatelja. Rešenje problema manipulacije je korišćenje metrike poverenja “otporan na napad”.

O'Donovan i Smyth su u svom radu implementirali sistem preporuka na MovieLens skupu podataka [58]. Proizvođači (oni koji daju preporuke) su dobili profil nivoa ocenjivanja poverenja koji ocenjuje koliko puta je data preporuka bila tačna (u okviru greške određenog praga tolerancije) za sve potrošače (oni koji dobijaju preporuke) i sve stavke, i stavka-nivo ocenjivanje poverenja, koji ocenjuje koliko često su proizvođači dali tačnu preporuku za određenu stavku (svim korisnicima). Za $CorrectSet(p)$ kao skup svih tačnih preporuka datih od proizvođača p , i $RecSet(p)$ kao skup svih preporuka datih od proizvođača p , profil nivoa poverenja $Trust^p$ je dat kao:

$$Trust^p(p) = \frac{|CorrectSet(p)|}{|RecSet(p)|} \quad (8)$$

Stavka nivo ocenjivanja poverenja $Trust^I$ za proizvođača p i stavku i , i za sve potrošače stavka n-torki (c_k, i_k) u skupu tačnih preporuka i svih preporuka su dati kao:

$$Trust^I(p, i) = \frac{|\{(c_k, i_k) \in CorrectSet(p) : i_k = i\}|}{|\{(c_k, i_k) \in RecSet(p) : i_k = i\}|} \quad (9)$$

O'Donovan i Smyth koriste vrednost poverenja i kao težinu da promene standardne vrednosti preporuka, i kao filter da zabrane profilima koji nisu ocenjeni visokom ocenom da doprinose sistemu preporuka. Korišćene su strategije koje kombinuju ova dva. Merili su rezultate u odnosu na standardnu Resnickovu formulu preporuka [52] koja meri efikasnost poverenja za unapređenje preporuka.

Filtriranjem profila koji imaju nisku ocenu poverenja, eksperiment je pokazao poboljšanja u odnosu na standardnu Resnickovu formulu preporuka, za 69.8%. Ove preporuke nisu bile bolje, pokazujući samo 3% poboljšanja u tačnosti, sa srednjom apsolutnom greškom od 0.84. Korišćenjem kombinovane strategije filtriranja i težine na stavka-nivou poverenja, broj preporuka u odnosu na standardnu Resnickovu formulu preporuka je 66.8%, a preporuke su veće tačnosti za 22%, sa srednjom apsolutnom greškom od 0.68.

9.9.1 PRIMENJIVOST NA SEMANTIČKI WEB

Metrika poverenja zahteva ocenjivanje sličnih korisnika kako bi bila efikasna. Korisnici sa sličnim skupovima ocena se koriste kako bi se dobile ocene o drugim produktima, sa filtriranjem i težinom na osnovu prethodne tačnosti. Canny predlaže da takvo vezivanje korisnika

nije jedino poželjno sredstvo za širenje preporuka [59]. Korisnici često žele ocene eksperata u određenim oblastima. Oni se onda okreću raznorodnim korisnicima i dešava se distribucija različitog znanja.

Na semantičkom webu se poverenje ne koristi samo za preporuke, već i za generalno korišćenje informacija. Ocene na osnovu tačnosti nisu uvek moguće, a samim tim i računanje poverenja neće biti moguće. Korisnici mogu da žele da prebrišu vrednost poverenja na osnovu realnih interakcija i predrasuda. Poželjna je neka vrsta širenja poverenja, gde mišljenje eksperata može da postane vredno raznorodnim korisnicima ili korisnicima bez iskustva u interakcijama sa stručnjacima.

9.10 ŠIRENJE POVERENJA: MODEL POVERENJA I METRIKA

Nije realno da svaki agent na velikoj društvenoj mreži proceni i da tvrdnje o svim drugim agentima. Procena drugih agenata je skup proces u uslovima kao što su vreme, propusni opseg, kompjuterska moć, itd. U odeljku 9.1 opisano je poverenje kao subjektivna mera između vere i poverljivosti. Inicijalne interakcije su privremene i na osnovu nivoa poverenja koje je stečeno iz postojećeg znanja. Metrika poverenja nastoji da oponaša mogućnost pronalaska početnog nivoa poverenja. Metrika poverenja treba tačno da opiše pouzdanost agenta za određenu akciju gde ne postoje prethodne interakcije na postojećem nivou.

Metrika poverenja traži da se tačno predvidi nivo poverenja agenta u drugog agenta. Ceo sistem poverenja (metrika, protokoli) mora da bude otporan na manipulacije ovih nivoa poverenja od strane zlonamernih strana.

9.10.1 PKI/X.509

X.509 je dizajniran kao sredstvo za pružanje autentifikacije i kontrolu pristupa za unose u direktorijume vezivanjem javnih ključeva za nazive direktorijuma [60]. Prvobitno je objavljen 1988. i doživeo je tri revizije. IETF je proizvela X.509 profil dizajniran za operacije na Internetu, poznatiji kao PKIX, u cilju smanjenja opštosti ITU-T standard, i omogućila X.509 korišćenje na Internetu.

X.509 Public Key Infrastructure (PKI) je baziran na ideji poverenja Certification Authority (CA), koje izdaje digitalno potpisane sertifikate koji sadrže ime subjekta i javni ključ. Sertifikati mogu da budu lančani, tako da je moguća konstrukcija hijerarhije (struktura stabla), a CA treba da se posmatra kao “oslonac poverenja”. Da bi proverio validnost sertifikata, korisnik treba da proverio izdavača sertifikata, ako je potrebno i rekurzivno sve dok ne pronađe poverljivog CA izdavača.

Korisnici su upoznati sa X.509 sertifikatima, jer su oni osnova HTTPS (HTTP protocol over TLS (Transport Layer Security)) protokola. CA sertifikati se distribuiraju sa pretraživačima (kao što su Internet Explorer ili Mozilla Firefox), i na taj način korisnik veruje web sajtu. Korisnici mogu da dodaju CA sertifikat kome veruju. Proizvođači pretraživača veruju CA sertifikatima koje uključuju u paket kako bi uključili detalje koje su verifikovali kao tačne u sertifikatima potpisanim od njihove strane. Nepouzdan CA može da izda sertifikate koji potvrđuju pogrešno vezivanje između imena (domena) i javnog ključa, dozvoljavajući lažnu autentifikaciju a samim tim ne predstavljaju upozerenje korisniku.

Dva pristupa za obezbeđivanje pouzdanog CA se preporučuju [61]:

- CA mogu da kreiraju samo-potpisane sertifikate: sertifikati koji imaju istog izdavača

kao subjekat;

- CA mogu da unakrsno-potpisu sertifikate čime postaju oslonci međusobnog poverenja.

Jedina validacija X.509 je omogućena vezom između imena i javnog ključa, tako je poverenje limitirano. Ako je zadatak CA da samo potpiše sertifikat “pouzdan” institucije, zadatak može da postane opterećujuć i verovatno će dovesti do sertifikata koji je samo potpisan za stranke koje imaju tačku gledišta koja se slaže sa CA. Iz ovog razloga globalna metrika poverenja nije poželjna.

9.10.2 PGP WEB POVERENJA

Pretty Good Privacy (PGP) je softver koji je dizajnirao Zimmerman kako bi omogućio sigurnosnu razmenu fajlova i poruka [62]. Prihvaćen je i kao sredstvo koje omogućava poverljivost, integritet, autentikaciju i neporecivost (cilj digitalnog potpisa) za nižu cenu u odnosu na PKI.

Slično kao i kod X.509, validnost vezivanja ime/ključ je sertifikovana digitalnim potpisom korisnika, gde ime obično sadrži korisnikovu e-mail adresu. Razlika u odnosu na X.509 je ta da ne postoji centralni CA, i da zavisi od prethodno poznatih “poznanika” da potvrdi ključ prethodnih nepoznatih korisnika.

PGP proširuje X.509 dozvoljavajući korisniku da precizira četiri različita nivoa poverenja prilikom potpisivanja ključa. Na taj način korisnikov softver može da izračuna “rezultat” javnog ključa sa kojim nije upoznat.

9.10.3 DISTRIBUIRANI MODEL POVERENJA

Abdul-Rahman i Hailes u svom radu objašnjavaju značenje poverenja i daju njegovu definiciju [49]. Ostali doprinosi rada su metrika poverenja koja razdvaja vrednosti poverenja preporuka i generalne vrednosti poverenja. Definišu da poverenje postoji u različitim domenima. Metrika poverenja koristi skalu od -1 do 4, kao što je prikazano u tabeli 11. Za ocenjivanje pouzdanosti preporuka se koristi ista skala sa vrednostima od 1 do 4 označavajući bliskost preporuka i agentovo ocenjivanje pouzdanosti treće strane. Vrednost poverenja za T kroz putanju p je data kao:

$$tv_p(T) = \frac{tv(R1)}{4} \times \frac{tv(R2)}{4} \times \dots \times \frac{tv(Rn)}{4} \times rtv(T) \quad (10)$$

Gde je:

$tv(Ri)$ vrednost poverenja preporuka od Ri

Value	Meaning	Description
-1	Distrust	Completely untrustworthy
0	Ignorance	Cannot make trust-related judgement
1	Minimal	Lowest possible trust
2	Average	Mean trustworthiness. Most entities have this trust level
3	Good	More trustworthy than most entities
4	Complete	Completely trust this entity

Tabela 11. Semantika vrednosti poverenja [17]

$rtv(T)$ vrednost preporučenog poverenja T (od poslednje preporuke u lancu)

$tv_p(T)$ vrednost poverenja T kroz putanju p

U slučaju da distribuiranim entitetima nije dozvoljeno da budu u lancu, deljenjem vrednosti poverenja svake preporuke u lancu sa 4, vrednost poverenja se mapira u raspon $[0...1]$. Ovo ima smisla kao lanac sa dve „nepoverljive“ preporuke koje sa ovom formulom kreiraju pozitivnu vrednost. Autori rada priznaju da je formula proizvoljna, ali da demonstrira važan koncept koji preovladava u metrici poverenja: što duži lanac poverenja, to će niže poverenje da bude.

U prethodnom radu se takođe diskutuje o protokolu za širenje poverenja, uključujući i sistem pretplate i obaveštenja za oduzimanje i osvežavanje preporuka. Prilagođavajući protokol tako da koristi samo zahteve, protokol bi mogao biti odgovarajući za web, i na taj način koristan za semantički web. Protokol treba da sadrži neku formu kontrole pristupa. U delu 9.8, kontrola pristupa je neophodna za prevenciju igranja sa sistemom.

9.10.4 ADVOGATO

Advogato metrika poverenja je stekla veliku pažnju kao napad-otporna metrika poverenja [63, 64, 65]. Metrika je slična PageRank algoritmu [20], kao kapacitet ograničenog protoka mreže. Ideja metrike je da dobri čvorovi retko sertifikuju loše čvorove, a loši čvorovi mogu da sertifikuju jedni druge. Poverenje neće da teče od loše do dobre sekcije mreže. Advogato metrika se često vidi kao globalna metrika poverenja sa implikacijom da ne prepoznaje poverenje kao subjektivnu meru. Metrika može da se implementira od strane svakog klijenta i na taj način koristi kao lokalna metrika poverenja.

9.11 POVERENJE NA SEMANTIČKOM WEBU

Web je distribuirani prostor informacija, gde svako može da objavi bilo koju informaciju. Semantički web ne menja protokole (HTTP) ili osnovu weba, URI-a. Korisnici ocenjuju kredibilitet informacija na webu na različite načine, uključujući vizuelne dokaze kao što je dizajn web sajta. Na semantičkom webu informacije su podaci sa jednostavnom formom koji su dizajnirani za obradu od strane računara, što znači da mnoge ocene kredibiliteta weba nisu primenjive na semantički web. Na semantičkom webu pouzdanost podataka treba da se ocenjuje pre obrade, tako da rezultat koji se predstavlja korisniku bude precizan. Bez razumnog uverenja da će sistem predstaviti precizne rezultate, preuzimanje tehnologija baziranih na semantičkom webu neće biti veliko.

Golbeck je uradio brojna istraživanja koja se odnose na semantički web [46], [66], [67], [68], [1]. Njegov najveći doprinos su dve metrike poverenja koje su testirane i izmerene [30], i RDF rečnik za opisivanje veze poverenja na semantičkom webu [5].

Videli smo u odeljku 9.8 da poverenje mora da bude distribuirano. Svaki agent u sistemu ima jedinstven pogled na pouzdanost drugih agenata. Golbeck metrika poverenja je distribuirana u ovom smislu, kao i u smislu da vrednost poverenja može biti distribuirana i objavljena kao bilo koji drugi dokument na webu.

9.11.1 ONTOLOGIJA POVERENJA

Ontologija poverenja je dizajnirana da proširi FOAF ontologiju i da dozvoli korisniku da opiše koliko veruje drugim korisnicima uopšteno ili u određenim oblastima.

@prefix trust: <http://trust.mindswap.org/ont/trust.owl		
:a	rdf:type	foaf:Person .

:a	trust:trustsRegarding	:b .
:b	trust:trustSubject	<http://trust.mindswap.org/> .
:b	trust:trustedPerson	:c .
:c	rdf:type	foaf:Person .
:c	foaf:name	"Jen Golbeck" .
:b	trust:trustValue	10 .
:a	trust:trust10	:d .
:d	rdf:type	foaf:Person .
:d	foaf:name	"Barry Irwin"

Ontologija ocenjuje poverenja na skali od 1 do 10, bez eksplicitnog nepoverenja, i dozvoljava opcioni opis poverenja u određenom domenu znanja. Prethodno su data oba primera.

Date su dve izjave o poverenju. Prva koristi trustsRegarding formu da izjavi da je foaf:Person sa imenom "Jen Golbeck" poverljiv u pogledu trustSubject koji ima URI <http://trust.mindswap.org/>, sa trustValue 10. Druga forma izjave poverenja pokazuje da prva osoba ima trust10 svojstvo, pri čemu je vrednost foaf:Person "Barry Irwin". Ova izjava znači da prva osoba ima poverenje u drugu osobu u svim domenima znanja sa vrednošću 10.

Eksplicitnom izjavom poverenja koristeći semantičke web tehnologije, ontologija poverenja dozvoljava opis poverenja. Dostupnim korisnim opisom vrednosti poverenja, podaci mogu da se prikupljaju i koriste od strane servisa koji će da zaključi ocenu poverenja nepoznatih korisnika.

9.11.2 POVERLJIV SEMANTIČKI WEB

Kredibilitet web sajtova je postala briga među onih koji veruju da je web moćno sredstvo, što je dovelo do formiranja organizacija kao što su Consumer Reports WebWatch, koja "želi da poboljša kredibilitet sadržaja na webu".

Deo problema je da korisnici često ne mogu da imaju poverenja u sadržaj koji pronađu online [41], [47].

Ako semantički web prevaziđe ovaj problem, sistem će se posmatrati kao pouzdan. Sistem mora da bude otporan na uobičajene napade i probleme kao što su phishing i spam. Slika 31 prikazuje tehnologije potrebne za realizaciju semantičkog weba. Jedino implementacijom svih tehnologija postojaće poverljiv semantički web. Implikacija ovoga je da su digitalni potpis i enkripcija u saradnji sa drugim tehnologijama obavezni, u cilju podrške semantičkom webu u kome će korisnici moći da definišu svoje poverenje.

