

SEMANTIKA PODATKOV IN ONTOLOGIJE

Dejan Lavbič, Marjan Krisper

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Tržaška 25, 1000 Ljubljana

{Dejan.Lavbic, Marjan.Krisper}@fri.uni-lj.si

Povzetek

Semantično strukturiranje, ki ga dosežemo z uporabo ontologij, se razlikuje od površinskega oblikovanja informacij, ki nam jih nudijo relacijske in XML podatkovne baze, saj mora biti pri podatkovnih bazah večina semantične vsebine zajeta v uporabniških programih. Ontologije predstavljajo objektivno specifikacijo informacij za določeno domeno, kjer s predstavitvijo konceptov in medsebojnih relacij prikažemo znanje. Ta specifikacija je prvi korak pri gradnji naprednejših informacijskih sistemov, ki temeljijo na souporabi obstoječega znanja. V prispevku prikažemo stopnjo inteligence, ki so jo podatki skozi čas pridobivali, se ustavimo ter podrobneje opišemo najvišjo raven, ontologije in avtomatsko sklepanje. Predstavljena je opredelitev ontologij s filozofskega stališča in uporaba na področju računalništva in informatike, kjer omenimo jezike za opis in predstavitev ontologij.

Abstract

DATA SEMANTICS AND ONTOLOGIES

Semantic structuring, achieved by ontologies, differs from superficial formatting of information offered by relational and XML databases, because with databases the major part of semantic content is captured in application logic. Ontologies provide an objective specification of specific domain information where the knowledge is expressed by means of concepts and relations among them. This specification is the first step in building advanced information systems that are based on reusing the existing knowledge. In the following paper we present the level of intelligence the data gained through time and describe its highest level in more detail: ontologies and automated reasoning. The definition of ontologies from the philosophical point of view is being presented as well as its practical use in the field of computer and information science, where we emphasize languages for describing and representing ontologies.

1 Uvod

Kot je na konferenci v San Franciscu leta 1999 poudaril Tim Berners Lee, idejni vodja semantičnega spleta, je prvi korak pri gradnji semantičnega spleta objava podatkov na svetovnem spletu v takšni obliki, da jih bodo računalniki razumeli [2]. Tako pridemo do semantičnega spleta, to je spleta podatkov, ki jih lahko posredno ali neposredno obdelujemo z računalniki.

Osnovna ideja Tima Berners Leeja ni bila le v dostopu do spletnih strani na strežnikih, ampak v ustvarjanju medsebojnih povezav med posameznimi podatki. Svetovni splet teh medsebojnih povezav v tem trenutku nima opredeljenih, zato se je pojavila tehnologija RDF/S,¹ ki je namenjena ravno zajemu teh povezav. Težava nastane pri poskusu opisovanja le-teh, saj potrebujemo dodatne metapodatke (zaradi samodejne računalniške obdelave vsebin), ki pa se v tem trenutku na svetovnem spletu ne nahajajo.

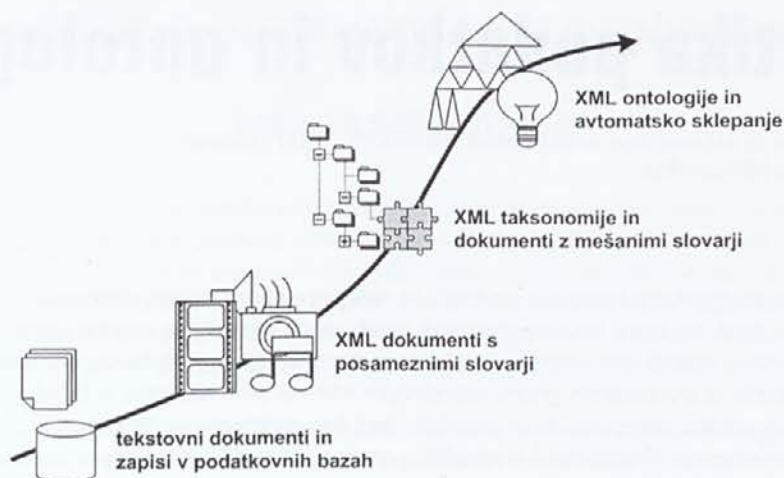
Danes se soočamo z vprašanjem, kako zgraditi splet podatkov, ki bi ga lahko računalniki samodejno obdelovali. Najprej moramo spremeniti pogled na podatke. V preteklosti so bili podatki shranjeni in zaklenjeni v zaščitene programe ter pri sami obdelavi niso igrali glavne vloge. Takšni podatki, za katere se je uveljavil izraz GIGO,² imajo veliko pomanjkljivost v odvisnosti od njihove obdelave oz. odvisnost programske opreme od dobrih podatkov. Ker so nekateri spoznali, da takšen pristop ni pravilen, so začeli razmišljati v drugo smer. Velik uspeh je doživel internet in z njim XML³ ter v zadnjih časih zelo priljubljeni semantični splet, kjer se vedno bolj osredotočamo na podatke in ne na programe. Ključni element pri samodejni računalniški obdelavi podatkov pa je priprava »pametnejših«⁴ podatkov [2].

¹ Resource Definition Framework/Schema (RDF/S) – S pomočjo jezika RDF predstavimo semantične relacije na ravni primerkov v obliki trojčkov <oseba, predikat, predmet>. Z RDF/S pa opišemo relacije na ravni koncepta ontologije, kjer določimo sprejemljive relacije za raven primerkov.

² Garbage In, Garbage Out – Računalniki, v nasprotju z večino ljudi, brez vprašanj sprejmejo nesmiselne podatke, jih obdelajo in tudi vrnejo kup nesmiselnih podatkov na izhodu, če jim tega izrecno ne prepovemo v programski kodi.

³ eXtensible Markup Language – Razširljivi označevalni jezik, ki predstavlja format podatkov za izmenjavo strukturiranih dokumentov v spletu.

⁴ Od aplikacij neodvisni podatki; lahko jih enostavno dopolnjujemo, klasificiramo in predstavljajo del večjega informacijskega ekosistema, to je ontologije.



Slika 1: Rastoča »inteligenca« v podatkih

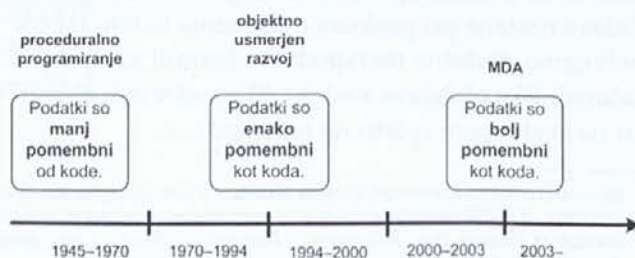
Slika 1 prikazuje stopnjo inteligence, ki so jo podatki pridobivali skozi čas. Začnemo s takšnimi, ki za nadaljnje sklepanje vsebujejo zelo malo semantičnih informacij, in končamo pri ontologijah:

- **Tekst in podatkovne baze (pred XML)**
Začetna faza, kjer so podatki pripadali določenemu programu. Inteligenca je zato bila skrita v samem programu in ne v podatkih.
- **XML dokumenti za določeno domeno**
Faza, kjer podatki v določeni problemski domeni dosežejo določeno neodvisnost od programov. Podatki so sedaj dovolj »inteligentni«, da se lahko prenašajo med različnimi programi znotraj določene problemske domene. Kot primer lahko navedemo XML standarde v zdravstvu, zavarovalništvu ali nepremičninski industriji.
- **Taksonomije⁵ in dokumenti z mešanimi slovarji**
Podatke lahko združimo iz različnih domen in jih tudi ustrezno klasificiramo v obliki hierarhične taksonomije. Pravzaprav lahko klasifikacijo uporabljamo za odkrivanje podatkov, enostavna razmerja med kategorijami in taksonomijo pa za povezovanje in združevanje podatkov. Podatki so v tej fazi dovolj »inteligentni«, da jih lahko enostavno odkrijemo in razumno združimo z ostalimi podatki.
- **Ontologije in pravila**
Na tej ravni lahko iz obstoječih podatkov izpeljemo nove s pomočjo logičnih pravil. Podatki so sedaj že na dovolj visoki ravni, da jih lahko

opišemo z dejanskimi medsebojnimi povezavami in naprednimi formalizmi, s pomočjo katerih nad to »semantično algebro« izvajamo logične izračune. Takšne podatke lahko povezujemo na bolj atomarnem nivoju in izvajamo analize podatkov z majhnimi zrni. V tem primeru podatki ne predstavljajo zgolj surovine, ampak so del nekakšnega naprednega mikrokozmosa. Primer takšne predstavitve podatkov je samodejni prevod dokumenta iz enega problemskega področja v ekvivalenten (oz. čimbolj podoben) dokument v drugi problemski domeni.

Ko govorimo o pogledu razvijalcev na evolucijo podatkov, si lahko na sliki 2 ogledamo vpliv podatkov na razvoj programske opreme [13]. Proceduralno programiranje je osredotočeno na funkcionalno dekompozicijo opravil. Podatke so spreminjale in obvladovale procedure, kjer je veljalo, da so podatki manj pomembni od kode.