9.11.3 ŠIRENJE POVERENJA KROZ DRUŠTVENE MREŽE

U kontekstu semantičkog weba, FOAF je postao uobičajen i popularan RDF rečnik za opisivanje društvenih podataka [69].

Korišćenjem društvenih mreža moguće je širenje poverenja. Ovi izrazi mogu da se kreiraju korišćenjem FOAF i drugih rečnika.

Pitanje privatnosti

Na webu se informacije najčešće objavljuju javno, što znači da svako može da pristupi tim informacijama.

To dovodi do dva pitanja:

- Ako su informacije dostupne, da li ovo omogućava rizik manipulacije mrežom

poverenja (sigurnosno pitanje);

- Ako su informacije dostupne, kako će da utiču na društvene interakcije uključenih strana.

Prvo pitanje može da se reši kroz istraživanje sigurnosti metrike poverenja [26]. Moguća je upotreba kontrole pristupa kako bi samo poverljive strane mogle da pristupe informaciji.

Drugo pitanje uključuje kompleksne društvene probleme. Ako korisnik objavi informaciju lošeg kvaliteta, korisnik koji poznaje kvalitet sadržaja i korisnika koji je objavio informaciju treba da objavi izjavu poverenja koja upozorava druge korisnike na loš kvalitet sadržaja. Ako je poverenje osnova prijateljstva, a autor ima mogućnost da vidi nisku ocenu poverenja, onda to može da bude štetno za prijateljstvo.

10. ODLUČIVANJE

10.1 PREGLED RAZVOJA NAUKE O ODLUČIVANJU

Ljudi su se oduvek bavili proučavanjem procesa odlučivanja. Ipak, neposredno izučavanje počinje relativno kasno, negde tridesetih godina XX veka. Pre toga su vršene određene analize odlučivanja pri rešavanju različitih tehničkih problema, dok su neke analize rađene i u sociologiji. Međutim sve je to bilo u krajnje rudimentarnoj formi, bez učešća pravih specijalista za odlučivanje, znači isključivo u oblasti matičnih disciplina. Pomoć su pružala znanja iz matematike i ekonomije uz ključnu pretpostavku da donosilac odluke reaguje na potpuno predvidljiv način, pa je zbog toga potrebno imati na raspolaganju samo određeni skup pravila, koja ako se slede, sigurno vode ka dobroj odluci. Naravno, takve pretpostavke, uz činjenicu da u situacijama odlučivanja po pravilu ne raspolažemo pravim podacima i informacijama, ne predstavljaju nikakvu garanciju izbora najbolje odluke. Zbog tih činjenica je dugo vladalo mišljenje da je odlučivanje prevashodno socijalna, a ne tehnička aktivnost. Najveća pokretačka snaga u razvoju teorije odlučivanja došla je razvojem niza metoda koje su kasnije postale poznate kao metode operacionih istraživanja, gde su se na interdisciplinarni način našla na okupu moćna znanja matematike, statistike, ekonomije i sl. Do 1960. godine su i naučnici, koji su se bavili odlučivanjem, imali interes u učenju i razvijanju novih metoda i tehnika u cilju dobijanja optimalnih rešenja različitih operativnih problema. Do 1960. godine ni akademske institucije nisu dovoljno pažnje posvećivale nauci o odlučivanju. Tada je samo šest univerziteta pružalo formalne programe obrazovanja u ovoj oblasti. Šest godina kasnije taj broj se povećao na 37. Takav razvoj je doneo niz pozitivnih rezultata, naročito u oblasti upravljanja. Obezbeđen je poseban uticaj obrazovanih specijalista u poslovnom odlučivanju. Ipak, u jednom takvom razvoju pojavili su se i negativni aspekti. Još uvek se većina naučnika u oblasti odlučivanja bavila teorijskim istraživanjima sa relativno malim mogućnostima primene svojih rezultata na realnim problemima. Sa druge strane većina praktičara je bila prinuđena da svoje probleme, naročito u fazi modeliranja, pojednostavljuje, kako bi ih mogli rešiti metodama kojima se raspolagalo [70].

U fazi razvoja nauke o odlučivanju, koja je nastupila posle 1970. godine, ističu se neke specifičnosti kao što su:

- Veliki naglasak u izučavanju i analizi okruženja, u okviru koga se vrši odlučivanje;
- Veći značaj dobijanju zadovoljavajućih rešenja za izvesne probleme nego traženju optimalnih rešenja;
- Maksimalni naponi za što bolju integraciju kvantitativne analize sa analizama ponašanja i okruženja u rešavanju problema;
- Računarska tehnologija koja se sve više koristi u cilju dobijanja rešenja.

U poslednjih tridesetak godina naglo se razvija tzv. višekriterijumsko odlučivanje, kao dalji korak u sve većoj primeni kvantitativnih metoda u rešavanju realnih problema. Kod višekriterijumskog odlučivanja, donošenje odluka vrši se u prisustvu većeg broja, najčešće konfliktnih kriterijuma. Višekriterijumsko odlučivanje se može, šire posmatrano, podeliti u dve veće kategorije: višeatributsko odlučivanje i višeciljno odlučivanje.

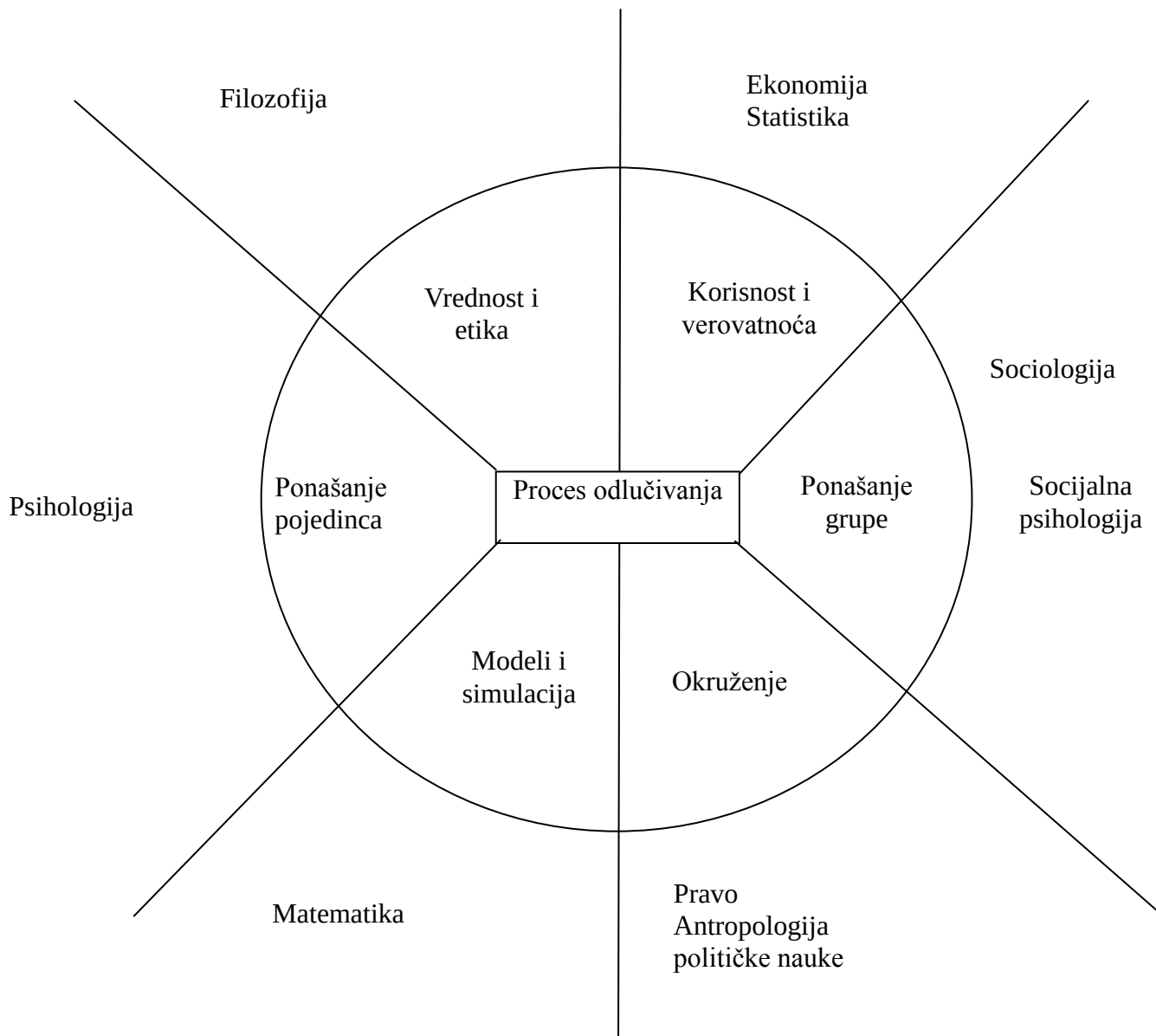
Nagli razvoj informacionih sistema i računarske tehnike, uvodi ih na velika vrata u procese odlučivanja. Specifičnim pristupom primene informacionih sistema u procesima odlučivanja, nametnuli su se i tzv. ekspertni sistemi koji obezbeđuju tzv. inteligentnu podršku u

odlučivanju, što ih čini najboljim alatom, koji u poslu odlučivanja stoji na raspolaganju donosiocima odluka svih nivoa.

10.2 INTERDISCIPLINARNOST ODLUČIVANJA

Odlučivanje, sa procesima koji ga prate, je kompleksna aktivnost. Problem odlučivanja je izrazito interdisciplinaran, pa ga je s toga teško, kako izučavati tako i uspešno sprovoditi. O tome su, svako sa svog aspekta pisali mnogi autori. Posebno se ukazuje na dve reference: Harrison [71] i Klein i Methlie [72].

Harrison je posebno bio studiozan u analizi problematike interdisciplinarnosti odlučivanja. Koristeći rezultate istraživanja ogromnog broja autora iz nekih drugih oblasti, došao je do onog što je nazvao "interdisciplinarnim okvirom odlučivanja" (Slika 51).



Sl. 51. Interdisciplinarni okvir odlučivanja [71]

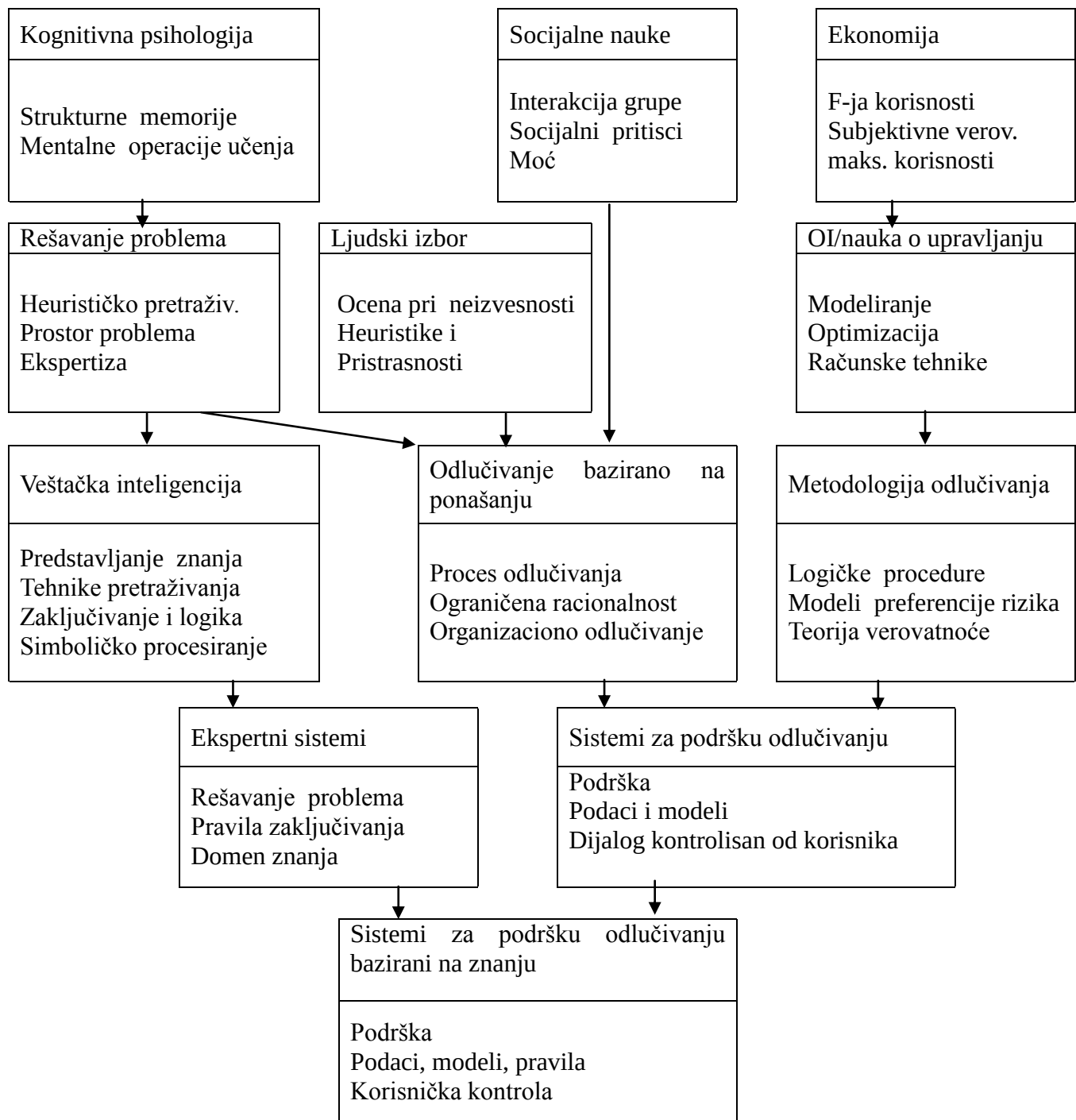
Kao i većina poznatih autora iz oblasti odlučivanja, Harrison smatra da se njenom izučavanju, a pogotovo primeni može pristupiti koristeći jedan od dva važeća pristupa. Prvi pristup, koji se uobičajeno naziva bihejvorističkim aspektom (aspekt ponašanja) odlučivanja, podrazumeva akcenat na izučavanju nauka kao što su: filozofija, psihologija, sociologija i socijalna psihologija kao i pravo, antropologija i političke nauke. Sa druge strane, klasična teorija odlučivanja ima izrazitu tendenciju da svoj fokus usmeri na kvantitativne aspekte odlučivanja, pri čemu, Harrison pod tim podrazumeva: ekonomiju i statistiku i matematiku.

10.2.1 EKONOMIJA I STATISTIKA

Ekonomija i statistika se, po Harrisonu uglavnom primenjuju na procese odlučivanja kroz korišćenje koncepta korisnosti. Taj koncept podrazumeva da pojedinac kontinualno teži da maksimizira svoje preference, odnosno korisnost. Analiza rizika se koristi i sprovodi u situacijama odsustva tačnih informacija, ali uz raspolaganje ograničenim znanjima o željenim rezultatima. Donosilac odluke je u najgoroj situaciji kada ne raspolaže ni sa kakvim informacijama o željenim rezultatima.

10.2.2 MATEMATIKA

Matematika, kao naučna disciplina se pojavljuje u procesima odlučivanja prvenstveno kroz razvoj matematičkih modela i simulacija i njihovog korišćenja radi modeliranja i simuliranja situacija iz realnog života. Na slici 52 je prikazan naučni okvir odlučivanja.



Sl. 52. Naučni okvir odlučivanja [72]

10.3 PROCES DONOŠENJA ODLUKA

Donošenje odluke je proces koji se sastoji u obavljanju određenih postupaka ili aktivnosti, koje treba da ukažu na problem, njegove uzroke i pravce rešavanja problema. U osnovi može se govoriti o dva pristupa procesu donošenja odluka [73] i to:

- tradicionalni i

- savremeni.

10.3.1 TRADICIONALNI, ANALITIČKI MODEL DONOŠENJA ODLUKA

Po tradiciji, naučnici su otkrili da je korisno da se proces donošenja odluka opiše kao serija postupaka ili koraka koje pojedinac ili grupa mora da preduzme da bi se rešio neki problem. Jedan opšti model procesa donošenja odluka može da nam pomogne da shvatimo kompleksnu prirodu organizacionog odlučivanja (slika 51). Ovaj model ističe dva važna aspekta procesa donošenja odluka: formulaciju, proces shvatanja problema, i donošenja odluke u vezi s njim, i implementaciju, proces izvršenja odluke koja je doneta. Međutim, ovaj model je veoma koristan, jer ukazuje na koji način proces donošenja odluka u principu funkcioniše [74].

Prvi korak u odlučivanju je identifikacija problema. Da bismo mogli da odlučimo kako da rešimo problem, moramo prvo da ga prepoznamo i opišemo. Ovaj korak nije tako jednostavan. Činjenica je da ljudi često iskrivljuju, izostavljaju, ignorišu ili zanemaruju informaciju koja im je dostupna, a predstavlja bitne nagoveštaje da postoje problemi. Negiranje problema je prva smetnja na putu za njegovo rešavanje.

Pošto se problem identifikuje, sledeći korak je da se odrede ciljevi koji će se postići rešavanjem problema. Potrebno je da se problemi sagledavaju na takav način da mogu da se identifikuju moguća rešenja.

Treći korak u procesu donošenja odluke je donošenje predodluke. Ranije su se donosioci odluka oslanjali na svoju intuiciju ili empirijski dobijene informacije u vezi sa ponašanjem. U novije vreme, međutim, napravljeni su računarski programi u kojima je sumiran veliki broj ovih informacija u formi koja je veoma pristupačna. Ti, takozvani sistemi za podršku u donošenju odluka (**D**ecision **S**upport **S**ystems - **DSS**) mogu da budu samo onoliko dobri koliko su validni naučni podaci ugrađeni u njih. Istraživanja su pokazala da su DSS tehnike efektne kao pomoć ljudima da donesu odluku i kako da reše neki problem. Smatra se da korišćenje tehnologije za donošenje odluka donosi bolje rezultate nego kada se ta tehnologija prilikom odlučivanja ne primenjuje. Dalje, otkriveno je da su računarske DSS tehnike bile veoma korisne kao podstrek ljudima da smisle veći broj rešenja.

Četvrti korak ovog procesa je smišljanje alternativa, faza u kojima se identifikuju moguća rešenja datog problema. U nastojanju da iznađu rešenje, ljudi se često oslanjaju na ranije korišćene pristupe koji bi mogli da im ponude gotova rešenja.

Peti korak predstavlja procenjivanje alternativnih rešenja, upravo zbog toga što sva navedena rešenja ne moraju da budu podjednako prihvatljiva. Koja je solucija najbolja? Različite alternative treba da se identifikuju. Neke bi možda bile efektnije od drugih, a neke bi se možda teže primenile od drugih.

Na sledećem koraku se pravi izbor. Nakon što se nekoliko alternativa proceni, bira se ona za koju se smatra da je najprihvatljivija. Različiti pristupi procesu odlučivanja nude različite stavove prema pitanju koliko temeljno ljudi razmatraju alternative i koliko su optimalna njihova odabrana rešenja. Izbor pravog toka akcije je korak na koji najčešće pomišljamo kada razmišljamo o procesu odlučivanja.

Sedmi korak je implementacija izabrane alternative. Drugim rečima, sprovodi se izabrana alternativa.

Osmi i poslednji korak je ishod. Merenje efektnosti odluka koje su važan faktor za uspeh.

Veoma je važno da se traži povratna informacija o efektivnosti bilo kog primenjenog rešenja. Ako rešenje funkcioniše problem se smatra rešenim. Ako ne, treba primeniti novo rešenje.

10.3.2 TEORIJA „IMIDŽA“: INTUITIVNI PRISTUP DONOŠENJU ODLUKE

Teorija imidža se prvenstveno bavi odlučivanjem u pogledu preduzimanja nekog kursa akcije ili promene kursa akcije. Prema ovoj teoriji: ljudi donose odluke o prihvatanju na osnovu jednostavnog procesa. Prvo se testira kompatibilnost, upoređivanjem u kom stepenu je određeni kurs akcije konzistentan sa raznim imidžima - naročito sa individualnim principima, trenutnim ciljevima i planovima za budućnost. Ako ne postoji kompatibilnost u pogledu bilo koje od ovih tačaka donosi se odluka o odbijanju. Ako se test kompatibilnosti uspešno prođe, sledi test profitabilnosti. Ljudi razmatraju do koje mere bi korišćenje različitih opcija odgovaralo njihovim vrednostima, ciljevima i planovima. Zatim se donosi odluka o najboljem predlogu. Ovi testovi se koriste unutar određenog okvira odlučivanja - tj. u saglasnosti sa značajnim informacijama u vezi s kontekstom odluke (kao što je ranije iskustvo). Osnovna ideja se zasniva na tome da mi učimo na osnovu prošlosti i ona nas usmerava kada donosimo odluke [75].

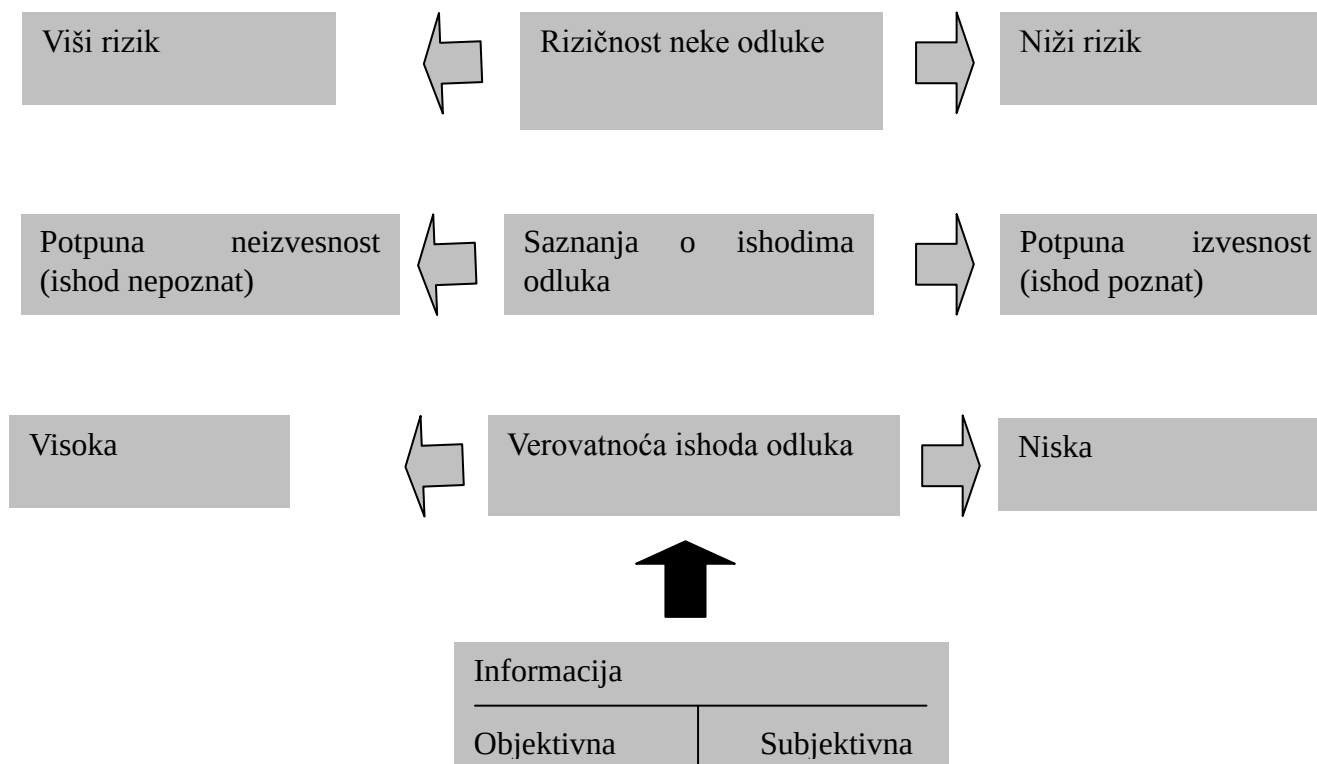
Prema teoriji imidža, proces donošenja odluka je veoma jednostavan i brz. Ova teorija pretpostavlja da ljudi ne zastajkuju i ne razmišljaju kada donose odluke, već do njih dolaze jednim ujednačenim, intuitivnim procesom sa minimalnom kognitivnom obradom.

Opisali smo tradicionalni, analitički pristup odlučivanja i teoriju imidža - noviji, intuitivniji pristup istom procesu. Oba ova pristupa su potkrepljena dokazima, i ne treba jedan posmatrati kao zamenu za drugi. Umesto toga u donošenje odluka može da bude uključeno nekoliko procesa. Ne sprovodi se svako odlučivanje na isti način: ponekad ono može da bude analitičko, a nekad više intuitivno. Naučnici modernog organizacionog ponašanja prihvataju vrednosti i jednog i drugog pristupa. Kao što može da se pretpostavi, oba opisana procesa donošenja odluka mogu da se primene na veliki broj različitih odluka.

10.3.3 IZVESNE I NEIZVESNE ODLUKE

Stepen izvesnosti ili neizvesnosti izražava se kao rizik. Sve odluke sadrže određeni stepen rizika - od potpune izvesnosti (bez rizika), do potpune neizvesnosti (visok rizik). Da bi se donele najbolje moguće odluke, ljudi se trude da "upravljaju" rizicima koje preduzimaju - tj. oni umanjuju rizičnost neke odluke na taj način što se trude da dođu do informacija koje su relevantne za nju.

Ono što definiše ishod rizičnim, ili ne, je verovatnoća da se željeni ishod postigne. Donosioći odluka se trude da dođu do informacija o verovatnoći ili mogućnosti da se neke stvari dogode, pod uslovom da su se druge stvari dogodile. Mnoge odluke se zasnivaju na subjektivnim verovatnoćama - ličnim uverenjima ili predosećajima šta će se dogoditi (slika 53).



Sl. 53. Rizičnost odluka: pregled

Tačne informacije smanjuju neizvesnost u situacijama kada se donose odluke. Računarstvo i informatika su u mnogome pomogli ljudima da brzo donose odluke, a da pri tom koriste najtačnije i najpotpunije raspoložive informacije. Čitav niz informativnih servisa trenutno dostupnih je smišljen da bi pružio donosiocima odluka najnovije informacije u vezi sa pitanjima o kojima treba da odluče.

10.4 OGRANIČENJA ZA DONOŠENJE KVALITETNIH ODLUKA

Mnoge donete odluke nisu kvalitetne zato što nisu utemeljene na činjenicama, odnosno na potpunim i pouzdanim informacijama. Na činjenici da postoje ograničenja, Sajmon je kreirao teoriju ograničene racionalnosti i zadovoljavajućeg rešenja [76]. On je ukazao na ograničenost ljudskih sposobnosti da se savladaju sve potrebne informacije koje bi vodile perfektnoj odluci. Ljudi su sa svojim sposobnostima, veštinama i znanjem prvi ograničavajući faktor efikasnog odlučivanja. Ljudi su nesavršeni donosioci odluka iz najmanje dva razloga: prvo, zbog ograničenosti njihovih sposobnosti da prikupe sve relevantne informacije da ocene njihovu važnost i da ih tačno i temeljno obrade; i drugo, zbog predrasuda koje se tiču formulacije problema, odnosno načina kako im je problem prezentiran i načina donošenja odluka, koji je često pristrasan, kada se odluke donose na bazi slobodne procene.

Na kvalitet odluka utiču i druga individualna ograničenja, kao što su:

- etika i lične moralne norme, koje se ogledaju u: integritetu ličnosti i konzistentnom, odnosno nekonzistentnom ponašanju donosilaca odluke,

- okruženje, odnosno uslovi u kojima se odluka donosi,
- tradicija i kultura,
- resursi: vreme, informacije, novac itd. i
- tačnost informacija.

10.5 VIŠEKRITERIJUMSKO ODLUČIVANJE

Suština višekriterijumskog odlučivanja je da se uoče svi mogući kriterijumi za poređenje mogućih alternativa, da se sve alternative uporede po izborima relevantnih kriterijuma i tako izabere najpovoljnija alternativa.

U realnim situacijama skoro nikad nema alternativa koje su najbolje po svim kriterijumima. Zbog toga su razvijene i razvijaju se posebne metode za višekriterijumsku optimizaciju.

Važnu ulogu igraju i relevantne težine koje dodeljujemo pojedinim kriterijumima. Ove težine utiču na donošenje odluke. U ovom radu se bavimo njihovim određivanjem, gde pored subjektivnog određivanja, uzimamo u obzir i objektivne pokazatelje za njihovo određivanje, a koji proizilaze iz karakteristika problema kojima se bavimo.

11. EDUKACIONI MODEL NA UNIVERZITETIMA

Prikažaćemo jedan edukacioni model koji je tipičan za univerzitete u SAD-u. Na osnovu njega i uz određene modifikacije realizovaćemo edukacioni model tipičan za visokoškolski sistem u našoj zemlji.

Školska godina počinje 15. Avgusta. Pratimo studenta prve godine koji je upisao određeni univerzitet posle završene srednje škole sa željom da studira određenu naučnu oblast, odnosno studijski program. On je upoznat sa predmetima koje mora da položi u toku studija da bi došao do željene diplome. Određeni predmeti su obavezni, a pojedine može birati u zavisnosti od svojih pretenzija i želja i u zavisnosti od toga da li će se ti predmeti držati u semestrima kada ih on može slušati (da li je za njih zainteresovano dovoljno studenata). Student je takođe prošao (ako je to potrebno) odgovarajuće provere predznanja za slušanje željenih predmeta. Na primer, ako želi da sluša matematički predmet mora posedovati znanje iz srednjoškolske algebre i trigonometrije. Ako se na proveru ustanovi da ta znanja nema, biće posavetovan da studira nešto drugo (gde ova predznanja nisu potrebna). Ako student ne želi da promeni stav, upućuje se na časove iz predmeta koji prethode datom predmetu (srednjoškolska algebra ili trigonometrija ili oba). Ovi časovi ne ulaze u fond časova za završavanje studija i posebno se naplaćaju. Posle završenih časova ponovo se ide na proveru i u narednom semestru se uzima željeni kurs. Dakle, svi studenti koji su došli na kurs poseduju neophodno minimalno predznanje za slušanje i savladavanje kursa.