Pri objektno usmerjenem razvoju se pojavijo enkapsulacija, dedovanje in polimorfizem. Ideja o varo-



Slika 2: Pogled razvijalcev na podatke

⁵ Sistematika, nauk o klasifikaciji, ki predstavlja segmentacijo ter ureditev elementov v razvrstitveni sistem glede na medsebojne povezave.

vanju podatkov s pomočjo rezerviranih besed in metod za dostop povzroči dvig ravni pomembnosti podatkov, tako da postanejo enakovredni metodam, ki s temi podatki manipulirajo. S prihodom XML-a, MDA-ja⁶ in podpore metapodatkom v času izvajanja vstopamo v obdobje, kjer podatki postajajo bolj pomembni kot sama koda. V ospredje ne prihaja zgolj sintaktična pravilnost, ampak predvsem semantična.

2 Opredelitev ontologije

Ontologijo opredeljujejo besede in njihov pomen, ki se uporabljajo za opis in predstavitev znanja določenega področja. Vendar, kaj ta opredelitev sploh pomeni? Če pogledamo v ustrezne slovarje [5, 6], najdemo naslednje opredelitve:

- filozofska disciplina, veja metafizike, ki obravnava osnovo, vzroke in najsplošnejše lastnosti stvarnosti,
 - določena teorija o naravi bitij in njihovem obstoju.
- Opredelitvi nakazujeta, da izraz ontologija izhaja iz filozofije. Ontologija je tako po eni strani filozofska disciplina, medtem ko se ontologija na področju računalništva in informatike ukvarja s predstavitvijo podatkov in znanja. Tudi za ontologijo na področju informatike poznamo več opredelitev:
- ontologijo opredeljujejo besede in njihov pomen, ki se uporabljajo za opis in predstavitev znanja določenega področja;

- ontologijo sestavljata slovarja, ki opisujeta določeno domeno in eksplicitne domneve o pomenu slovarja – specializacija konceptualizacije.⁷

Če si podrobneje pogledamo zgornji opredelitvi, ju lahko razdelimo na dva dela:

- opis in predstavitev znanja določenega področja,
- opredelitev uporabljenih besed in njihov pomen.

2.1 Opis ontologije

Opisovanja znanja določenega področja se lahko lotimo v pisni ali ustni obliki, tako da poudarimo pomembnejše elemente. Če želimo opisati potovanje z letalom, ga lahko na način, ki ga prikazuje tabela 1.

Ko opisujemo znanje določenega področja, opisujemo samo pomembne stvari, njihove lastnosti, medsebojne povezave in pravila, ki veljajo za to področje. Takšen opis je torej ontologija in vsebuje pojme: razred, primerek, razmerje, lastnost, vrednost in pravilo, kot je razvidno iz tabele 1.

2.2 Predstavitev ontologije

Ko ontologijo opišemo, jo moramo tudi predstaviti, kar pomeni, da jo zapišemo v obliki, ki jo lahko uporabljajo tudi drugi uporabniki.⁸ Opis je sestavljen iz besed in fraz v naravnem jeziku (slovenščina, angleščina, japonščina itd.) oz. uporabljamo slovar pojmov in stavke, ki predstavljajo povezavo med pojmi.

Pojem	Primer
Razredi (splošne stvari)	Letalo, letalska družba, let, razred potovanja, letališče, pilot itd.
Primerki (določene stvari)	Boeing 777-200LR, Airbus A340, TP121, AF1019, Air France, Lufthansa, Iberia, ekonomski, poslovni, Caracas Simon Bolivar, Bogotá El Dorado, Jean-Luc Piccard itd.
Razmerja : podrazred, del, primerek, ima-končno-destinacijo, izvaja, uporablja, upravlja itd.	Airbus A340 je primerek letala. Let TP121 ima končno destinacijo na letališču Caracas Simon Bolivar.
Lastnosti	Število sedežev, čas potovanja, razdalja, cena, opremljenost, izkušenosť itd.
Vrednosti	405, 10 ur 30 min, 8209 km, 164.800 SIT itd.
Pravila	IF letalo (X, 'Airbus A340') & st-sedežev (X, 405) THEN st-ekonomskih-sedežev (X, 325) [Pomen: Letalo Airbus A340 ima skupno 405 sedežev, od tega jih 325 spada v ekonomski razred.]

Tabela 1: Primer enostavne ontologije na področju letalskih potovanj

⁶ Model-Driven Architecture je metodologija za razvoj programske opreme, ki so jo predlagali pri Object Management Group (OMG). Funkcionalnosti sistema določimo v modelu, ki je neodvisen od platforme (PIM), s pomočjo ustreznega jezika za modeliranje (npr. UML). Za dejansko implementacijo pa je treba model PIM prevesti v model, ki je specifičen za izbrano platformo (PSM).

⁷ Način razmišljanja o delu sveta oz. ustvarjanje pojmov za abstraktne izraze.

⁸ To je pomembno predvsem zaradi dejstva, da je ena od ključnih lastnosti ontologij souporaba znanja med različnimi uporabniki področja, ki ga opisuje ontologija.

Predstavitev ontologije tako pomeni predstavitev znanja v obliki pojmov in stavkov, kjer najprej določimo pojme (oz. jih že imamo izoblikovane v našem mentalnem modelu slovarja), kasneje pa te pojme medsebojno povežemo in tako izdelamo še poglobljeno znanje o določeni domeni.

V informatiki se uporablja nekoliko bolj zapleten način predstavitve, saj moramo izdelati model, ki ga lahko programska oprema neposredno uporabi. Za predstavitev znanja tako uporabljamo razrede, primerke, razmerja, lastnosti, vrednosti in pravila, s katerimi izražamo pomen na določenem področju, pojme naravnega jezika pa uporabljamo za oznake prej omenjenih konceptov. Za predstavitev ontologije po navadi namesto naravnega jezika uporabimo logični jezik za predstavitev znanja, saj je logični jezik bolj jasno in točno izrazno sredstvo, medtem ko je naravni jezik pogosto dvoumen. Formalne jezike za predstavitev ontologij si bomo podrobneje ogledali v razdelku 8.

3 Vloga ontologij

Ontologije so ključni element pri semantičnem spletu, kjer se prepleta človeško razumevanje simbolov z zmožnostjo računalniškega procesiranja. V zadnjem času se ontologije vedno bolj uporabljajo na področjih integracije inteligentnih sistemov, zbiranja informacij, elektronske trgovine in obvladovanja znanja. Razlog za vedno večjo priljubljenost leži v njihovi obljudi: »skupno razumevanje domene, ki ga lahko souporabljajo ljudje in računalniški sistemi« [8]. Uporaba ontologij na področju kognitivnih agentov⁹ prav tako vzbuja vedno več zanimanja, saj se potencial uporabe ontologij odpira predvsem pri medsebojni komunikaciji in mehanizmih sklepanja pri delovanju agentov v večagentnih sistemih.¹⁰

Uporaba ontologij in podpornih orodij prinaša možnosti za izboljšanje obvladovanja znanja predvsem v večjih organizacijah. Ključni koraki pri uporabi ontologij so:

- **pridobivanje oz. učenje ontologij in povezovanje le-teh z velikimi količinami podatkov** – zaradi prilagodljivosti mora biti ta proces avtomatiziran s pomočjo metod za luščenje informacij in procesiranje naravnega jezika;

- **shranjevanje in vzdrževanje ontologije in njenih primerkov** – sprejeti je treba odločitev, v kakšni obliki bo shranjena ontologija, tako da nam je na voljo repozitorij z lastnostmi podatkovne baze in preprosti obrazci za sklepanje na podlagi elementov ontologije;
- **iskanje po semantično obogatenih informacijskih virih** – sama predstavitev znanja in informacij ni dovolj, uporabniki ontologije morajo uporabljati in po njih povpraševati. Pojavljajo se novi načini povpraševanja, ki nam omogočajo pridobivanje znanja iz porazdeljenih virov, vse raziskave pa konvergirajo k semantično obogatenim iskalnikom. Pri vseh omenjenih korakih so zaradi lažjega obvladovanja ontologij potrebna orodja, ki so podrobneje predstavljena v razdelku 9.

3.1 Semantični splet

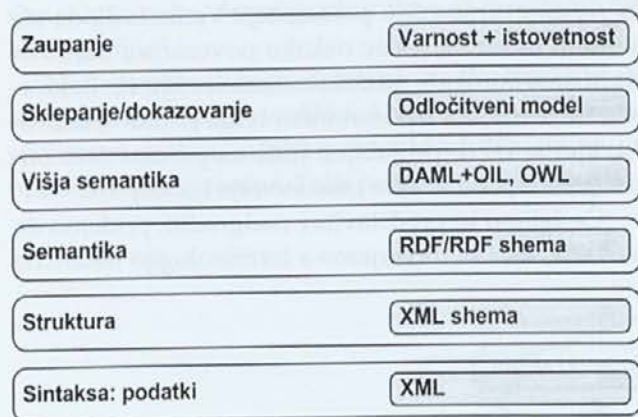
Svetovni splet doživlja zelo velik uspeh, kar nakazuje število uporabnikov in dokumentov,¹¹ ki se na njem nahajajo. Vendar ostaja dostop do informacij še vedno težaven, saj mora uporabnik vsebino najprej poiskati, nato pa še tolmačiti. Pri tem se pojavljajo vprašanja, kaj je vsebina dokumenta in kakšna je semantika samih povezav. Pri iskanju informacij na svetovnem spletu moramo tako poznati bodisi natančen naslov iskanega dokumenta, bodisi ga poiščemo s pomočjo spletnega iskalnika. Problem spletnih iskalnikov je predvsem v predstavitvi iskalne zahteve vmesniku iskanja, kasneje pa tudi predstavitev rezultatov iskanja. Tim Berners Lee je zato v začetku tega tisočletja predstavil idejo semantičnega spleta kot nadgradnjo obstoječega spleta, kjer informacije dobijo pomen, kar omogoča boljše medsebojno sodelovanje ljudi in računalnikov [2]. Moč semantičnega spleta je v združevanju tehnologij za obvladovanje znanja na področju umetne inteligence in spletnih tehnologij. Največ raziskovanja poteka na področju spletnih ontologij, saj predstavljajo način obvladovanja raznovrstnih predstavitev spletnih virov. Domenski model iz ontologije se lahko uporabi kot enotna struktura za skupno predstavitev in semantiko informacij. Prav tako se v viziji semantičnega spleta jasen pomen dialoga med oddaljenimi aplikacijami ali agenti doseže s pomočjo uporabe in s sklicevanjem na ontologije.