Kurs drži nastavnik i najčešće ima maksimalno pet časova nedeljno iz tog predmeta sa tom grupom studenata. Student nedeljno sa nastavnicima provede od dvanaest do dvadeset časova.

Pored ovih časova sa nastavnicima, student ima i minimalno tri časa rada sa savetnicima koji su nezavisni od nastavnika, rade sa grupama studenata na savlađivanju gradiva, izradi domaćih zadataka, seminarskih radova itd. Savetnici su za svoj rad odgovorni menadžmentu univerziteta.

Nastavnik može imati i asistenta – svog postdiplomca ili doktoranta koji može držati manje od pola nastavnih časova. Odgovornost i za ove časove je na nastavniku.

Savetnici prate rad svakog svog studenta ali i rad svakog od nastavnika.

Nastavnik radeći sa studentima prati njihovo angažovanje, aktivnosti na časovima i izradu domaćih zadataka. Ako student ima problem to se evidentira i rešava. Ako se proceni da je problem nerešiv student se savetuje da promeni smer.

Sastavni deo ispita su i kolokvijumi koji se održavaju u toku semestra.

Roditelji (ili drugi finansijeri) studenata koji nisu zainteresovani za studiranje se obaveštavaju i takvi studenti se po pravilu udaljavaju na početku studiranja.

Do kraja decembra nastavnik iz svakog predmeta formira ocenu o radu svakog studenta i to šalje šefu departmana. Nastavnik saopštava ocene i studentima i oni mogu tada ako žele da poprave ocene.

Studenti na kraju semestra popunjavaju anketu o tom predmetu i radu nastavnika sa njima. Savetnici takođe ocenjuju rad svakog studenta iz svakog predmeta ali i rad nastavnika sa tim studentima iz datog predmeta. Te ocene dobija i šef departmana. Posle toga komisija departmana na osnovu ocene nastavnika i mišljenja savetnika zaključuje ocene za svakog studenta i šalje ih u registar.

Novi semester počinje sredinom januara kada studenti upisuju nove predmete, odnosno ponovo upisuju predmete koje nisu položili.

Nastavnici će biti više puta ocenjivani u toku školske godine od strane studenata, savetnika i menadžmenta.

Najbolja strana ovog sistema je što nema studenata koji su “izgubili” godinu, nema naknadnih ispitnih rokova, nastavnik se uvek može zameniti novim nastavnikom bez bojazni da će neko biti oštećen. Student lako može promeniti univerzitet i odrediti tempo završavanja studija.

Sličan model je prihvaćen bolonjskom deklaracijom i za studije u Evropi [77], [78], koji se i kod nas primenjuje uz određene modifikacije.

11.1 OCENJIVANJE RADA NASTAVNIKA I STUDENATA

11.1.1 OSNOVNE PRETPOSTAVKE (HIPOTETIČKI SISTEM BEZ SAVETNIKA)

Nastavnik drži kurs iz nekog predmeta studentima koji su stekli odgovarajuće predznanje položivši odgovarajuće kurseve u prethodnim semestrima odnosno koji su stekli odgovarajuće predznanje u odgovarajućim srednjim školama ako je u pitanju predmet iz startne godine. Podaci o predznanjima postoje za svakog studenta.

Studenti dolaze na nastavu, rade domaće zadatke i polažu kolokvijume posle određenih celina po sledećim principima:

- Posle završene određene celine nastavnik određuje dan probnog kolokvijuma, tog dana se radi probni kolokvijum i nastavnik ga ocenjuje;
- Sledeći čas se radi ispravka, ukazuje na greške i komentariše šta se može popraviti;
- Za sledeći čas se zakazuje kolokvijum i tada se radi i potom ocenjuje i analizira – ustanovljava se saglasnost sa predznanjem i predispozicijama studenata;
- Tako se radi do kraja i na poslednjem času se sumiraju rezultati svih kolokvijuma i formira se završna ocena za svakog studenta;
- Zatim se analizira ukupna saglasnost sa polaznim predznanjima.

Tog dana i studenti daju svoje mišljenje o radu na predmetu – da li su dobili to što su očekivali, i ocenjuju rad nastavnika i asistenata.

Ocene rada nastavnika se mogu izvesti na sledeći način:

$$O = \frac{p_1 O_1 + p_2 O_2 + p_3 O_3}{p_1 + p_2 + p_3} \quad (11)$$

O_1 - komponenta koja uzima u obzir saglasnost novodobijenih ocena i prethodnih ocena, na osnovu sledećih ocena i kriterijuma:

1. Nema saglasnosti uspeh je jako nizak;
2. Nema saglasnosti i uspeh je niži;
3. Nema saglasnosti a uspeh je manji ili veći;
4. Ocena je saglasna sa ocenom iz ulaznih kurseva i prosek je malo niži;

5. Ocena je saglasna sa ocenom iz ulaznih kurseva i podignuta je u proseku.

Komponenta O_2 je srednja ocena studenata koji ocenjuju nastavnika:

- Prosečna ocena koju studenti daju nastavniku se nalazi u opsegu od 1 do 5.

Komponenta O_3 je ocena koju menadžment daje nastavniku:

- Ocena naučnog (studijskog) rada nastavnika i njegov uticaj u naučnoj (stručnoj) javnosti se nalazi u opsegu od 1 do 5.

Koeficijenti p_1, p_2 i p_3 su težinski koeficijenti čiji je zbir jednak 1.

NAPOMENA: Podrazumeva se da je obezbeđeno ocenjivanje bez prepisivanja – jer bi potencijalna kontrola znanja (koja se može izvršiti) studenata koji su završili neki kurs, morala biti saglasna sa datim ocenama.

11.1.2 MODIFIKACIJA OCENE ZA NAŠE USLOVE

Polazimo od sledeće formule za određivanje modifikovane druge komponente ukupne ocene nastavnika:

$$O_2^{(1)} = \frac{\sum_{i=1}^a q_i O_{2i}^{(1)}}{\sum_{i=1}^a q_i} \quad (12)$$

Gde je:

- q_i - procenat dolaznosti određenog studenta na časove (uzima se samo ako je dolaznost veća od 50%, u protivnom je vrednost 0);
- a – ukupan broj studenata;
- i – indeks koji uzima vrednost od 1 do a ;
- $O_{2i}^{(1)}$ - ocena i -tog studenta prilagođena uslovima studiranja u našoj zemlji.

Pri definisanju modifikovane prve komponente ocene (prosečne ocene i saglasnosti) $O_1^{(1)}$, takođe treba uzeti u obzir prisutnost čak i u slučaju da je student položio ispit, a nije dolazio na nastavu. U krajnjoj liniji uticaj takvih studenata se treba smanjiti korišćenjem težinskih koeficijenata.

Formula (12) može u principu da se koristi i za ocenjivanje sa drugačijim vrednostima za p_i za različite situacije (ponekad su nedolasci u procesu nastave opravdani). U skladu sa ovim trebalo bi voditi računa o prisutnosti nastavi.

NAPOMENA: U našim uslovima gde se ne studira ispit za ispit, semestar za semestar, gde se ne slušaju u svakom semestru svi predmeti studijskog programa, već se studira godina – što često dovodi do nepotrebnog gubljenja vremena, naročito sposobnih studenata, a isto i onih koji su vredniji, ima smisla uvesti i kategoriju prolaznosti na godišnjem nivou (odnosi se na motivaciju studenata koji su neredovno dolazili da uspešno završe započeti predmet).

Dakle, prolaznost na godišnjem nivou bi mogla biti četvrta komponenta ocene za nastavnika i onda bi formula (11) dobila oblik:

$$O = \frac{p_1 O_1 + p_2 O_2 + p_3 O_3 + p_4 O_4}{p_1 + p_2 + p_3 + p_4} \quad (13)$$

11.2 ODREĐIVANJE TEŽINSKIH KOEFICIJENATA KORISTEĆI AHP (ANALYTIC HIERARCHY PROCESS) METOD

Polazeći od poznatih činjenica da problemi koji se rešavaju višekriterijumskom optimizacijom imaju najčešće različit stepen značaja za donosioca odluke, razvijeno je više postupaka za određivanje vrednosti težinskih koeficijenata. Često donosioci odluka unapred zadaju ove vrednosti na način za koji se može reći da je iskustven (subjektivan). Pri definisanju kriterijuma mora se voditi računa da se neki kriterijumi ne ponavljaju (lako se može dogoditi da se elementi koji ulaze u neki kriterijum implicitno ugrade u još neki od kriterijuma). Vrlo je važno pravilno definisati težine kriterijuma. Ovi momenti su u nadležnosti neposrednog donosioca odluke. Analitičar koji pomaže donosiocu odluke mora u svakom trenutku formiranja višekriterijumskog modela da prati proces tog formiranja i da ukazuje na eventualne propuste i nedostatke, u cilju donošenja najbolje odluke.

U radu je korišćen AHP (Analytic Hierarchy Process). Analitički hijerarhijski proces je višekriterijumski pristup donošenja odluka koji je predstavio Saaty [87], [88]. AHP je privukao interesovanje mnogih istraživača uglavnom zbog dobrih matematičkih svojstava metoda i činjenice da je prilično lako dobiti potrebne ulazne podatke. AHP je alat za podršku pri odlučivanju koji može da se koristi za rešavanje kompleksnih problema odlučivanja. On koristi više nivoa hijerarhijske strukture ciljeva, kriterijuma, podkriterijuma i alternativa. Odgovarajući podaci su dobijeni korišćenjem skupa „pairwise“ poređenja dva kriterijuma (par kriterijuma). Ova poređenja se koriste za dobijanje težine od značaja za kriterijum odluke, i relativne mere performansi alternativa u pogledu svakog individualnog kriterijuma odlučivanja. Ako poređenja nisu savršeno konzistentna, onda AHP metod obezbeđuje mehanizam za poboljšanje konzistentnosti. Višekriterijumsko odlučivanje (MCDM Multi-criteria decision-making) igra ključnu ulogu u mnogim (realnim) životnim problemima.

AHP i njegova upotreba pairwise poređenja su inspirisali stvaranje mnogih drugih metoda odlučivanja. Pored svoje široke prihvaćenosti, AHP metod je i kritikovan, iz teorijskih i praktičnih razloga. Postoji veliki broj različitih varijanti originalnog AHP-a.

Pairwise poređenja se koriste za utvrđivanje relativnog značaja svake alternative u odnosu na svaki kriterijum. U ovom pristupu donosilac odluke mora da izrazi svoje mišljenje o vrednosti jednog pairwise poređenja u jednom trenutku. Obično, donosilac odluke mora da izabere svoj odgovor između 10-17 diskretnih izbora. Svaki izbor je jezička fraza (izraz). Primer takve jezičke fraze je: „A je važnije od B“, ili „A je iste važnosti kao B“, ili „A je malo važnije od B“, itd.

Glavni problem pairwise poređenja je kako da se kvantifikuju lingvistički izbori, izabrani od strane donosioca odluke tokom evaluacije. Sve metode koje koriste pristup pairwise poređenja izražavaju kroz brojeve kvalitativne odgovore donosioca odluke, koji su u većini slučajeva odnos celih brojeva. Kako su pairwise poređenja ključna u procesu donošenja odluka, ispravno kvantifikovanje je najvažniji korak u metodi višekriterijumskog odlučivanja koja koristi kvalitativne podatke.

Skala	Faktor poređenja i i j
1	Podjednako važno
3	Slabo važno
5	Jako važno
7	Veoma važno
9	Ekstremno važno
2, 4, 6, 8	Međuvrednosti

Tabela 12. Saaty-eva skala pairwise poređenja sa devet vrednosti

Pairwise poređenja se kvantifikuju korišćenjem skale (tabela 12). Ovakva skala je jedan-na-jedan mapiranje između skupa diskretnih jezičkih izbora dostupnih donosiocu odluke i diskretnog skupa brojeva koji predstavljaju važnost, ili težinu, prethodnih jezičkih izbora. Postoji veliki broj alternativnih skala i sve one polaze od različitih psiholoških teorija.

Godine 1846. Weber navodi svoj zakon o podsticanju merljive veličine. Prema njegovom zakonu, promena osećaja se primećuje ako se podsticaj povećava konstantnim procentom samog podsticaja [87]. To znači da ljudi nisu u stanju da naprave izbor iz beskonačnog skupa. Na primer, ljudi ne mogu da naprave razliku između dve veoma bliske vrednosti od značaja (3,00 i 3,02). Psihološki eksperimenti su takođe pokazali da pojedinci ne mogu istovremeno da porede više od sedam objekata (plus ili minus dva) [88]. Ovo su glavni razlozi koji su korišćeni od strane Saaty-a da uspostavi 9 kao gornju granicu, 1 kao donju granicu i razliku jedan između dve uzastopne vrednosti skale. Vrednosti pairwise poređenja u AHP-u su određene prema skali koju je uveo Saaty [87], prema kojoj su raspoložive vrednosti za pairwise poređenja članovi skupa: {9, 8, 7, 6, 5, 4, 3, 2, 1, 1/2, 1/3, 1/4, 1/5, 1/6, 1/7, 1/8, 1/9}.

Kao ilustrativan primer posmatramo izbor najboljeg kompjuterskog sistema od tri alternative konfiguracije A, B i C. Takođe, pretpostavimo da je jedan od kriterijuma odluke proširivanje harvera (fleksibilnost dodavanja sistemu drugih perifernih uređaja kao što je štampač, memorija itd.). Pretpostavimo da je sistem A mnogo bolji od sistema B, a sistem C je najmanje poželjan kada je kriterijum proširenja hardvera u pitanju. Tabela 13 definiše matricu procena.

C_1 Proširivanje hardvera	A	B	C
A	1	6	8
B	1/6	1	4
C	1/8	1/4	1

Tabela 13. Matrica procena

Kada se sistem A poredi sa sistemom B, tada donosilac odluke odlučuje da se sistem A klasifikacije između „jako važno“ i „veoma važno“ u odnosu na sistem B. Dakle, odgovarajuće poređenje uzima vrednost 6. Slična interpretacija važi i za ostale stavke.

Sledeći korak je određivanje glavnog eigen vektora prethodne matrice. Datoj matrici procena sa pairwise poređenjima, odgovarajući maksimalni eigen vektor je približno određen pomoću geometrijske sredine svakog reda, tj. elementi u svakom redu se množe međusobno i onda se uzima n -ti koren (gde je n broj elemenata u redu). Zatim se brojevi normalizuju deljenjem sa sumom elemenata reda. Dakle, za prethodnu matricu odgovarajući vektor prioriteta je: (0,754, 0,181, 0,065). Postoji alternativno rešenje gde se koristi aritmetička sredina za elemente kolone, kao i suma elemenata kolone.