⁹ Za agenta kot entiteto velja, da je sposoben delovati v svojem okolju, deluje avtonomno – ima kontrolo nad svojim delovanjem, lahko komunicira z drugimi agenti in je zmožen zaznavanja svojega okolja. Kognitivni agent pa kot specializacija agenta ob vseh že omenjenih lastnostih deluje proaktivno in uporablja svojo lastno predstavo okolja [12].

¹⁰ Večagentni sistem (MAS) je sestavljen iz množice agentov, ki medsebojno sodelujejo v imenu uporabnikov z različnimi cilji.

¹¹ Trenutne statistike kažejo, da je bilo konec leta 2002 kar 320 milijonov uporabnikov interneta, svetovni splet pa vsebuje več kot 800 milijonov spletnih strani.

Semantični splet je zgrajen v obliki sklada, kjer zgornje ravni nadgrajujejo spodaj ležeče, kot to prikazuje slika 3.



Slika 3: Sklad arhitektur semantičnega spleta

Vsi jeziki semantičnega spleta na vseh ravneh uporabljajo XML sintakso, vsaj za potrebe izmenjave podatkov. Na sliki 3 lahko vidimo, da se XML nahaja na dnu sklada in ima vlogo osnovne sintakse za medsebojno izmenjavo podatkov na svetovnem spletu. Nad ravnijo XML-a se nahaja XML shema, ki omogoča strukturiranje spletnih objektov, podobno tistemu v podatkovnih bazah in je tako primerljiva s shemami podatkovnih baz. Naslednja raven je RDF/S, s katero pridobimo enostaven jezik za izražanje ontologij (pojmov, relacij in primerkov). To raven razširja DAML+OIL¹² ali OWL,¹³ s katerima lahko veliko bolj podrobno zapišemo ontologije in uporabljamo raven RDF/S za predstavitev primerkov konstruktov ontologije. DAML+OIL in OWL oba neposredno uporabljata podatkovne tipe sheme XML.

Na najvišji ravni, t. i. »spletu zaupanja«, se nahajajo metode za sklepanje in dokazovanje, ki uporabljajo samodejno dokazovanje, kot tudi varnostne ele-

mente in elemente, povezane z istovetnostjo, ki so še zelo slabo raziskani in nezreli za uporabo. Na samem vrhu sklada arhitektur se nahajajo programi, ki uporabljajo vse spodaj ležeče tehnologije in jih lahko imenujemo »inteligentne aplikacije«, saj opravila opravljajo na naprednejši način in prinašajo bolj »inteligentne« storitve.

3.2 Praktični primeri ontologij

Aplikacije, kjer se uporabljajo ontologije, so iz dneva v dan številčnejše in jih lahko najdemo na različnih področjih semantično obogatenih spletnih aplikacij, porazdeljenega obvladovanja znanja, luščenja informacij itd. V svetovnem spletu se ontologije uporabljajo za povezovanje vsebine v celoto, kjer se pojavljajo semantični portali,¹⁴ kot je npr. OntoWeb,¹⁵ ki je eden najbolj razširjenih. Namen projekta je omogočiti integracijo informacij na svetovnem spletu s podporo semantični integraciji porazdeljenih virov, vpeljavo semantičnih poizvedb in navigacijskih pogledov ter ne nazadnje tudi programskim agentom omogočiti dostop do vsebine portala.

Zelo veliko raziskovanja na področju ontologij poteka v medicini, kjer je najbolj razširjena ontologija GALEN.¹⁶ Ne samo da jo odlikuje velika izrazna moč pri zajemu bogate semantike na področja delovanja zdravnikov, ampak se ukvarja tudi z začasnimi in prostorsko razpršenimi vidiki znanja ter sklepanjem na podlagi znanja. Ena od prednosti so tudi zelo razširljiva razvojna orodja, saj uporabnikom omogočajo prilagajanje osnovne terminologije specifičnim zahtevam.

Obstajajo še številne druge ontologije, kot je npr. Cyc,¹⁷ ki velja za eno največjih in najbolj popolnih splošnih baz znanja z integriranim mehanizmom sklepanja, OBO¹⁸ na področju biologije, SUMO,¹⁹ kot prav tako ena največjih formalnih ontologij v splošni rabi, DOLCE²⁰ kot del WonderWeb²¹ in mnoge druge.

¹² DARPA (Defense Advanced Research Program) Agent Markup Language – Ontology Inference Layer. Jezika, ki podpirata semantični splet in temeljita na svetovnem spletu in XML. DAML se je razvil iz programa, sponzoriranega s strani ameriške DARPA-e, OIL pa iz projekta, ki ga financira Evropska unija. Skupaj tako tvorita enega semantično najbolj izraznih jezikov za opis dokumentov na svetovnem spletu in sta podprta tudi s strani W3C konzorcija.

¹³ Web Ontology Language oz. Ontology Web Language je razvila skupina za spletne ontologije na konzorciju W3C in igra vlogo naslednika DAML+OIL. OWL je v tem trenutku najbolj izrazen jezik za predstavitev znanja na področju semantičnega spleta.

¹⁴ SEAL – SEmantic portALS

¹⁵ <http://OntoWeb.org>

¹⁶ <http://www.opengalen.org>

¹⁷ <http://www.openccyc.org>

¹⁸ <http://obo.sourceforge.net>

¹⁹ Suggested Upper Merged Ontology, <http://www.ontologyportal.org>

²⁰ <http://www.loa-cnr.it/DOLCE.html>

²¹ <http://wonderweb.semanticweb.org/>

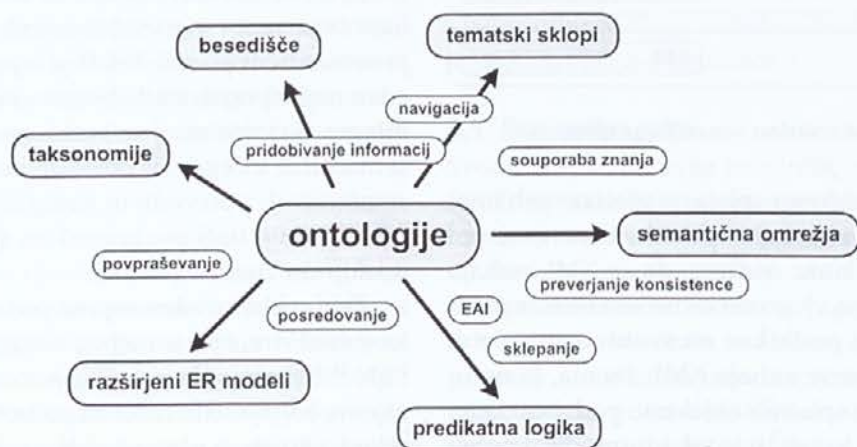
4 Različni pogledi na ontologije

Ontologije so povezane s številnimi področji in tehnologijami. Na sliki 4 vidimo, da se pri pojmovanju ontologije razširjeni modeli ER²² in predikatna logika, s podporo odločanju, mehanizmi sklepanja in EAI,²³ nahajajo bolj v zalednem delu (spodnji del slike 4), saj je na tej ravni semantika formalno določena. Drugi elementi pridejo v ospredje pri opravih, kot so povpraševanje, pridobivanje informacij, navigacija in ne nazadnje predvsem souporaba znanja.