Važno praktično pitanje u AHP metodologiji je nekonzistentnost (nedoslednost) pairwise poređenja. Ako su sva poređenja savršeno konzistentna, onda sledeći odnos treba uvek da važi za bilo koju kombinaciju poređenja uzetih iz matrice procene: $a_{ij} = a_{ik}a_{kj}$. Međutim, savršena konzistencija se retko dešava u praksi. AHP pairwise poređenje u matrici procene se smatra adekvatno konzistentno ako je odgovarajući odnos konzistencije (CR) manji od 10% [87]. CR koeficijent se izračunava na sledeći način. Prvo se određuje indeks konzistencije (CI). To se radi dodavanjem kolona u matrici procene i množenjem dobijenog vektora, vektorima prioriteta (tj, aproksimiranom vrednošću eigenvectora) dobijenim ranije. Ovo daje aproksimaciju maksimalne eigen vrednosti, označene sa λ_{max} . Zatim se vrednost CI izračunava formulom: $(\lambda_{max} - n)/(n - 1)$. Sledi, CR se dobija deljenjem vrednosti CI sa slučajnim konzistentnim indeksom (RCI) iz tabele 14.

Kada se primene ove aproksimacije na prethodnu matricu procene, dobija se sledeće: $\lambda_{max} = 3,136$, $CI = 0,068$ i $CR = 0,117$. Ako je vrednost CR veća od 0,10, onda je dobra ideja da se problem i dalje proučava i ponovo procenjuje pairwise poređenjem.

N	1	2	3	4	5	6	7	8	9
RCI	0	0	0,58	0,90	1,12	1,24	1,32	1,41	1,45

Tabela 14. RCI vrednosti za različite vrednosti n

Nakon što se alternative uporede međusobno u odnosu na svaki kriterijum odlučivanja, i izvedu individualni vektori prioriteta, sledi korak sinteze. Vektori prioriteta postaju kolone matrice odlučivanja. Težine od važnosti kriterijuma se takođe određuju pairwise poređenjem. Dakle, ako problem ima M alternativa i N kriterijuma, onda donosilac odluka konstruiše N matrica procene (jednu za svaki kriterijum) veličine $M \times M$ i jednu matricu procene veličine $N \times N$ (za N kriterijuma). Konačno, data je matrica odluke sa konačnim prioritetima, označena sa A_{AHP}^i , alternativa u pogledu svih kombinovanih kriterijuma određenih sledećom formulom:

$$A_{AHP}^i = \sum_{j=1}^N a_{ij}w_j \text{ za } i = 1, 2, 3, \dots, M. \quad (14)$$

U slučaju idealnog AHP režima, kolone matrice odlučivanja su normalizovane deljenjem sa najvećom vrednošću svake kolone.

12. REALIZOVANO REŠENJE

Kao primer uzećemo postupak ocenjivanja profesora za predmet „Java programiranje“ na Visokoj poslovnoj školi strukovnih studija u Blacu, na drugoj godini, u četvrtom semestru sa fondom časova 2+2, pri čemu nastavnik drži i predavanja i vežbe.

Polazimo od formule (13) i definišemo sastavne komponente ukupne ocene nastavnika O , pri čemu je:

- O_1 - komponenta koja uzima u obzir saglasnost novodobijenih ocena i prethodnih ocena;
- O_2 - srednja ocena studenata koji ocenjuju nastavnika;
- O_3 - ocena koju menadžment daje nastavniku;
- O_4 - prolaznost na predmetu u toku cele školske godine.

Opisaćemo način dobijanja komponente ocene O_1 . Ova komponenta pokazuje koliki je uticaj nastavnika na poboljšanje (smanjenje) ulaznih ocena za srodne predmete. Koristimo sledeću formulu:

$$S = \frac{1}{a} \sum_{i=1}^a |t_i - p_i| \quad (15)$$

Gde je:

- S – srednja vrednost modula odstupanja novodobijene ocene u odnosu na ulaznu (prethodnu) ocenu;
- a – ukupan broj studenata;
- t – novodobijena (trenutna) ocena;
- p – prethodna ocena.

Tabela 15 daje prikaz odstupanja trenutnih od ulaznih ocena za grupu od 12 studenata. Primenjujući formulu (15) dobija se vrednost za $S = 0,208$.

Student	Prethodni predmeti (p)	Ocena (t)	Razlika
Student1	6.5	7	0.5
Student2	10	10	0
Student3	6	6	0
Student4	6	6	0
Student5	7	6	-1
Student6	6	6	0
Student7	8	8	0
Student8	6.5	7	0.5
Student9	6	6	0
Student10	6	6	0
Student11	7.5	8	0.5
Student12	6	6	0
Σ			0.5
S			0,208

Tabela 15. Odstupanje od ulaznih ocena

Za prethodno definisanu vrednost S iz prvog reda tabele 16 sledi da je vrednost prve komponente ukupne ocene $O_1 = 3$.

		S	$\leq$	0,15	$O_1 = 5$
0,15	$<$	S	$\leq$	0,20	$O_1 = 4$
0,20	$<$	S	$\leq$	0,30	$O_1 = 3$
0,3	$<$	S	$\leq$	1	$O_1 = 2$
		S	$>$	1	$O_1 = 1$

Tabela 16. Granice odstupanja i odgovarajuće ocene

Prosečna ocena nastavnika O_2 za odgovarajući predmet dobija se direktnim popunjavanjem upitnika gde studenti sami ocenjuju svoje nastavnike ($O_2 = 4,8$).

Ocena menadžmenta O_3 se dobija popunjavanjem upitnika gde menadžment ocenjuje nastavnike (naučni rad u tekućoj godini, $O_3 = 4,5$).

Tabela 17 prikazuje procenat prolaznosti po rokovima i srednju vrednost procenta prolaznosti studenata na kraju školske godine.

Rok	Prolaznost
Jun	57,62%
Jul	45,45%
Septembar	83,33%
Oktoibar	100%
p	71,6%

Tabela 17. Procenat prolaznosti po ispitnim rokovima i prosečna prolaznost.

Komponenta O_4 se dobija iz prvog reda tabele 18. ($p = 71,6$ sledi $O_4 = 5$).

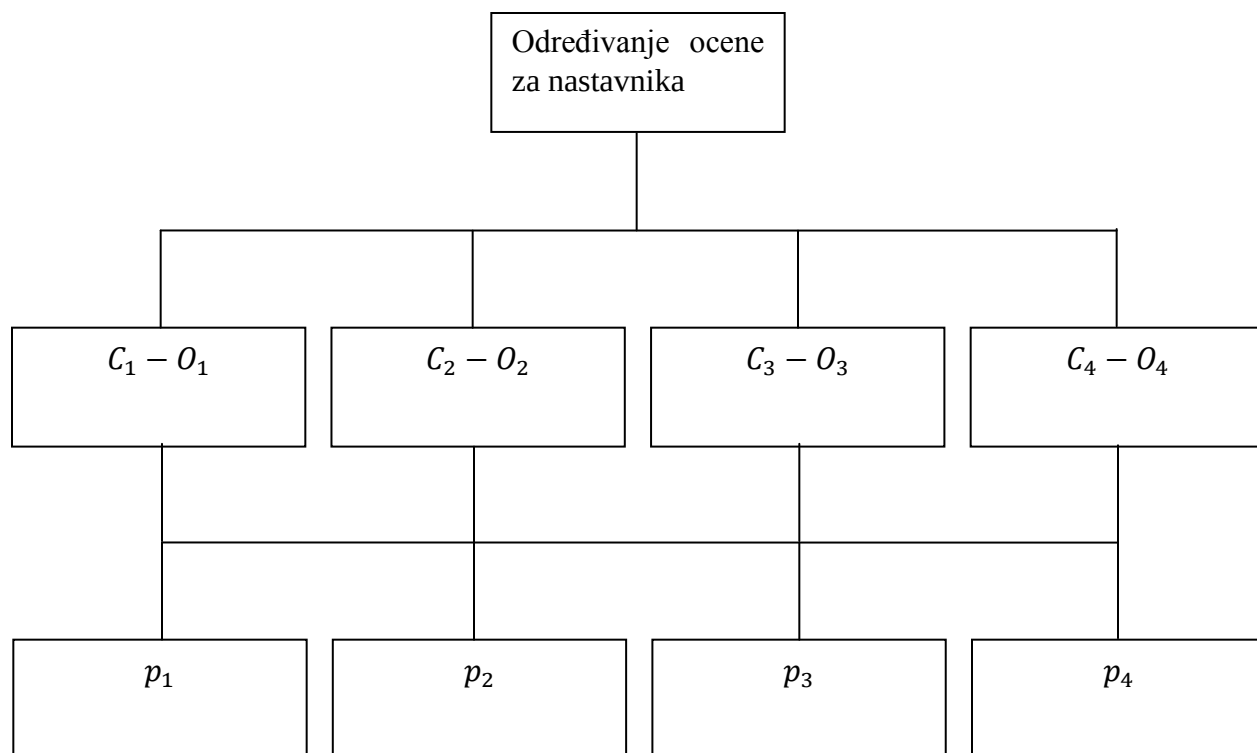
70%	$<$	p	$<$	90%	$O_4 = 5$
60%	$\leq$	p	$<$	70%	$O_4 = 4$
50%	$\leq$	p	$<$	60%	$O_4 = 3$
40%	$\leq$	p	$<$	50%	$O_4 = 2$
91%	$<$		$\leq$	100%	
		p	$<$	40%	$O_4 = 1$

Tabela 18. Veza između ocene O_4 i prosečne prolaznosti.

Iz tabele 18 se može videti i uticaj menadžmenta na definisanje prolaznosti studenata po predmetima i rokovima.

Ocene koje studenti daju nastavnicima koji pripadaju ili ne pripadaju menadžmentu visokoškolske ustanove se razlikuju. Ta razlika je definisana u tabeli 25 kroz težine koje odgovaraju različitim komponentama ukupne ocene.

Opisaćemo način na koji primenjujemo AHP metodu za određivanje težina kriterijuma polazeći od definisanja hijerarhije određivanja ocena nastavnika i odgovarajuće matrice procena.



Sl. 54. Hijerarhija određivanja ocene za nastavnika

Korak 1: Matrica procena A određuje važnost četiri kriterijuma odlučivanja koji odgovaraju delovima ukupne ocene nastavnika.

Kriterijum	C_1	C_2	C_3	C_4
C_1	1	0,25	0,20	0,25
C_2	4	1	3	2
C_3	5	0,33	1	2
C_4	4	0,50	0,50	1

Tabela 19. Matrica procena važnosti četiri kriterijuma odlučivanja

Korak 2: Izračunavanje sume kolone $\sum_i C_{ij}$ za svaku kolonu.

Kriterijum	C_1	C_2	C_3	C_4
C_1	1	0,25	0,20	0,25
C_2	4	1	3	2
C_3	5	0,33	1	2
C_4	4	0,50	0,50	1
Suma	14	2,08	4,7	5,25

Tabela 20. Izračunavanje sume kolona

Korak 3: Normalizacija (deljenje) vrednosti svake ćelije koristeći izraz $X_{ij} = \frac{C_{ij}}{\sum_i C_{ij}}$.

Kriterijum	C_1	C_2	C_3	C_4
C_1	0,071	0,120	0,043	0,048
C_2	0,286	0,481	0,638	0,381
C_3	0,357	0,159	0,213	0,381
C_4	0,286	0,240	0,106	0,190

Tabela 21. Normalizacija vrednosti

Korak 4: Izračunavanje sume reda koristeći izraz $R_i = \sum_j X_{ij}$ i težine koristeći izraz $W_i = \frac{R_i}{n}$; gde je n = broj kriterijuma (4).

Kriterijum	C_1	C_2	C_3	C_4	Suma	W
C_1	0,071	0,120	0,043	0,048	0,282	0,070
C_2	0,286	0,481	0,638	0,381	1,786	0,446
C_3	0,357	0,159	0,213	0,381	1,11	0,278
C_4	0,286	0,240	0,106	0,190	0,822	0,206

Tabela 22. Izračunavanje sume reda i težine

Korak 5: Izračunavanje vektora prioriteta koristeći relaciju $V_i = A \times W_i$ za $i = 1, 2, \dots, n$.

Kriterijum	C_1	C_2	C_3	C_4	W	$P.Vector$
C_1	0,071	0,120	0,043	0,048	0,070	0,2886
C_2	0,286	0,481	0,638	0,381	0,446	1,972
C_3	0,357	0,159	0,213	0,381	0,278	1,187
C_4	0,286	0,240	0,106	0,190	0,206	0,848

Tabela 23. Izračunavanje vektora prioriteta

Korak 6: Izračunavanje $\lambda_i = \frac{V_i}{W_i}$ i izračunavanje λ_{max} (max. eigenvalue) na taj način što se nađe srednja vrednost λ_i iz sledeće tabele.

Kriterijum	C_1	C_2	C_3	C_4	W	$P.Vector$	Lamda	λ_{max}
C_1	1	0,25	0,20	0,25	0,070	0,2886	4,1228	4,2326
C_2	4	1	3	2	0,446	1,972	4,4215	
C_3	5	0,33	1	2	0,278	1,187	4,2698	
C_4	4	0,50	0,50	1	0,206	0,848	4,1165	
Suma	14	2,08	4,7	5,25	1	4,2956		

Tabela 24. Izračunavanje λ_i i λ_{max}

Korak 7: Izračunavanje indeksa konzistentnosti (CI) i odnosa konzistentnosti (CR).

$$C.I. = \frac{\lambda_{max} - n}{n - 1} = 0,0775$$

$$C.R. = \frac{C.I.}{R.I.} = 0,0691$$

Dobijena vrednost za CR je manja od predviđene vrednosti 0,10. Sledi, konzistentnost matrice procena je unutar prihvatljivih granica tolerancije. U slučaju da je odnos konzistentnosti veći od 0,10 potrebna je revizija subjektivne procene.

U drugoj koloni su prikazane težinske vrednosti za nastavnike dobijene na prethodno opisan način. U trećoj koloni su korišćene težine za nastavnike koji su u menadžmentu. Na osnovu iskustva primene težina iz druge kolone, promenjene su određene vrednosti težina za nastavnike koji su u menadžmentu.

Težine	Nastavnici koji nisu u menadžmentu	Nastavnici koji su u menadžmentu		
p_1	0,1	0,2		
p_2	0,4	0,3		
p_3	0,3	0,2		
p_4	0,2	0,3		

Tabela 25. Koeficijenti težine koji odgovaraju komponentama ukupne ocene O

Ukupnu ocenu dobijamo koristeći formulu (13):

$$O = \frac{p_1 O_1 + p_2 O_2 + p_3 O_3 + p_4 O_4}{p_1 + p_2 + p_3 + p_4} = 0,1 * 3 + 0,4 * 4,8 + 0,3 * 4,5 + 0,2 * 5 = 4,57,$$

gde su težinski koeficijenti uzeti iz kolone 2, tabele 25 pošto predmetni nastavnik ne pripada menadžmentu.

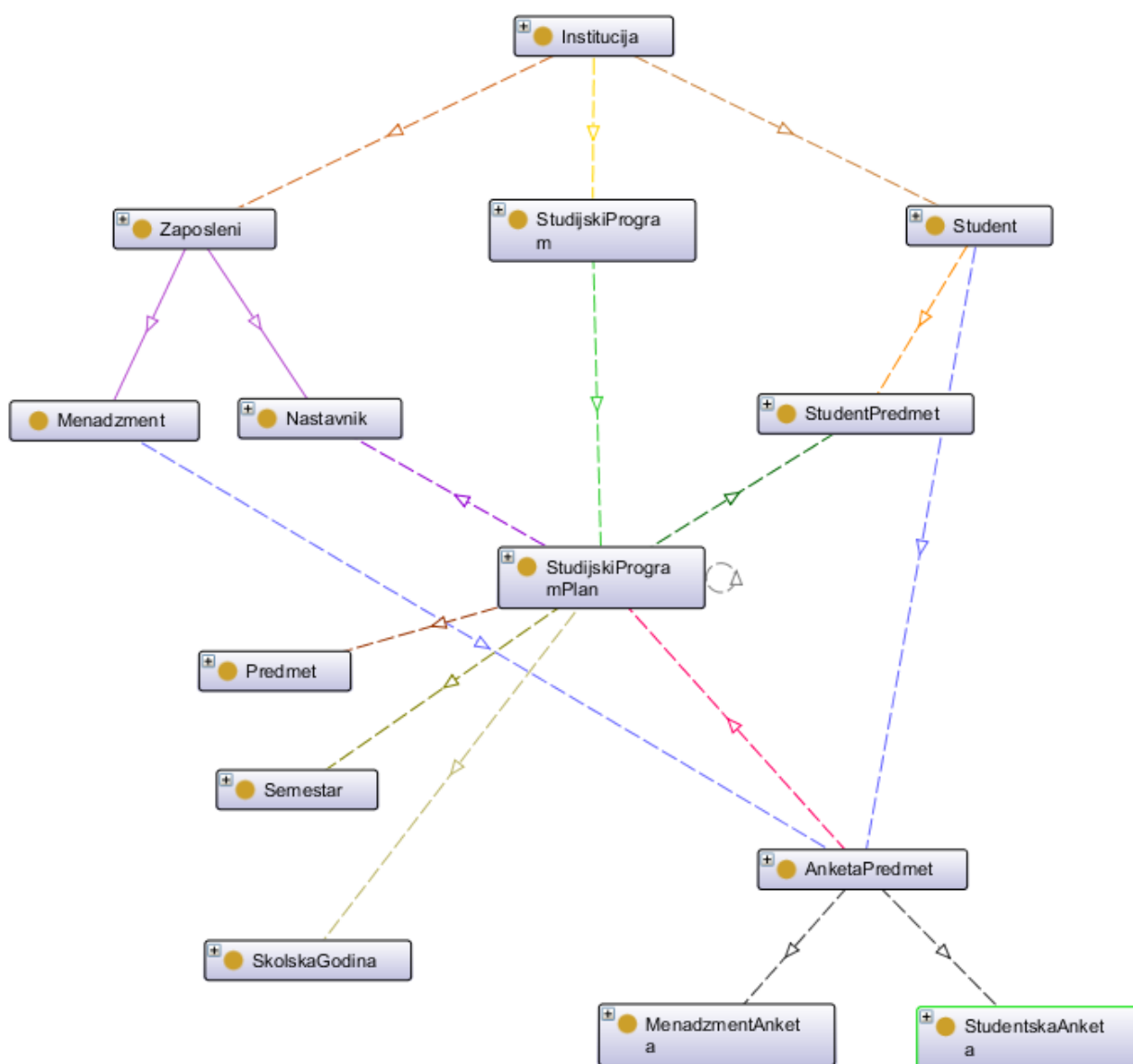
Ova ocena definiše i poverenje (trust) u nastavnika samo za taj predmet i za tu školsku godinu. Koristeći ove rezultate menadžment visokoškolske ustanove može donositi različite odluke o datom nastavniku kao što su: dalje angažovanje (ili ne), napredovanje, izbor, ...

U sledećem poglavlju dati su detalji realizacije (ontologija, upitnici, ...) prethodno opisanog primera, koristeći softverski paket za određivanje poverenja koji je razvijen na Visokoj poslovnoj školi strukovnih studija u Blacu.

13. ONTOLOGIJA VISOKOŠKOLSKE INSTITUCIJE

U ovom poglavlju uz korišćenje ontologije za predstavljanje visokoškolske ustanove skupom koncepata, dati su i upitnici za primer opisan u poglavlju 12.

Koristeći alat Protégé za OWL definisana je ontologija za visokoškolske institucije koja prikazana na slici 55.



Sl. 55. Ontologija visokoškolske institucije

Ontologija visokoškolske institucije obuhvata sledeće klase:

- Institucija;
- Zaposleni;
 - Menadžment;

- Nastavnik;
- Studijski program;
- Student;
- StudijskiProgramPlan;
- Predmet;
- Semestar;
- SkolskaGodina;
- StudentPredmet;
- AnketaPredmet;
- MenadžmentAnketa;
 - StudentskaAnekta;
- Anekta;
- Pitanje;
- Odgovor;
- OdgovorUcesnika.

Ove klase označavaju prirodne koncepte koji su zajednički ili bar razumljivi korisnicima upoznatim sa obrazovnim institucijama. Klase imaju skup instanci (objekti klase). Na primer VPSSS je instanca klase Institucija.

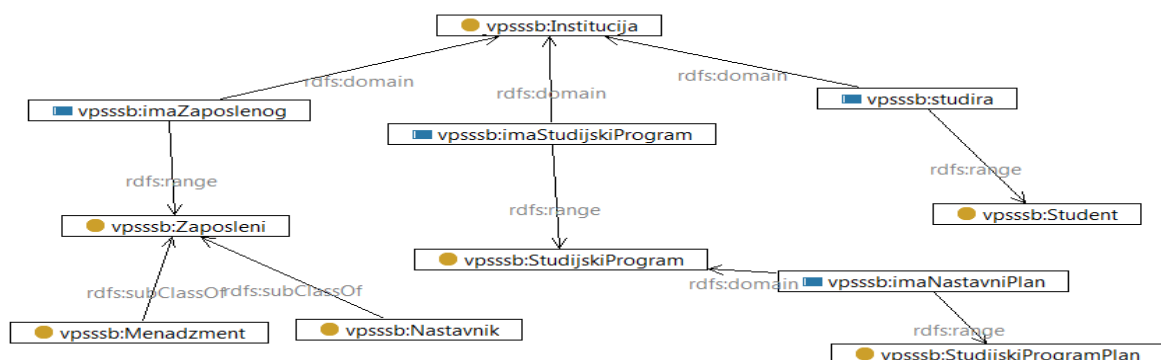
Ontologija takođe sadrži i relacije koje označavaju prirodne odnose između objekata u realnom svetu:

- imaNastavniPlan;
- imaOdgovore;
- imaStudijskiProgram;
- imaZaposlenog;
- jeDaoOdgovor;
- jeUcestvovaoUAnketi;
- naPitanje;
- nastavniPredmet;
- odgovorJe;
- predaje;
- sadrziPitanje;
- skolskaGodina;

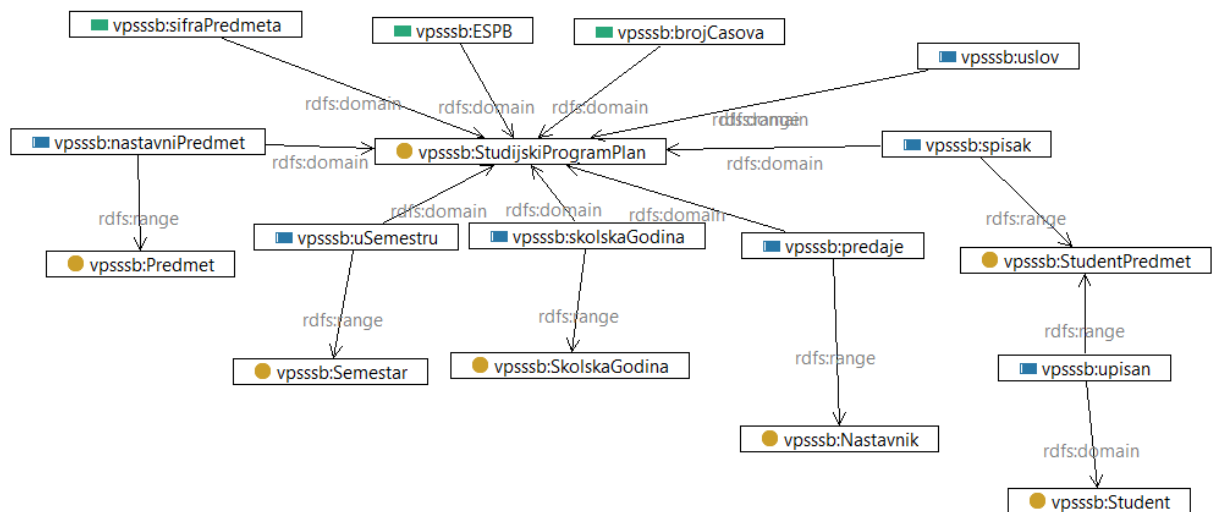
- slusa;
- spisak;
- studira;
- uAnketi;
- upisan;
- uSemestru;
- uslov;
- zaAnketu;
- zaPredmet;
- brojCasova;
- ESPB;
- ime;
- je;
- naziv;
- ocena;
- prezime;
- sifraPredmeta;
- tekstOdgovora;
- tekstPitanja.

Relacije takođe imaju instance, na primer imaStudijskiProgram(VPSSS, RiI), je instanca imaStudijskiProgram koja ima značenje da VPSSS (Visoka poslovna škola strukovnih studija Blace) ima studijski program RiI (Računarstvo i Informatika).

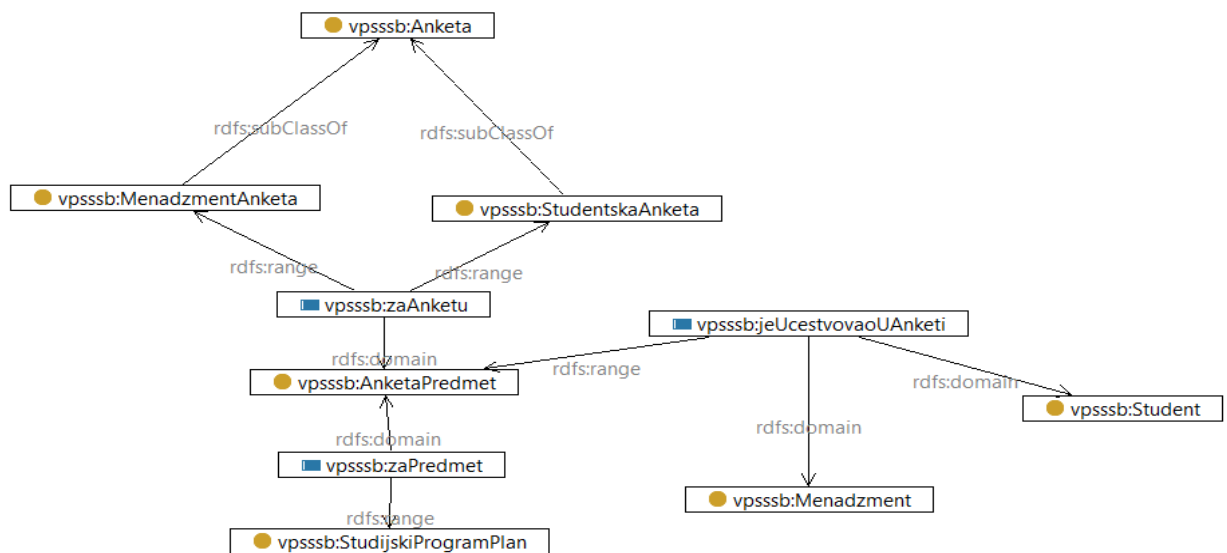
Na slikama 56, 57 i 58 prikazane su klase i relacije za instituciju, studijski program i ankete.



Sl. 56. Deo ontologije institucije sa klasama i relacijama

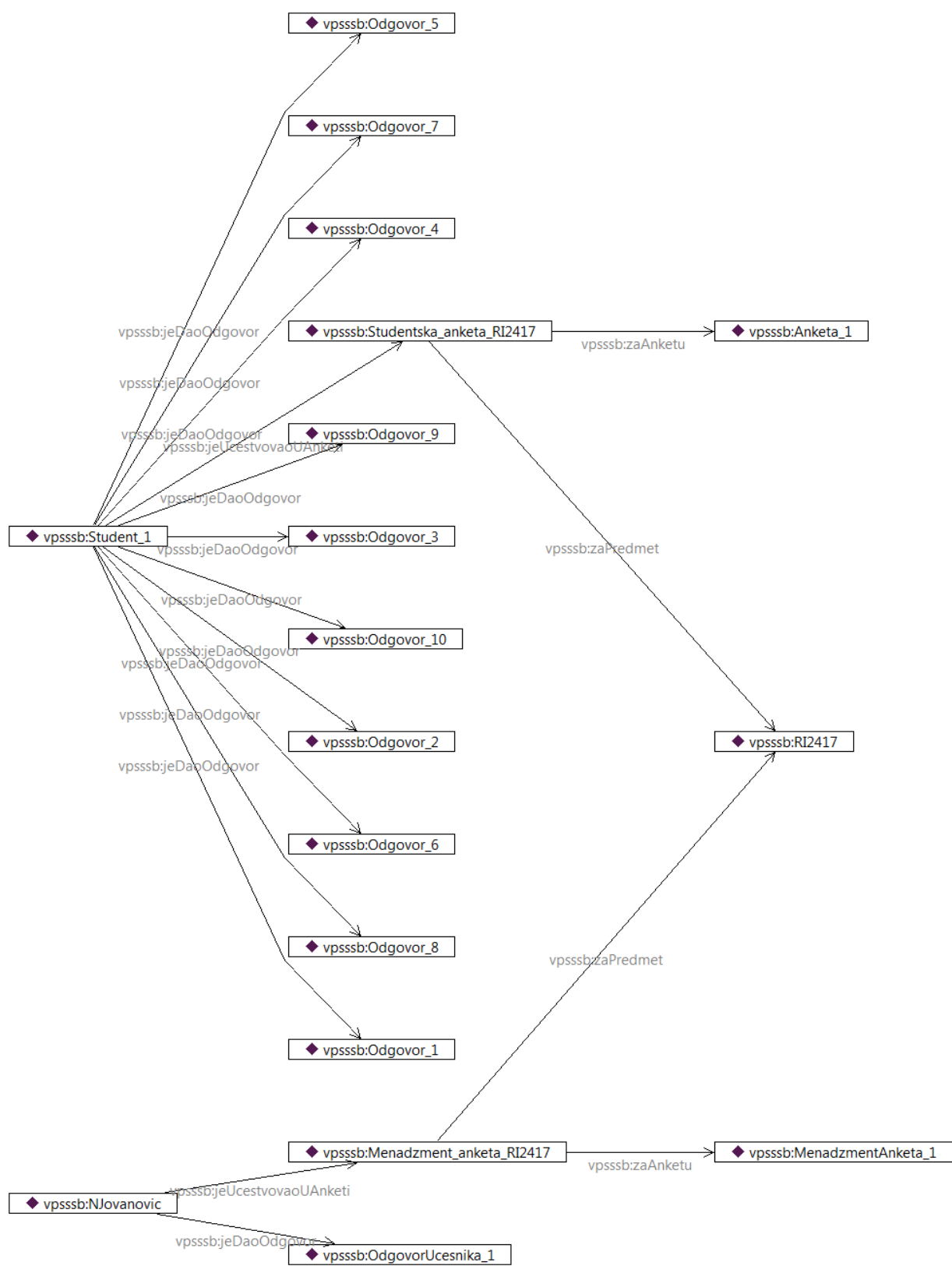


Sl. 57. Klase i relacije plana studijskog programa



Sl. 58 Klase i relacije vezane za ankete

Na slici 59 prikazane su instance prethodno definisanih klasa (institucija, studijski program, nastavnici, menadžment, studenti, predmeti, ...), dok su na slici 60 prikazane instance svih anketa.