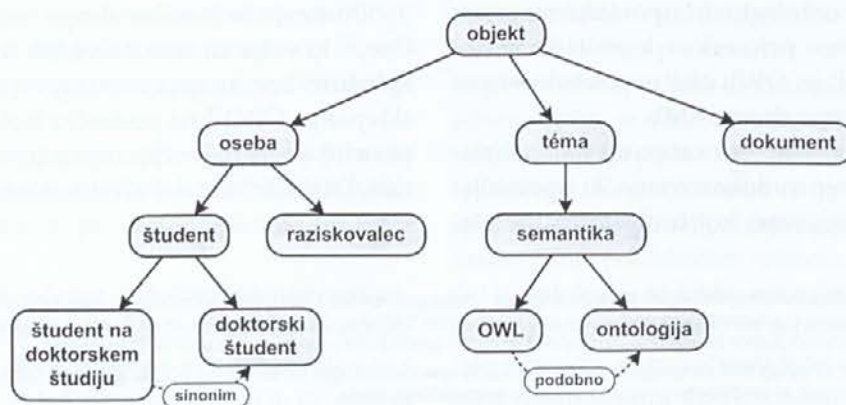
Najprej si podrobneje pogledjmo taksonomijo, ki predstavlja segmentacijo oz. klasifikacijo in ureditev

elementov v razvrstitveni sistem, glede na medsebojne povezave. Slika 5 prikazuje preprost primer iz akademske sfere. Modeliranje začnimo s splošnim objektom in pojmi (oseba, tema, dokument), ki se nam za razumevanje zdijo pomembni. Vemo tudi, da sta študent in raziskovalec nekako povezana z osebo in da imamo študente na doktorskem študiju oz. doktorske študente. Za nas zanimiva tema je tudi semantika, kjer je OWL priljubljen jezik za predstavitev ontologij. To je na kratko taksonomija konceptov.

Če želimo to predstavitev nadgraditi, pridemo do slovarja, kjer se ukvarjamo s terminologijo natančno



Slika 4: Različni pogledi na ontologije



Slika 5: Primer taksonomije in slovarja

²² Entity Relation – entiteta razmerij, ki se uporablja za prikaz relacij med dvema ali več entitetami.

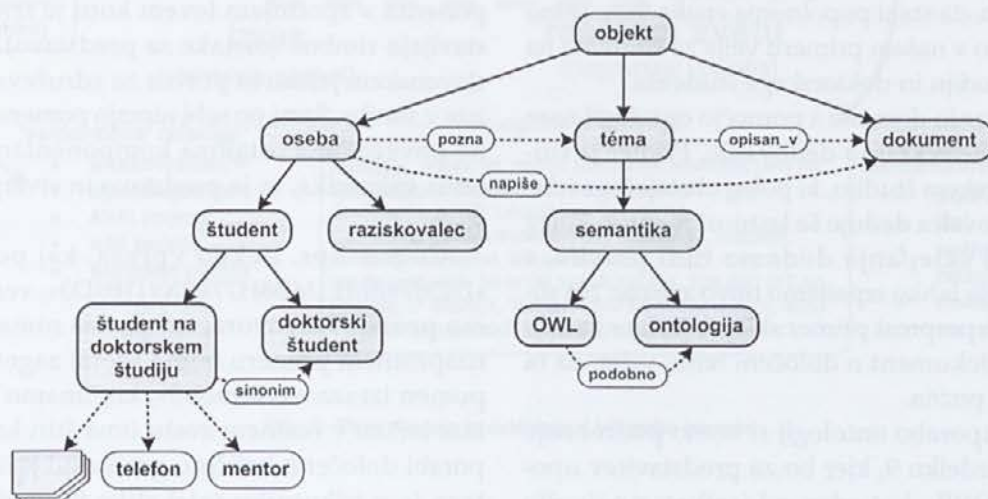
²³ Enterprise Application Integration – integracija aplikacij: metode, tehnike in orodja za povezovanje in uskladitev uporabniških programov in podatkov zaradi večje združljivosti.

določenega področja. V našem primeru se pojavita dve fiksni povezavi, s katerima povemo, da je študent na doktorskem študiju sinonim za doktorskega študenta in da je koncept OWL podoben ontologiji. Omeniti je treba, da pri gradnji slovarja temeljimo na bibliografiji.²⁴

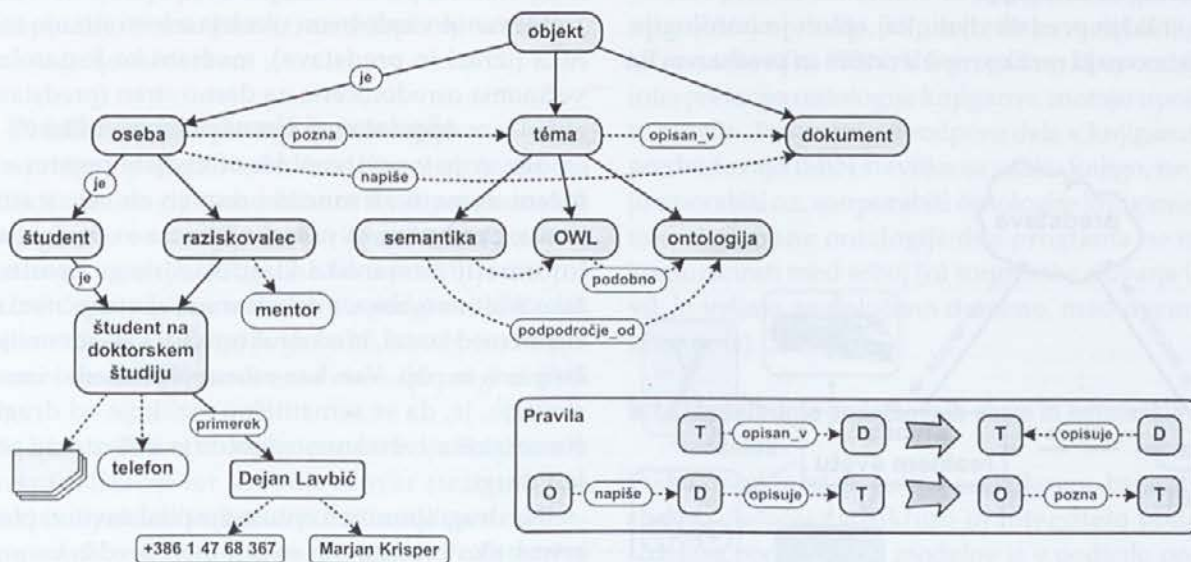
Naslednji korak bi predstavljal uvedbo tematskega sklopa, prikazanega na sliki 6. Uvedemo medsebojne povezave, kot je »oseba pozna téma« in »téma je opisana v dokumentu«. Taka predstavitev se uporab-

lja predvsem za navigacijo in vizualizacijo, saj lahko npr. pri pregledovanju dokumentov preprosto poiščemo še druge dokumente, ki jih je napisala neka oseba. Na tej ravni uvedemo tudi attribute, s katerimi lahko opišemo dodatne podatke o študentu na doktorskem študiju, kot je npr. telefon in mentor.

Tako pridemo do preprostega primera ontologije, prikazanega na sliki 7. Nadgradnjo predstavlja predvsem hierarhija tipa »je«, s čimer opišemo dedovanje. V omenjenem primeru npr. velja, da študent na



Slika 6: Primer tematskega sklopa



Slika 7: Primer ontologije

²⁴ Po določeni tematiki urejen seznam del.

doktorskem študiju podeduje vse lastnosti, ki jih ima raziskovalec.

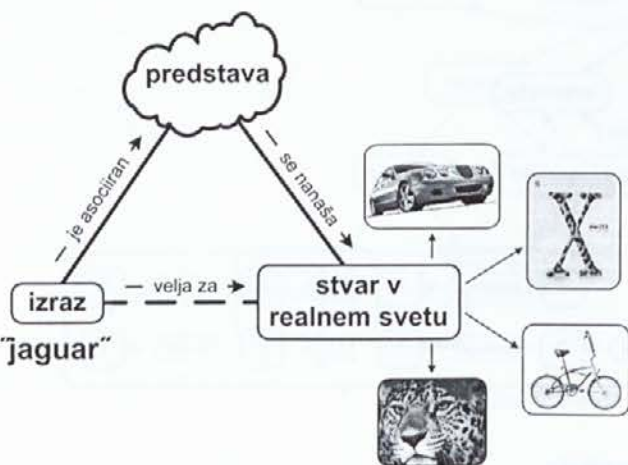
Prav tako to velja tudi za medsebojna razmerja, saj če oseba pozna določeno temo, potem jo pozna tudi študent. Ravno zaradi tega se koncepta OWL in ontologija pomakneta raven višje, saj ni nujno, da vse, kar velja za semantiko, velja tudi za ontologijo. Da pa ne izgubimo semantičnega razmerja, dodamo relacijo, ki pravi, da je semantika podpodročje ontologije. Če je potrebno, lahko v tem primeru dodamo tudi dodatne tranzitivne povezave. V ontologiji lahko za dva koncepta zapišemo, da sta si popolnoma enaka in ne samo podobna, kot to v našem primeru velja za študenta na doktorskem študiju in doktorskega študenta.

Pri modeliranju domene s pomočjo ontologij nam je na voljo tudi večkratno dedovanje. Primer je študent na doktorskem študiju, ki poleg obstoječega telefona od raziskovalca deduje še lastnost mentor. Poleg tega na ravni sklepanja dodamo tudi pravila, s pomočjo katerih lahko izpeljemo novo znanje. Na sliki 7 je prikazan preprost primer sklepanja, kjer za osebo, ki napiše dokument o določeni temi, velja, da ta oseba to temo pozna.

Dejansko uporabo ontologij si bomo podrobneje pogledali v razdelku 9, kjer bo za predstavitev uporabljen jezik OWL, kot eden od jezikov z največjo izrazno močjo.

5 Slovar in ontologija

Da bi si lažje predstavljali, kaj sploh je ontologija, poskusimo najti razliko med izrazom in predstavo. En



Slika 8: Homonim "jaguar" v trikotniku pomena

pogled na to je razlikovanje slovarja in ontologije. Kot primer si pogledjmo izraz »jaguar«, ki lahko v realnem svetu predstavlja številne stvari [5] (glej sliko 8):

- leopardu podobna črna lisasta zver, ki živi v Ameriki,
- avtomobil angleške tovarne Jaguar,
- kodno ime operacijskega sistema Mac OS X 10.2,
- športno kolo angleške tovarne Jaguar.

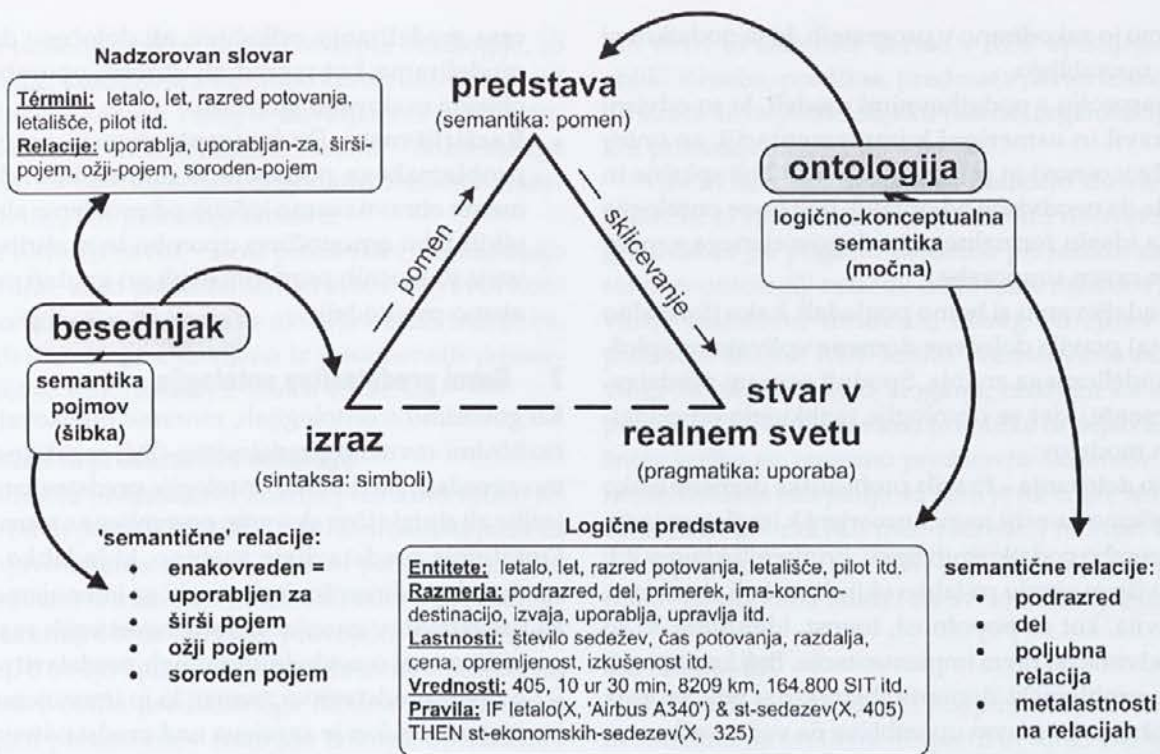
Slika 9 poskuša v strnjeni obliki prikazati tri komponente pomena [2] (oglišča trikotnika), ki se pojavljajo v naravnem jeziku (npr. v slovenščini). Prva komponenta v spodnjem levem kotu je izraz, ki ga sestavljajo simboli (oznake za predstavo) ali besede v slovenskem jeziku in pravila za združevanje teh izrazov v stavke. Sami po sebi nimajo pomenov, dokler jih ne povežemo z ostalima komponentama oz. ogliščema trikotnika, to je predstavo in stvarjo v realnem svetu.

Če nas npr. nekdo vpraša, kaj pomeni izraz »D23978BMF1MB8H77E8X9TJB9D3«, verjetno ne bomo poznali odgovora, saj za nas nima pomena. V nasprotnem primeru bomo skoraj zagotovo poznali pomen izraza »avtomobil«, ker imamo predstavo o tem izrazu v realnem svetu (ima štiri kolesa, motor, porabi določeno količino goriva itd.). Ravno zaradi tega je v trikotniku (glej sliko 9) med izrazom in stvarjo v realnem svetu povezava črtkana, saj ni neposredne povezave, ker človek pri povezovanju pojmov in stvari v realnem svetu potrebuje predstavo.

Slovar se v splošnem ukvarja z levo stranjo trikotnika (izrazi in predstave), medtem ko je ontologija večinoma osredotočena na desno stran (predstave in stvari v realnem svetu), kot to prikazuje slika 9.

Slovar je v prvi vrsti klasifikacijski prostor v določeni domeni ali množici domen ali celo v širšem svetu, predvsem za potrebe iskanja in pridobivanja informacij. Semantika klasifikacijskega prostora je tako relativno šibka, z enostavnimi semantičnimi relacijami med izrazi, ki so strukturirani s taksonomijo relacij širši in ožji. Vse, kar moramo poznati o izrazu v slovarju, je, da se semantično razlikuje od drugih (s čimer izločimo dvoumnost) in da je širši oz. ožji pojem kot drugi.

Na drugi strani želi ontologija predstaviti zapleteno semantiko predstav in relacij med predstavami, njihovimi lastnostmi, vrednostmi, omejitvami in pravili. Namen ontologije se razlikuje od namena slovarja, saj skuša ontologija zajeti in predstaviti pomen določenih domen, množice domen ali celotnega sveta, ker skuša



Slika 9: Vloga slovarja in ontologije v trikotniku pomena

eksplicitno simulirati razumevanje, ki ga ima človek v svojih mentalnih modelih problemske domene ali sveta. V nasprotju s slovarjem želimo pri ontologijah konceptualno semantiko izraziti čim bolj natančno, obsežno in konsistentno.

6 Podatkovno modeliranje in ontologije

V nasprotju s podatkovnim modeliranjem je glavna prednost ontologij v neodvisnosti od programskih rešitev, saj je ontologija sestavljena iz sorazmerno splošnega znanja, ki ga lahko uporabljamo pri različnih opravilih in v različnih programih. Korak bliže konceptu ontologij je globalni konceptualni podatkovni model, ki je tudi neodvisen od programskih rešitev.

Računalniška ontologija predstavlja dogovor o konceptualizaciji ter vsebuje slovar (terminov in oznak) in opredelitive konceptov ter njihove medsebojne odvisnosti v določeni domeni. Podatkovni model v nasprotju z ontologijo predstavlja strukturo in integriteto podatkovnih elementov praviloma v določenem programu, ki uporablja ta podatkovni model. Konceptualizacija in uporaba slovarja podat-

kovnega modela tako ni a priori namenjena za souporabo v več različnih programih [1].

Kot primer omenimo ontologijo knjigarne, kjer npr. obstaja pravilo, ki pravi, da je vsaka knjiga določena s številko ISBN. Vsi programi, ki uporabljajo to interpretacijo ontologije knjigarne, morajo upoštevati to pravilo. Programi za podporo dela v knjigarni, ki ne predvidevajo ISBN številke za vsako knjigo, ne morejo uporabiti oz. souporabiti ontologije knjigarne. Brez tako definirane ontologije dva programa ne moreta komunicirati med seboj (ni souporabe slovarja in pravil, ki veljajo za določeno domeno, med dvema programoma).

6.1 Modeliranje podatkovnih shem in ontoloških modelov

Podatkovni modeli, kot so podatkovne baze ali XML sheme, določajo strukturo in integriteto podatkov. Izdelava podatkovnih modelov je v podjetju po navadi odvisna od določenih zahtev in opravil, ki morajo izvedena biti v podjetju. Semantiko podatkovnih modelov v večini primerov določajo neformalni dogovori med razvijalci in uporabniki podatkovnega modela.

Najdemo jo zakodirano v programih, ki ta podatkovni model uporabljajo.

V nasprotju s podatkovnimi modeli, ki so odvisni od opravi in usmerjeni k implementaciji, so ontologije že v osnovi in po opredelitvi čim bolj splošne in kar se le da neodvisne od opravi. Bolj ko se ontologija približa idealu formalnega in dogovorjenega zapisa, večja je raven souporabe.

V nadaljevanju si bomo pogledali, kako (formalno izražena) pravila določene domene vplivajo na splošnost modeliranega znanja. Spodnji seznam predstavlja elemente, kjer se ontologije razlikujejo od podatkovnih modelov:

- **Nivo delovanja** – Pravila problemske domene lahko zapišemo na nižji ravni, usmerjeni k implementaciji, z uporabo podatkovnih tipov, primarnih ključev itd. Na višjem nivoju pa lahko vključimo bolj abstraktna pravila, kot so popolnost, togost, identiteta, ki so neodvisna od ravni implementacije. Bolj ko so pravila o problemski domeni abstraktna, bolj so le-ta splošna in ponovno uporabljiva na več mestih.
- **Moč izražanja** – Namen jezikov za manipulacijo s podatki, kot je SQL, je ohranitev integritete podatkov in uporaba splošnih konstruktov, kot so npr. ključi. V splošnem morajo pravila o domeni izražati ne samo integriteto podatkov, ampak tudi konceptualizacijo domene. Tako mora jezik za izražanje pravil o problemskem področju vsebovati konstrukte, s katerimi lahko izrazimo druge pomenske omejitve, kot je taksonomija ali podpora sklepanju, kar najdemo npr. pri DAML+OIL in OWL.
- **Povezanost z uporabnikom, namenom in ciljem** – Podatkovni model se zelo dobro prilega določenim ciljem in potrebam uporabnikov, ki program uporabljajo. Jasno je, da na ta račun trpi splošnost konceptualizacije, ki je odvisna od granularnosti pro-

cesa modeliranja, odločitev, ali določeno dejstvo modeliramo kot razred ali atribut, uporabo tipa objekta iz slovarja ali ne itd.