Sl. 60. Instance učesnika i anketa

Na slikama 61, 62 i 63 prikazani su upitnici:

- Upitnik za vrednovanje pedagoškog rada nastavnika i saradnika (popunjavaju studenti);
- Upitnik za vrednovanje naučnog (studijskog) rada nastavnika (popunjava menadžment);
- Upitnik za unos prosečne prolaznosti studenata na ispitu (popunjava menadžment).

 АНКЕТЕ
СТУДЕНТИ

Упитник за вредновање педагошког рада наставника и сарадника

Приликом попуњавања упитника студент не сме нарушавати достојанство наставника / сарадника, чији рад је предмет оцењивања.

Датум: 13.12.2012

Пол: ☒ М ☐ Ж

Начин студирања: ☒ Буџет ☐ Самофинансирање

Студијски програм: Рачунарство и информатика

Година студија: Друга

Предмет: Јава програмирање

Наставник: Иван Станковић

Предавања из овог предмета посећујем: ☒ РЕДОВНО ☐ УГЛАВНОМ РЕДОВНО ☐ НЕРЕДОВНО ☐ РЕТКО ☐ НИКАДА

1. Да ли Вас је наставник на почетку семестра упознао са планом и програмом рада? ☒ ДА ☐ НЕ

2. Да ли Вас је наставник упознао са вредновањем предиспитних обавеза? ☒ ДА ☐ НЕ

3. Да ли је наставник предочио потребни литературу? ☒ ДА ☐ НЕ

4. Начин извођења наставе? ☒ Интерактивна настава ☐ Презентације ☐ Диктирање

5. У којој мери наставник излаже јасно и разумљиво? ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

6. У којој мери је наставник припремљен за час? ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

7. У којој мери наставник подстиче укључивање студената у наставу? ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

8. У којој мери је коректан однос наставника према студентима? ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

9. У којој мери је наставник доступан за консултације? ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

ПОДНЕСИ АНКЕТУ

Sl. 61. Upitnik za vrednovanje pedagoškog rada nastavnika i saradnika

АНКЕТЕ

МЕНАЏМЕНТ

ПРОЈЕКТ

ИЗВЕШТАЈИ

Упитник за вредновање научног (студијског) рада наставника

Датум:

13.12.2012

Студијски програм:

Рачунарство и информатика

Година студија:

Друга

Предмет:

Јава програмирање

Наставник:

Иван Станковић

1. Како оцењујете научни (студијски) рад?

☒

 1

☐

 2

☐

 3

☐

 4

☐

 5

ПОДНЕСИ АНКЕТУ

Sl. 62. Upitnik za vrednovanje naučnog (studijskog) rada nastavnika

АНКЕТЕ

МЕНАЏМЕНТ

ПРОЛАЗНОСТ

ИЗВЕШТАЈИ

Пролазност студената на испиту

Датум:

13.12.2012

Студијски програм:

Рачунарство и информатика

Година студија:

Друга

Предмет:

Јава програмирање

Наставник:

Иван Станковић

Пролазност:

%

ПОДНЕСИ ПРОЛАЗНОСТ

Sl. 63. Prolaznost studenata na ispitu

Na poslednjoj slici prikazan je izveštaj koji sadrži sve komponente ocene kao i vrednost konačne ocene.

 АНКЕТЕ МЕНАЏМЕНТ ПРОЛАЗНОСТ ИЗВЕШТАЈИ							
Оцењивање рада наставника							
#	Наставник	Предмет	Улазна оцена	Оцена коју су дали студенти	Оцена научног (студијског) рада	Пролазност	Коначна оцена
1.	Иван Станковић	Јава програмирање	3	4,8	4,5	5	4,57

Sl. 64. Konačna ocena i njene komponente

Postoji veliki broj radova u kojima se koristi višekriterijumsko odlučivanje pri rešavanju različitih problema kao što su: analiza i merenje pouzdanosti sadržaja različitih dokumenata [79], izračunavanje pouzdanosti servisa [80], planiranje putanja uzimajući u obzir različite zahteve vozača [81], prikupljanje podataka kod bežičnih senzorskih mreža zasnovano na bezbednosti i poverenju [82], konstrukcija socijalnih mreža [83], hibridni sistem preporuka za G2B personalizovane e-servise [84], korišćenje racionalnih i objektivnih procena pri izračunavanju poverenja u edukaciji [85].

U ovom primeru je detaljno objašnjeno korišćenje višekriterijumskog odlučivanja pri definisanju težinskih koeficijenata koji se koriste za izračunavanje poverenja (ocena) u nastavnika jedne visokoškolske ustanove. Ova ocena nastavnika se može iskoristiti pri procesu povere predmeta za sledeću školsku godinu kao i pri izboru nastavnika u nova nastavnička zvanja (vanredni, redovni profesori), pri ocenjivanju studenata, razvoju karijere studenata, određivanja stipendija, usavršavanju, ...

Težinski koeficijenti (poverenje) se mogu iskoristiti i pri različitom online anketiranju, koristeći sličnost između korisnika, familijarnost između korisnika i društvenu reputaciju korisnika.

14. ZAKLJUČAK

U ovom radu je definisano poverenje koristeći semantički web i višekriterijumsko odlučivanje i dizajniran je edukacioni model za tipičnu visokoškolsku ustanovu.

U prvom delu rada dato je poređenje postojećeg weba i vizije semantičkog weba, kao i pregled standarda i alata neophodnih za realizaciju semantičkog weba. Dat je pregled i analiza dosadašnjih istraživanja iz oblasti edukacije i semantičkog weba, postojećih softverskih implementacija u edukacionom sistemu. Data je analiza nedostataka postojećih modela i sistema. Predloženo je više sigurnosnih ontologija koje omogućuju visok nivo sigurnosti web resursa, servisa i agenata. Dati su primeri sigurnosnih ontologija koje su dizajnirane korišćenjem dobro poznatih sigurnosnih tehnika. Opisani su osnovni pojmovi iz teorije grafova i objašnjena je primena u oblasti računarstva i informatike. Semantičke tehnologije su realnost, koje se stalno usavršavaju, i čijom se primenom razvijaju kvalitetnije aplikacije.

U drugom delu rada je prikazan koncept poverenja. Poverenje je neophodan preduslov za saradnju dva agenta. Prikazani su algoritmi dizajnirani za izračunavanje poverenja koji su prosti i konceptualno dokazivi. Složenost algoritama je definisana u zavisnosti od veličina grafova. Uzeti su u obzir i problemi društvenih mreža. Data su nova matematička rešenja ugrađena u algoritme određivanja poverenja.

U trećem delu rada su prikazani osnovni pojmovi iz teorije odlučivanja i višekriterijumskog odlučivanja. Vrlo je važno pravilno definisati težine kriterijuma. Ovi momenti su u nadležnosti neposrednog donosioca odluke. Analitičar koji pomaže donosiocu odluke mora u svakom trenutku formiranja višekriterijumskog modela da prati proces tog formiranja i da ukazuje na eventualne propuste i nedostatke, u cilju donošenja najbolje odluke. Kreiran je model edukacionog okruženja na semantičkom webu. Opisana je ontologija visokoškolske obrazovne institucije i detaljno je prikazano rešenje izračunavanja poverenja za nastavni kadar u edukacionom modelu. Realizovan je softverski paket za različita anketiranja u edukacionom sistemu baziran na osnovu algoritama koji su nastali kao rezultat dugogodišnjeg iskustva u oblasti edukacije. Predloženi su i novi načini korišćenja realizovanog rešenja u različitim softverskim modelima.

Budući rad se zasniva na automatizaciji svih faza u razvoju edukacionog modela koristeći veštačku inteligenciju kao i na generalizaciji, standardizaciji i dobijanju opšteg modela koji će biti primenjiv u različitim naučnim oblastima.

15. LITERATURA

- [1] Golbeck, J. & Hendler, J. & Parsia, B. *Trust networks on the Semantic Web*. In *Proceedings of Cooperative Intelligent Agents 2003 (Helsinki, Finland)*. University of Maryland. <http://www.mindswap.org/papers/CIA03.pdf>.
- [2] Broder, R. & Kumar, F. & Maghoul, P. & Raghavan, S. & Rajagopalan, R. & Stata, A. Tomkins, J. & Wiener. 2000. *Graph structure in the web*. In *Proc. 9th International World Wide Web Conference*.
- [3] Moore, C. & Newman, M. E. J. Epidemics and percolation in small-world networks. 2000. In *Physical Review E* 61, 5678–5682.
- [4] Dumbill, E. 2002. *XML Watch: Finding friends with XML and RDF*. In *IBM Developer Works*. <http://www-106.ibm.com/developerworks/xml/library/x-foaf.html>.
- [5] Golbeck, J. *The trust ontology*. <http://trust.mindswap.org/trustOnt.shtml>.
- [6] RDFWeb. *Co-depiction Photo Meta Data*. <http://rdfweb.org/2002/01/photo/>.
- [7] *Adding SVG Paths to Co-Depiction RDF*. <http://Jibbering.com/svg/codepiction.html>.
- [8] FOAFNaut. <http://foafnaut.org/>.
- [9] Ngo, L. & Haddawy, P. 1997. Answering queries from context-sensitive probabilistic knowledge bases. In *Theoretical Computer Science*, 171, 147-177.
- [10] French, S. 1985. Group consensus probability distributions: A critical survey. In J. M. Bernardo, M. H. DeGroot, D. V. Lindley and A. F. M. Smith (Eds.), *Bayesian statistics 2*, 183-202. Amsterdam, Netherlands: Elsevier.
- [11] Genest, C. & Zidek, J. V. 1986. Combining probability distributions: A critique and an annotated bibliography. In *Statistical Science*, 1, 114-148.
- [12] Pearl, J. 1988. Probabilistic reasoning in intelligent systems: Networks of plausible inference. In San Francisco, CA: Morgan Kaufmann.
- [13] Agresti, A. 1990. Categorical data analysis. In New York, NY: Wiley.
- [14] Richardson, M. & Agrawal, R. & Domingos, P. 2003. Trust Management for the Semantic Web. In *Proceedings of the Second International Semantic Web Conference*.
- [15] Agrawal, R. & Dar, S. & Jagadish, H. V. 1990. Direct transitive closure algorithms: Design and performance evaluation. In *ACM Transactions on Database Systems*, 15, 427-458.
- [16] Bellman, R. & Giertz, M. 1973. On the analytic formalism of the theory of fuzzy sets. In *Information Sciences*, 5, 149-156.
- [17] Bancilhon, F. 1985. Naive evaluation of recursively defined relations. In *Knowledge Base Management Systems (Islamorada)* (pp. 165-178).
- [18] Agrawal, R. & Jagadish, H. V. 1988. Multiprocessor transitive closure algorithms. In *Proceedings of the International Symposium on Databases in Parallel and Distributed Systems* (pp. 56-66). Austin, TX.
- [19] Warshall, S. 1962. A theorem on boolean matrices. In *Journal of the ACM*, 9, 11-12.

- [20] Page, L. & Brin, S. & Motwani, R. & Winograd, T. 1999. *The PageRank citation ranking: Bringing order to the Web*. In *Tech. Rep. SIDL-WP-1999-0120*. Stanford University. <http://dbpubs.stanford.edu/pub/1999-66>.
- [21] Motwani, R. & Raghavan, P. 1995. Randomized algorithms. In Cambridge University Press.
- [22] Richardson, M. & Domingos, P. 2002. The intelligent surfer: Probabilistic combination of link and content information in PageRank. In T. G. Dietterich, S. Becker and Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems 14*, 1441-1448. Cambridge, MA: MIT Press.
- [23] Barners-Lee, T. & Hendler, J. & Lassila, O. 2001. The Semantic Web. *Scientific American*.
- [24] Van Harmelen, F. & Fensel, D. 1999. Practical Knowledge Representation for the Web. *IJCAI'99 Workshop on Intelligent Information Integration*.
- [25] Antoniou, G. & Van Harmelen, F. 2008. The Semantic Web Vision. In *A Semantic Web Primer second edition*, 1-3. The MIT Press, ISBN: 978-0-262-01242-3.
- [26] Schaffert, S. 2006. *IkeWiki: A SemanticWiki for Collaborative Knowledge Management*. 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'06).
- [27] Della Valle, E. & Ceri, S. & Van Harmelen, F. & Fensel, D. 2009. *It's a Streaming World! Reasoning upon Rapidly Changing Information*. IEEE Intelligent Systems.
- [28] Hendler, J. 2001. *Agents and the Semantic Web*. IEEE Intelligent Systems.
- [29] Segaran, T. & Evans, C. & Taylor, J. 2009. Semantic Data. In *Programming the Semantic Web*, 5-18. O'Reilly Media, ISBN: 978-0-596-15381-6.
- [30] Tiropanis, T. & Davis, H. & Millard, D. & Weal, M. *Semantic Technologies for Learning and Teaching in the Web 2.0 Era*. IEEE Intelligent Systems.
- [31] Segaran, T. & Evans, C. & Taylor, J. 2009. Expressing Meaning. In *Programming the Semantic Web*, 19-22. O'Reilly Media, ISBN: 978-0-596-15381-6.
- [32] Euler, L. 1735. *Reading from the web*. Seven Bridges of Königsberg. http://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg (Visited 2009).
- [33] Guthrie, F. 1852. *Reading from the web*. Four color theorem. http://en.wikipedia.org/wiki/Four_color_theorem (Visited 2009).
- [34] Barners-Lee, T. 2000. Semantic Web on XML. XML 2000 Conference. Washington DC
- [35] Barners-Lee, T. 2005. Web for real people. W3C Talks.
- [36] Segaran, T. & Evans, C. & Taylor, J. 2009. Standards and Sources. In *Programming the Semantic Web*, 63-94. O'Reilly Media, ISBN: 978-0-596-15381-6.
- [37] Segaran, T. & Evans, C. & Taylor, J. 2009. Ontology. In *Programming the Semantic Web*, 127-145. O'Reilly Media, ISBN: 978-0-596-15381-6.
- [38] Stanford Center for Biomedical Informatics Research. 1997. *Reading from the web*. Protégé. <http://protege.stanford.edu/> (Visited 2009).