- **Razširljivost** – Pri konceptualizaciji ontologije problemskega področja je pomembno, da elemente obravnavamo ločeno od problema ali opravi in tako omogočimo uporabo in razširitev slovarja v splošnih primerih, ki jih pri gradnji mogoče nismo predvideli.

7 Ravnih predstavitev ontologije

Ko govorimo o ontologijah, moramo razlikovati med različnimi ravnimi predstavitev. Teh razlik se moramo zavedati, saj si lahko ontologije predstavljamo kot jezike ali sintaktične slovarje, opremljene s semantiko. Ontologije predstavljajo vsebino, ki jo lahko predstavimo le z ustreznim jezikom, ki ga imenujemo jezik za predstavitev znanja. Zaradi omenjenih razlogov govorimo vsaj o naslednjih ravnih predstavitev:

- o ravni predstavitev znanja, ki jo imenujemo tudi metaraven, ker je zgrajena nad predstavitvijo konceptov in jo tudi razširja;
- o ravni predstavitev ontologije oz. objektni ravni predstavitev znanja, kjer se nahaja vsebina in kjer so izražene ontologije;
- o ravni primerkov ontologije, kjer najdemo primerke razredov ontologij.

Raven predstavitev znanja (najvišja metaraven) opredeljuje konstrukte, ki se bodo uporabljali na ravni koncepta ontologije. Ti konstrukti so razred, razmerje, primerek itd., kot je prikazano v tabeli 2. Jezike za predstavitev znanja [2] delimo na tiste, ki so nastali pred semantičnim spletom (KL-ONE, Ontolingua, Classic, LOOM, KIF,²⁵ CycL, UML²⁹), in jezike, ki jih povezuje s semantičnim spletom (RDF/S, DAML+OIL, OWL).

Raven	Primer konstruktov
Raven predstavitev znanja (KR ²⁵)	Razred, razmerje, primerke, funkcija, atribut, lastnost, omejitve, aksiom, pravilo
Raven koncepta ontologije (OC ²⁶)	Oseba, avto, lokacija, dogodek, finančna transakcija, načrtovanje, potovanja, nakup, hiša itd.
Raven primerkov (OI ²⁷)	Dejan Lavbič, oseba312, Lanos 1.5 SE itd.

Tabela 2: Ravnih predstavitev ontologij

²⁵ Knowledge representation

²⁶ Ontology concept

²⁷ Ontology instance

²⁸ Knowledge Interchange Format je jezik za predstavitev znanja, ki je bil razvit pred pojavom semantičnega spleta. Temelji na predikatni logiki prvega reda, v današnjem času pa obstaja standard ISO-KIF, ki podpira številne oblike, vključno z XML in konceptualnimi grafi.

²⁹ Unified Modeling Language je jezik za specifikiranje, vizualiziranje, modeliranje in dokumentacijo izdelkov pri razvoju objektno usmerjenih sistemov. UML ni omejen samo na modeliranje programske opreme, ampak se pogosto uporablja tudi na drugih področjih (npr. modeliranje poslovnih procesov, organizacijske strukture itd.).

Na naslednji ravni, ravni koncepta ontologije, so ontologije, definirane z uporabo konstruktov iz ravni predstavitve znanja. Tukaj se ukvarjamo z modeliranjem generične oz. univerzalne vsebine – domenskega znanja o osebah, lokacijah, dogodkih, finančnih transakcijah itd., kot prikazuje tabela 2.

Na najnižji ravni, ravni primerkov, se nahajajo konstrukti, ki so primerki konstruktov na ravni koncepta ontologije. Ta raven se ukvarja z bazo znanja oz. bazo dejstev in je sestavljena iz posameznih primerkov, kot so npr. oseba312, Lanos 1.5 SE itd.

8 Jeziki za predstavitev ontologij

Na področju predstavitve znanja poznamo veliko jezikov. Eden prvih je bil KIF. To je računalniški jezik za izmenjavo znanja med različnimi programi in ima deklarativno semantiko (pomen izrazov v predstavitvi je razumljiv brez uporabe prevajalca za omenjene izraze). S svojo splošnostjo (predstavitev poljubnih stavkov v obliki predikatnega računa prvega reda) omogoča predstavitev znanja o znanju, opredelitev objektov, funkcij in relacij. Omembe vreden je tudi OKBC,³⁰ saj je prinesel enoten model predstavitve znanja in temelji na splošni konceptualizaciji razredov, primerkov, dedovanja itd.

V nadaljevanju se bomo osredotočili predvsem na jezike za predstavitev ontologij, ki so povezani s semantičnim spletom, saj se v praksi tudi najpogosteje uporabljajo.

8.1 RDF(S)

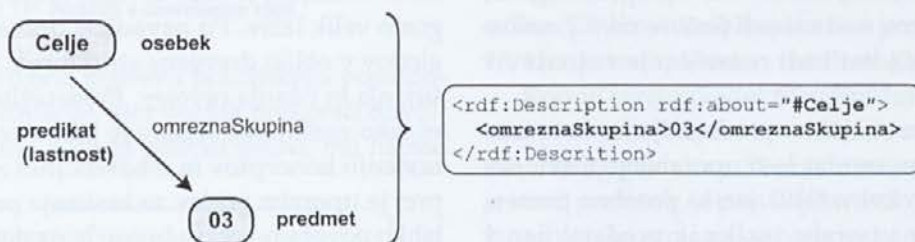
RDF³¹ je bil razvit za potrebe opisovanja metapodatkov virov na svetovnem spletu v obliki stavkov, dru-

gih virov in lastnosti. Stavki v RDF opisujejo vire v obliki <oseba, predikat, predmet>, ki so lahko spletne strani ali dejanski objekti (osebe, organizacije itd.), kot prikazuje primer na sliki 10.

Vire in lastnosti opišemo s pomočjo sheme RDF³² (RDF/S), ki RDF razširja z dodatnimi načrtovalskimi gradniki in jih pogosto najdemo pri jezikih za predstavitev ontologij: razredi, dedovanje razredov, dedovanje lastnosti, domena, obseg omejitev itd. S pomočjo sheme RDF lahko predstavimo razrede, vloge in omejitve nad vlogami, medtem ko lahko s pomočjo RDF predstavimo primerke in dejstva. V takšnem jeziku ne moremo predstaviti aksiomov in edini mehanizem sklepanja, ki nam je na voljo, temelji na obstoječih povezavah med razredi. Prednost RDF-ja je v uporabi XML-a in URI-ja,³³ saj je njegov namen ponovna uporaba entitet na svetovnem spletu.

8.2 DAML+OIL

DAML+OIL³⁴ sta jezika, ki podpirata semantični splet in temeljita na svetovnem spletu in XML. DAML se je razvil iz programa, ki ga sponzorira ameriška DARPA, OIL pa iz projekta, ki ga financira Evropska unija. Skupaj tako tvorita enega semantično najbolj izraznih jezikov za opis dokumentov na svetovnem spletu in ju podpira konzorcij W3C.³⁵ V zadnjem času se je pojavila razširitev DAML-Service³⁶ (DAML-S), ki predstavlja množico ontologij v DAML+OIL in se ukvarja s semantiko spletnih storitev, storitev modeliranih v obliki procesov, profilov storitev, storitvenih modelov in osnov samih storitev (npr. konkretnih realizacij abstraktno definiranih komponent storitev, primerljivi z delom jezika WSDL³⁷ za povezovanje).



Slika 10: Trditev RDF v obliki grafa in formalnega zapisa

³⁰ Open Knowledge Base Connectivity

³¹ <http://www.w3.org/RDF>

³² <http://www.w3.org/TR/rdf-schema>

³³ Uniform Resource Identifier – Splošen identifikator, ki se uporablja za razlikovanje konkretnih in abstraktnih virov na svetovnem spletu [14].

³⁴ <http://www.daml.org/2001/03/daml+oil-index.html>

³⁵ World Wide Web Consortium

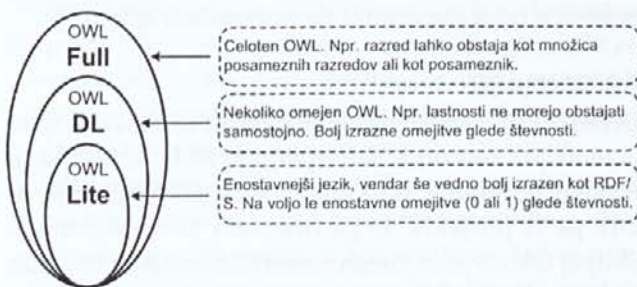
³⁶ <http://www.daml.org/services/owl-s>

³⁷ Web Service Description Language

8.3 OWL

Web Ontology Language³⁸ oz. Ontology Web Language je v tem trenutku najbolj izrazen jezik za predstavitev znanja na področju semantičnega spleta. V nasprotju z DAML+OIL je OWL nastal pod sponzorstvom W3C, ki je prvo verzijo tega jezika predstavila leta 2003.

OWL ima tri ravni: OWL Lite, OWL DL (za opisno logiko) in OWL Full (glej sliko 11). Omenjene ravni predstavljajo naraščajočo stopnjo izraznosti, kjer višje ravni jezika vsebujejo nižje oz. jih razširjajo. Trditev, veljavna v OWL Lite, je še vedno veljavna v OWL DL in OWL Full. Prav tako je trditev, veljavna v OWL DL, veljavna tudi v OWL Full, vendar ne nujno v OWL Lite.



Slika 11: Ravni jezika OWL

OWL je zgrajen na konceptih DAML+OIL, saj ima, podobno kot predhodnik, razrede (in podrazrede), lastnosti (in podlastnosti), omejitve nad lastnostmi in primerke razredov ter lastnosti. OWL, podobno kot DAML+OIL, za opisovanje razredov in informacij o podatkovnih tipih (iz XML sheme) uporablja konstrukte *subClassOf*, *disjointWith* in omogoča logične kombinacije izrazov nad razredi (*intersectionOf*, *unionOf*, *complementOf*), kot tudi razvrščanje razredov v sezname.

OWL si preprosto lahko predstavljamo kot množico RDF trojčkov, vendar le-ti uporabljajo OWL slovar, ki jim daje v okviru OWL jezika poseben pomen. Praktični primer uporabe jezika je predstavljen v razdelku 10.

9 Orodja za obvladovanje ontologij

Pri razvoju, integraciji, opisovanju, shranjevanju ontologij in povpraševanju po njih lahko uporabimo številna orodja, ki so na voljo kot komercialni produkti ali plod dela akademskih skupnosti in so prosto dostopna. Če izpostavimo širino in raznolikost lastnosti, je zelo primerno orodje Protégé,³⁹ ki so ga razvili na Stanford Medical Informatics. Razvojno okolje je odprtokodna rešitev urejevalnika ontologij, katerega funkcionalnosti je mogoče razširiti s številnimi vtičniki. S stališča natančne podpore jezikom za zapis ontologije sta dobra izbira orodji Ontolingua⁴⁰ in OpenCyc,⁴¹ ki prinašata visoko izrazno moč in popolno podporo specifikacijam. Med urejevalniki, ki neposredno podpirajo DAML+OIL, je zaradi močne podpore pri uporabi opisne logike uporabno orodje OilEd.⁴² Paket On-To-Knowledge tool suite⁴³ je prav tako zanimiva rešitev, sestavljena iz številnih orodij: OntoEdit, OntoShare, OntoExtract, OntoWrapper itd. Vsako od orodij pokriva določeno funkcionalnost: od urejanja ontologij do uporabe ontologij pri souporabi znanja, polavtomatskega učenja ontologij iz naravnega jezika, luščenja strukturiranih informacij iz spletnih virov itd. Nekateri avtorji pa predlagajo uporabo UML-ja kot predstavitvenega jezika ontologij. Prednost uporabe tega jezika je v dostopnosti orodij za modeliranje, slabost pa nekoliko slabša podpora specifikacijam in s tem tudi omejena izrazna moč.

Ključ pri uporabnosti orodja za organizacijo in obvladovanje ontologij leži tudi v intuitivni predstavitvi ontologije ter preprostem obvladovanju povezovanja konceptov in relacij. Ker številni modeli omogočajo večkratno dedovanje pri hierarhičnih konceptih in relacijah, je tudi obvladovanje teh asociacij pogosto velik izziv. Po navadi se uporablja pristop pogledov v obliki drevesne strukture z zmožnostjo razširjanja in ožanja nivojev. Predstavitev v obliki grafa ni tako razširjena, čeprav je lahko zelo uporabna pri urejanju konceptov in medsebojnih relacij. Korak naprej je uporaba grafov za brskanje po ontologiji, kjer lahko povečamo podrobnosti le na določenem delu, ki nas zanima. V orodju Protégé obstaja celo vtičnik Jambalaya, s katerim lahko izvajamo grafično približevanje, kjer podrejeni koncepti ostanejo v sklopu nadrejenega, tako da se lahko prosto sprehajamo po mreži konceptov, povezanih z medsebojnimi relacijami.

Ena od pomembnih lastnosti urejevalnika je tudi podpora delu z lupino odločitvenega sistema, ki nam

³⁸ <http://www.w3.org/TR/owl-features>

³⁹ <http://protege.stanford.edu/>

⁴⁰ <http://www.ksl.stanford.edu/software/ontolingua/>

⁴¹ <http://www.opencyc.org/>

⁴² <http://oiled.man.ac.uk/>

⁴³ <http://www.ontoknowledge.org/tools/toolrep.shtml>

omogoča povpraševanje po ontologiji. Kljub temu da lahko ontologije obravnavamo kot samostojne specifikacije, se navsezadnje uporabljajo tudi za iskanje odgovorov na vprašanja, ki jih vsebuje ontologija. Nekateri urejevalniki celo omogočajo uporabo dodatnih aksiomov in pravil, vendar so rešitve na tem področju implementirane različno za vsako orodje, tako da trenutno ne obstaja standardni jezik za zapis pravil in neposredne manipulacije z izrazi in strukturami v ontologiji. Eden od zelo verjetnih kandidatov, ki bo podprt v naslednjih urejevalnikih ontologij, je prav gotovo RuleML.

10 Primer uporabe ontologije

V tem razdelku si bomo ogledali preprost primer uporabe ontologije, v katerem bomo za predstavitev znanja uporabili jezik OWL.

Predpostavimo, da je prišlo do oboroženega ropa, kjer ropar na kraju zločina izgubi orožje in pobegne v neznano [4]. Ko na kraj zločina prispe policija, zberejo ustrezne podatke, poberejo dokaze, kjer najdejo tudi izgubljenorožje. Poročilo je zapisano v obliki, prikazani na sliki 12.

```
<rop>
  <datum>...</datum>
  <opis>...</opis>
  <dokaz>
    <orožje>
      <serSt>1234-5678</serSt>
    </orožje>
  </dokaz>
  <ropar>
    <oseba /> <!-- neznan oseb -->
  </ropar>
</rop>
```

Slika 12: Poročilo o oboroženem ropu

Kasneje, v drugem primeru policijskega posredovanja, pride do situacije, kjer voznik prekorači dovoljeno hitrost vožnje, zato ga policist ustavi, mu napiše

```
<prehitraVoznja>
  <datum>...</datum>
  <opis>...</opis>
  <voznik>
    <oseba>
      <ime>Hitri Tone</ime>
      <stVozDov>S-1276209</stVozDov>
    </oseba>
  </voznik>
</prehitraVoznja>
```

Slika 13: Poročilo o prometnem prekršku

kazen ter na policijsko postajo pošlje poročilo o storjenem prekršku (slika 13).

Na policijski postaji analizirajo obe prejeti poročili. S pomočjo serijske številke orožja v svoji podatkovni zbirki najdejo osebo, na katero je to orožje registrirano. Sliki 14 in 15 tako v strnjeni obliki prikazujeta osnovne podatke obeh dogodkov.

```
<orozniList>
  <prijavljenoOrozje>
    <orožje>
      <serSt>1234-5678</serSt>
    </orožje>
  </prijavljenoOrozje>
  <lastnik>
    <oseba>
      <ime>Nevarni Janez</ime>
    </oseba>
  </lastnik>
</orozniList>
```

Slika 14: Orožni list najdenega orožja

```
<oseba>
  <ime>Hitri Tone</ime>
  <stVozDov>S-1276209</stVozDov>
</oseba>
```

Slika 15: Podatki o vozniku, ki je storil prometni prekršek

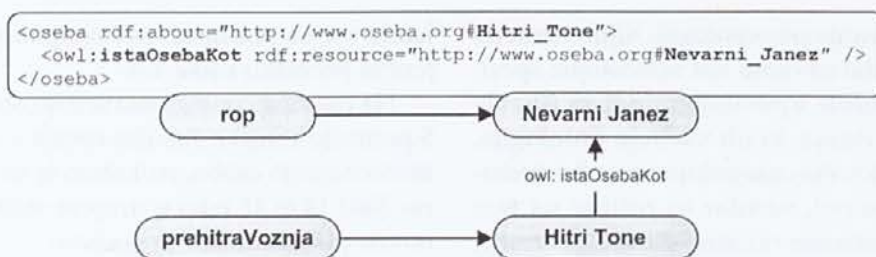
Pri pregledu kartoteke osebe Hitri Tone policisti pridejo do sklepa, da sta ropar in voznik, ki je prekoračil dovoljeno hitrost, ena in ista oseba.

Do tega sklepa jih je pripeljalo dejstvo, da se oseba Hitri Tone predstavlja tudi pod imenom Nevarni Janez, kar je zapisano v kartoteki Hitrega Toneta. To trditev lahko vidimo zapisano v jeziku OWL na sliki 16, kjer je tudi prikazan rezultat sklepanja.