- [39] Hobbs, J. & Lassila, O. & McDermott, D. & McIlraith, S. & Narayanan, S. & Paolucci, M. & Parsia, B. & Payne, T. & Sirin, E. & Srinivasan, N. & Sycara, K. 2004. *Reading from the web*. OWL-S: Semantic Markup for Web Services. <http://www.w3.org/Submission/OWL-S/> (Visited 2009).
- [40] Martin, D. & Denker, G. 2004. *Reading from the web*. DAML Services - Security and Privacy. <http://www.daml.org/services/owl-s/security.html> (Vsited 2009).
- [41] Fogg, B. & Tseng, S. 1999. The elements of computer credibility. In *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems* (New York, NY, USA), vol. 1, ACM Press, pp. 80–87. <http://captology.stanford.edu/pdf/p80-fogg.pdf>.
- [42] Gambetta, D. 2000. Can we trust trust?. In *Trust: Making and Breaking Cooperative Relations* (electronic edition). <http://www.sociology.ox.ac.uk/papers/gambetta213-237.pdf>.
- [43] Axelrod, R. 1984. *The Evolution of Cooperation*. Basic Books
- [44] Wiktionary. 2006. Definition: credibility, <http://en.wiktionary.org/w/index.php?title=credibility&oldid=1177659>.
- [45] Fogg, B. & Soohoo, C. & Danielson, D. & Marble, L. & Stanford, J. & Tauber, E. R. 2002. *How do people evaluate a Web site's credibility?*. <http://www.consumerWebwatch.org/dynamic/Web-credibility-report-evaluate.cfm>.
- [46] Golbeck, J. 2005. *Computing and Applying Trust in Web-Based Social Networks*. PhD thesis, University of Maryland, College Park.
- [47] Fogg, B. & Marshall, J. & Laraki, O. & Osipovich, A. & Varma, C. & Fang, N. & Paul, J. & Rangnekar, A. & Shon, J. & Swani, P. & Treien, M. 2001. *What makes a Web site credible? A report on a large quantitative study*. In *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems* (New York, NY, USA), vol. 1. ACM Press, pp. 61–68.
- [48] Fogg, B. J. 2002. *Stanford guidelines for Web credibility*. <http://www.Webcredibility.org/guidelines/>.
- [49] Abdul-Rahman, A. & Hailes, S. 1997. A distributed trust model. In *NSPW '97: Proceedings of the 1997 workshop on New security paradigms* (New York, NY, USA), ACM Press, pp. 48–60. <http://doi.acm.org/10.1145/283699.283739>.
- [50] Josang, A. 1996. The right type of trust for distributed systems. In *Proceedings of the 1996 New Security Paradigms Workshop*, C. Meadows, Ed., ACM. <http://sky.fit.qut.edu.au/~josang/papers/Jos1996-NSPW.pdf>.
- [51] Abdul-Rahman, A. & Hailes, S. 1997. Using recommendations for managing trust in distributed systems. In *IEEE Malaysia International Conference on Communication*. <http://www.cs.ucl.ac.uk/staff/F.AbdulRahman/docs/micc97.ps>.
- [52] Resnick, P. & Varian, H. R. 1997. *Recommender systems*. In *Communications of the ACM* (New York, NY, USA), vol. 40. ACM Press, pp. 56–58. <http://doi.acm.org/10.1145/245108.245121>.
- [53] Goldberg, D. & Nichols, D. & Oki, B. M. & Terry, D. 1992. *Using collaborative filtering to weave an information tapestry*. In *Communications of the ACM* 35, 12, 61–70. <http://doi.acm.org/10.1145/138859.138867>.

- [54] YAHOO! INC. *Bloglines*. <http://www.bloglines.com/>.
- [55] LAST.FM LTD. *Last.fm – the social music revolution*. <http://www.last.fm/>.
- [56] Massa, P. & Bhattacharjee, B. 2004. *Using Trust in Recommender Systems: An Experimental Analysis*. In *Proceedings of iTrust 2004*. <http://doi.acm.org/10.1145/223904.223929>.
- [57] Massa, P. & Avesani, P. 2004. *Trust-aware collaborative filtering for recommender systems*. In *CoopIS/DOA/ODBASE*, pp. 492–508. <http://dblp.uni-trier.de/db/conf/coopis/coopis2004-1.html#MassaA04>.
- [58] O'Donovan, J. & Smyth, B. 2004. Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces*, J. Riedl, A. Jameson, D. Billsus, and T. Lau, Eds., pp. 167–174. <http://doi.acm.org/10.1145/1040830.1040870>.
- [59] Canny, J. 2002. Collaborative filtering with privacy. In *SP '02: Proceedings of the 2002 IEEE Symposium on Security and Privacy (Washington, DC, USA)*, IEEE Computer Society, p. 45.
- [60] Yao, W. T. M. 2003. *Trust Management for Widely Distributed Systems*. PhD thesis, Jesus College, University of Cambridge, <http://www.cl.cam.ac.uk/Research/SRG/opera/publications/Theses/wtmy2.pdf>.
- [61] Housley, R. & Ford, W. & Polk, W. & Solo, D. 1999. *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*. In *RFC 2459 (Proposed Standard)*. <http://www.ietf.org/rfc/rfc2459.txt>.
- [62] Zimmerman, P. R. 1995. *The official PGP user's guide*.
- [63] Levien, R. 1998. *Attack-resistant trust metrics for public key certification*. In *Proceedings of the 7th USENIX Security Symposium*, pp. 229-242. <https://www.usenix.org/publications/library/proceedings/sec98/fullpapers/levien/levienhtml/levien.html>.
- [64] Levien, R. 2000. *Advogato's trust metric*. <http://www.advogato.org/trust-metric.html>.
- [65] Levien, R. & Aiken, A. 2000. *An attack-resistant, scaleable name service*. <http://www.levien.com/fc.ps>.
- [66] Golbeck, J. & Hendler, J. 2004. Accuracy of metrics for inferring trust and reputation in Semantic Web-based social networks. In *Proceedings of EKAW'04*. <http://www.mindswap.org/papers/GolbeckEKAW04.pdf>.
- [67] Golbeck, J. & Hendler, J. 2004. Inferring reputation on the Semantic Web. In *Proceedings of the Thirteenth International World Wide Web Conference*. <http://www.mindswap.org/papers/GolbeckWWW04.pdf>.
- [68] Golbeck, J. & Hendler, J. 2004. Reputation network analysis for email filtering. In *Proceedings of the First Conference on Email and Anti-Spam*. <http://www.mindswap.org/papers/Email04.pdf>.
- [69] Brickley, D. & Miller, L. 2005. *The Friend of a Friend vocabulary*. <http://xmlns.com/foaf/0.1/>.
- [70] Stanković, R. 2004. Prilog razvoju matematičkih metoda u poslovnom odlučivanju. Fakultet za menadžment, Novi Sad.

- [71] Harrison, F. 1987. In *The Managerial Decision – Making Process*. HMH, ISBN: 9780395908211
- [72] Klein, M. & Methlie, L. B. 1995. *Knowledge-based decision support systems: With applications in business*. Wiley, ISBN: 0471952958
- [73] Greenberg, J. & Baron, A. R. 2008. *Behavior in Organizations*, 9th Edition. Prentice Hall, ISBN: 978-0-13-154284-6
- [74] Wedley, W. C. & Field, R. H. G. 1984. A Predecision Support System. *Acad. Management Rev.*, 9, 4, 696-703
- [75] Wedley, W. C. & Field, R. H. G. 1984. A Predecision Support System. *Acad. Management Rev.*, 9, 4, 696-703
- [76] Herbert, S. A. 1955. *Biometrika*, Vol. 42, No. $\frac{3}{4}$, pp. 425-440
- [77] Lazic, J. & Zizovic, M. 2006. Menadžment univerziteta i Bolonjska deklaracija, SYMOPIS 2006, Banja Koviljaca, 349-351
- [78] Cvijanovic, J.M. & Zizovic, M. & Lazic, J. 2007. Bolonjska deklaracija i menadžment u visokoškolskim ustanovama, *INDUSTRIJA* 1/2007, 113-120.
- [79] Bong, CW. & Holtby, D.W. & Ng, K.S. 2012. Fuzzy Multicriteria Decision Analysis for Measurement of Document Content Reliability. 5th International Symposium on Computational Intelligence and Design (ISCID).
- [80] Wu, Y & Yan, C. & Ding, Z. & Liu, G. & Wang, P. & Jiang, C. & Zhou, M. 2013. A Novel Method for Calculating Service Reputation. *IEEE Transactions on Automation Science and Engineering*.
- [81] Bozkurt, S. & Yazici, A. & Keskin, K. 2012. A multicriteria route planning approach considering driver preferences. *IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*.
- [82] Stelte, B. & Matheus, A. 2011. Secure trust reputation with multi-criteria decision making for wireless sensor networks data aggregation. *2011 IEEE Sensors*
- [83] Yin, G. & Jiang, F. & Cheng, S. & Li, X. & He, X. 2012. AUTrust: A Practical Trust Measurement for Adjacent Users in Social Networks. *2012 Second International Conference on Cloud and Green Computing (CGC)*.
- [84] Shambour, Q. & Jie, L. 2010. A Framework of Hybrid Recommendation System for Government-to-Business Personalized E-Services. *2010 Seventh International Conference on Information Technology: New Generations (ITNG)*.
- [85] Imazawa, A. & Naoe, N. 2010. Trust literacy and education. *2010 2nd International Congress on Engineering Education (ICEED)*.
- [86] Aaron, G. & Rasmussen, R. 2011. *Global Phishing Survey*. In *APWG*. http://www.antiphishing.org/reports/APWG_GlobalPhishingSurvey_2H2010.pdf
- [87] Saaty, T.L. 1980. *The Analytic Hierarchy Process*. In McGraw-Hill International, New York, NY, U.S.A.
- [88] Miller, G.A. 1956. The Magical Number Seven Plus or Minus Two: Some Limits on Our

Capacity for Processing Information. In Psychological Review, 63: 81-97.

[89] Stanković I. & Milovanović I. & Milovanović E. & Mirković T. 2013. A Note On Chebyshev Inequality for (P, Q) – Convex Sequences. In Metalurgia International Vol. XVIII, No. 6. pp. 116-117. ISSN 1582-2214

[90] Mirković T. & Stanković I. 2013. Some Inequalities for Graph Eigenvalues. In Metalurgia International Vol. XVIII, No. 6. pp. 77-78. ISSN 1582-2214

16. SPISAK SLIKA I TABELA

Slika 1: Mreža poverenja

Slika 2: TrustMail sa ocenom poverenja

Slika 3: Algebra putanje spajanje uverenje – primer weba poverenja

Slika 4: Jaka i slaba invarijantnost

Slika 5: Inteligenti personalni agenti

Slika 6: Jednostavna šema podataka o restoranima

Slika 7: Relacioni podaci o restoranu

Slika 8: Rezultat pretrage

Slika 9: Veza barova na postojeću šemu

Slika 10: Normalizovana šema koja odvaja ugostiteljski objekat od njegove svrhe

Slika 11: Primer velike šeme

Slika 12: Šema ugostiteljskih objekata sa kompletno prilagođenim svojstvima

Slika 13: Podaci o ugostiteljskim objektima u fleksibilnoj formi

Slika 14: Dodavanje novih svojstva ugostiteljskom objektu bez menjanja šeme

Slika 15: Svojstva ugostiteljskog objekta

Slika 16: Parametrizovani prikaz ugostiteljskih objekata

Slika 17: Dijagram rečenice koji prikazuje subjekat-predikat-objekat odnos

Slika 18: Graf tripleta koji prikazuje informacije o ugostiteljskom objektu

Slika 19: Triple koji opisuje naziv filma Blade Runner

Slika 20: Graf koji opisuje film Blade Runner

Slika 21: Skupovi čvorova i grana

Slika 22: Skupovi čvorova i grana

Slika 23: Primer crteža grafa

Slika 24: Primer orijentisanog grafa

Slika 25: Primer orijentisanog grafa

Slika 26: Primer kompletnog grafa

Slika 27: Primer put grafa

Slika 28: Primer ciklus grafa

Slika 29: Primer stepena grafa

Slika 30: Primer regularnog grafa

Slika 31: “layer cake” semantičkog weba

Slika 32: Alternativni “layer cake” semantičkog weba

Slika 33: Dijagram koji pokazuje odnos URL-ova i URI-a

Slika 34: Prazni čvorovi u grafu socijalne mreže

Slika 35: Korišćenje praznog čvora za modelovanje poštanske adrese

Slika 36: Ivanov FOAF graf

Slika 37: Traka Ivanovog FOAF grafa

Slika 38: Domen i opseg svojstva koje izražava boju očiju

Slika 39: Graf koji opisuje film Blade Runner

Slika 40: Hijerarhija klasa za film ontologiju

Slika 41: Definicije svojstava za film ontologiju

Slika 42: Klasa performans u film ontologiji

Slika 43: Kreiranje ontologije u Protégéu

Slika 44: Kreiranje hijerarhije klasa u Protégéu

Slika 45: Kreiranje svojstva objekta u Protégéu

Slika 46: Kreiranje svojstva podataka u Protégéu

Slika 47: Kreiranje pojedinačne instance u Protégéu

Slika 48: Loš model koji se bavi napravljen-za-TV filmove

Slika 49: Bolji model koji se bavi napravljen-za-TV filmove

Slika 50: Preduzimanje akcije (poverenje) kao rizika u fukciji cena neaktivnosti

Slika 51: Interdisciplinarni okvir odlučivanja

Slika 52: Naučni okvir odlučivanja

Slika 53: Rizičnost odluka: pregled

Slika 54: Hijerarhija određivanja ocene za nastavnika

Slika 55: Ontologija visokoškolske institucije

Slika 56: Deo ontologije institucije sa klasama i relacijama

Slika 57: Klase i relacije plana studijskog programa

Slika 58: Klase i relacije vezane za ankete

Slika 59: Instance određenih klasa

Slika 60: Instance učesnika i anketa

Slika 61: Upitnik za vrednovanje pedagoškog rada nastavnika i saradnika

Slika 62: Upitnik za vrednovanje naučnog (studijskog) rada nastavnika

Slika 63: Prolaznost studenata na ispitu

Slika 64: Konačna ocena i njene komponente

Tabela 1: Tabela restorana

Tabela 2: Previše podataka u tabeli

Tabela 3: Novi skup podataka barova

Tabela 4: Proširena tabela ugostiteljskih objekata sa lokacijom

Tabela 5: Povezivanje susednih ugostiteljskih objekata

Tabela 6: Rezultat pretrage

Tabela 7: Rezultat pretrage

Tabela 8: Rezultat pretrage

Tabela 9: Rezultat pretrage

Tabela 10: Osnovna statistika – izveštaj Anti-Phishing Working Group

Tabela 11: Semantika vrednosti poverenja

Tabela 12: Saaty-eva skala pairwise poređenja sa devet vrednosti

Tabela 13: Matrica procene

Tabela 14: RCI vrednosti za različite vrednosti n

Tabela 15: Odstupanje od ulaznih ocena

Tabela 16: Granice odstupanja i odgovarajuće ocene

Tabela 17: Procenat prolaznosti po ispitnim rokovima i prosečna prolaznost

Tabela 18: Veza između ocene O_4 i prosečne prolaznosti

Tabela 19: Matrica važnosti četiri kriterijuma odlučivanja

Tabela 20: Izračunavanje sume kolona

Tabela 21: Normalizacija vrednosti

Tabela 22: Izračunavanje sume reda i težine

Tabela 23: Izračunavanje vektora prioriteta

Tabela 24: Izračunavanje λ_i i $\lambda_{\max}$

Tabela 25: Koeficijenti težine koji odgovaraju komponentama ukupne ocene O