Rop je torej izvedel Nevarni Janez, ki smo ga že prej povezali z najdenim orožjem, saj je le-to registrirano na njegovo ime. Voznik z imenom Hitri Tone, ki je storil prekršek, je pravzaprav ista oseba kot Nevarni Janez, kar je zapisano v njegovi kartoteki.

Vseeno pa moramo biti pred dokončno odločitvijo previdni, saj je treba pred aretacijo osebe, ki je prekoračila hitrost zaradi storjenega kaznivega dejanja ropa, odgovoriti na naslednja vprašanja:

1. Ima več vrst orožja lahko isto serijsko številko? Če bi ta trditev veljala, potem orožje, uporabljeno v ropu, ne bi nujno pripadalo osebi Hitri Tone. Vendar nam spodnji stavek OWL (glej sliko 17) pove,

Slika 16: **Sklepanje o roparju in vozniku s prometnim prekrškom**

- da je vsako orožje enolično določeno s svojo serijsko številko, torej serijska številka 1234-5678 pripada točno določenemu orožju.
2. Lahko različnim osebam pripadajo iste številke vozniškega dovoljenja? Če to velja, potem obstaja možnost, da podatki o registriranem orožju pripadajo nekomu drugemu. Vendar lahko podobno kot v prejšnjem primeru s spodnjo izjavo OWL (slika 18) opredelimo, da številka vozniškega dovoljenja nedvoumno določa osebo. S tem je zagotovljeno, da številka vozniškega dovoljenja S-1276209 pripada točno določeni osebi, in sicer Hitremu Tonetu.

3. Obstaja možnost registracije orožja v različnih orožnih listih? Če to velja, se lahko zgodi, da je lastnik orožja nekdo drug in ne Hitri Tone. Podobno kot smo to trditev ovrgli pri prejšnjih vprašanjih, to storimo tukaj, saj je lahko vsako orožje registrirano le v enem orožnem listu. Dokaz je trditev OWL na sliki 19.
4. Ali se na orožnem listu lahko nahaja več lastnikov več registriranih orožij? Spodnja izjava OWL nam pove, da vsak orožni list velja za eno orožje in eno osebo, tako da ni dvoma, da je oseba, ki je prekoračila največjo dovoljeno hitrost, ista kot tista, ki je izvedla rop.

```

<owl:InverseFunctionalProperty rdf:ID="serSt">
  <rdfs:domain rdfs:resource="#orozje" />
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal">
</owl:InverseFunctionalProperty>

```

Slika 17: **Serijska številka kot enolični identifikator orožja**

```

<owl:InverseFunctionalProperty rdf:ID="stVozDov">
  <rdfs:domain rdfs:resource="#oseba" />
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal">
</owl:InverseFunctionalProperty>

```

Slika 18: **Številka vozniškega dovoljenja je za osebo enolična**

```

<owl:InverseFunctionalProperty rdf:ID="prijavljenoOrozje">
  <rdfs:domain rdfs:resource="#orožniList" />
  <rdfs:domain rdfs:resource="#orozje" />
</owl:InverseFunctionalProperty>

```

Slika 19: **Vsako orožje je povezano le z enim orožnim listom**

```

<owl:Class rdf:ID="orožniList">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Restriction>
      <owl:onProperty rdf:resource="#prijavljenoOrozje" />
      <owl:cardinality>1</owl:cardinality>
    </owl:Restriction>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#lastnik" />
      <owl:cardinality>1</owl:cardinality>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>

```

Slika 20: **Števnost pri orožnem listu**

Kot vidimo, lahko vse odgovore na zgoraj zastavljena vprašanja dobimo s pomočjo ontologije OWL o registraciji orožja. V ontologiji so opredeljeni vsi potrebni koncepti in njihove medsebojne odvisnosti, na podlagi katerih lahko izvajamo samodejna sklepanja, kot smo to storili v primeru roparja in voznika, ki je storil prometni prekršek.

11 Sklep

Dodana vrednost uporabe ontologij in semantičnega spleta se skriva v dejstvu, da lahko semantiko svojih podatkov, množice dokumentov in sistemov opišemo z enotnim semantičnim pripomočkom, ki ga razumejo tudi računalniški sistemi. Vse, kar smo razvili v preteklosti, lahko ponovno uporabimo in uvedemo ontologijo za določeno področje, to pa kasneje razširimo in omogočimo njeno souporabo ter tako pripravimo skupno semantiko za celotno organizacijo.

Zakaj torej sploh potrebujemo ontologije?

- S pomočjo ontologije lahko domneve o problemski domeni formalno zapišemo, s čimer lažje obvladujemo domneve in razumemo obstoječe podatke.
- Operativno znanje ločimo od znanja problemske domene, tako da lahko ti dve vrsti znanja uporabljamo in obvladujemo ločeno.
- Skupni pomen informacij lahko zaradi konsistentnega razumevanja souporabljamo na različnih mestih.

Z veliko razširjenostjo ontologij v zadnjem času lahko s pomočjo informatike naredimo velik korak naprej v komunikaciji računalniških sistemov s človekom. Komunikacija tako poteka na višji konceptualni ravni, kjer človek ni prisiljen komunicirati na ravni računalnika. Rast produktivnosti zaradi izmenjave »pomena« med človekom in računalnikom namesto podatkov brez semantične dodane vrednosti zagotovo pomeni manjšo revolucijo na področju računalništva. Ovira za še večjo priljubljenost leži tudi v uporabnikih samih, ki računalnikom in tehnologiji še vedno ne zaupajo dovolj. Dvom obstaja predvsem pri zaupanju ob računalniškem sprejemanju odločitev na podlagi znanja, ki ga imajo na voljo (npr. v obliki on-

tologij). Upajmo, da se bo to zaupanje povečalo s prehodom znanja iz akademskih krogov v industrijo v obliki uspešno izpeljanih projektov, kjer bo vidna dodana vrednost uporabe ontologij. Druge razloge za še večjo priljubljenost lahko iščemo tudi v zrelosti področja, saj je še relativno novo in v pomanjkanju orodij, s katerimi bi lahko bolje in lažje obvladovali kompleksnost ontologij.

12 Viri in literatura

- [1] SPYNS, Peter, MEERSMAN, Robert, JARRA, Mustafa: Data modeling versus Ontology engineering, ACM SIGMOD Record, Volume 31, Issue 4, 2002.
- [2] DACONTA, Michael C., OBRST, Leo J., SMITH, Kevin T.: The Semantic Web: A Guide to the future of XML, Web Services, and Knowledge Management, Wiley publishing, 2003.
- [3] DENNY, Michael: Ontology Building: A Survey of Editing Tools, <http://www.xml.com/pub/a/2002/11/06/ontologies.html>, 2004.
- [4] IJS, Semantically-Enabled Knowledge Technologies, <http://seminars.ijs.si/sekt>, 2005.
- [5] Inštitut za slovenski jezik Frana Ramovša ZRC SAZU, Slovar slovenskega knjižnega jezika, elektronska izdaja, verzija 1.0, DZS.
- [6] Merriam-Webster Inc., Merriam-Webster OnLine dictionary, <http://www.m-w.com>, 2005.
- [7] RIBIÈRE Myriam, CHARLTON, Patricia: Ontology Overview, Networking and Applications Lab, Centre de Recherche de Motorola-Paris.
- [8] DAVIES, John, FENSEL, Dieter, HARMELEN, Frank Van: Towards The Semantic Web – Ontology-driven Knowledge Management, 2003.
- [9] NOY, Nataly F., MCGUINNESS, Deborah L.: Ontology Development 101: Guide to creating your first ontology, Stanford University, http://protege.stanford.edu/publications/ontology_development/ontology101.html, 2001.
- [10] USCHOLD, Michael, GRUNINGER, Michael: Ontologies and Semantics for Seamless Connectivity, SIGMOD Record, Volume 33, Issue 4, 2004.
- [11] TAMMA, Valentina, PHELPS, Steve, DICKINSON, Ian, WOOLDRIDGE, Michael: Ontologies for supporting negotiation in e-commerce, Engineering Applications of Artificial Intelligence 18, str. 223 – 236, 2005.
- [12] COSENTINO, Massimo, BERNON, Carole, PAVÒN, Juan: Modelling and Meta-modelling Issues in Agent Oriented Software Engineering: The AgentLink AOSE TFG Approach, Ljubljana, 2005.
- [13] DACONTA, Michael C.: Designing the Smart-Data Enterprise, Semantic (XML Web Services) Interoperability Community of Practice (SICoP), <http://web-services.gov/>, 2003.
- [14] GEROIMENKO, Vladimir: Dictionary of XML Technologies and the Semantic Web, Springer, 2004.

Dejan Lavbič je diplomiral na Fakulteti za računalništvo in informatiko Univerze v Ljubljani, kjer je zaposlen kot mladi raziskovalec. Na raziskovalnem področju se ukvarja z inteligentnimi sistemi, večagentnimi sistemi, uporabo ontologij in odkrivanjem zakonitosti v podatkih.

Marjan Krisper je docent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Vodi številne projekte razvoja informacijskih sistemov, elektronskega poslovanja in metodologij razvoja informacijskih sistemov v največjih sistemih v gospodarstvu, državni upravi in javnem sektorju. Je ustanovni član mednarodnega združenja za informacijske sisteme AIS (Association of Information Systems), član izvršnega odbora Slovenskega društva INFORMATIKA in član Slovenskega društva za umetno inteligenco.