

Pravila na semantičnem spletu

Dejan Lavbič, Marko Bajec, Marjan Krisper

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Tržaška 25, 1000 Ljubljana, Slovenija
E-pošta: Dejan.Lavbic@fri.uni-lj.si, Marko.Bajec@fri.uni-lj.si, Marjan.Krisper@fri.uni-lj.si

Povzetek. Jeziki za predstavitev pravil igrajo vedno večjo vlogo pri uveljavljanju semantičnega spleta. Po eni strani pravila omogočajo zapis deklarativnega znanja pri jezikih za predstavitev ontologij in izražajo omejitve ali pretvorbe, ki so povezane z opisno logiko. Po drugi strani pa lahko s pravili opredelimo uveljavljanje politik, odzivanje na spremembe, specifikacijo delovanja inteligentnih agentov in jih uporabimo še za številne druge namene. Jeziki za predstavitev pravil bodo tako igrali ključno vlogo, saj bodo kot lingua franca za izmenjavo pravil med različnimi sistemi in orodji ter bodo omogočali razvijanje, izvajanje, objavljane in medsebojno komunikacijo pravil na semantičnem spletu. V članku je predstavljeno ogrodje za predstavitev znanja na semantičnem spletu, kjer je poudarek na integraciji ontologij in pravil. Povezovanje je izpeljano s pomočjo pretvornikov, saj trenutno ni podpore izvajanju pravil v obliki SWRL, ki je eden najobetajočih predlogov pravil na semantičnem spletu.

Ključne besede: semantični splet, pravila, RuleML, SWRL

Semantic Web Rule Languages

Extended abstract. Semantic Web rule languages play an important role at putting forward Semantic Web and are gaining their support. Rules can be used for describing declarative knowledge of ontology languages and to express restrictions or transformations that can include the portion of description logic. On the other hand rules can express policy enforcement, define responses to changes, specify intelligent agent behaviours and be used for other purposes. Therefore Semantic Web rule languages will play an important role of being lingua franca for interchanging rules among different systems and tools, enabling us to develop, run, publish and interact among Semantic Web rules. In the article we present a framework for knowledge representation that uses ontologies and rules. The main focus is on dynamic, heterogenous environment and use of knowledge base where ontologies and rules are combined extending OWL DL expressiveness. Merging phase is supported by converters, because there are no reasoning engines available at the time that support latest SWRL proposal.

Key words: Semantic Web, rules, RuleML, SWRL

1 Uvod

V ožjem smislu jezik za predstavitev pravil na semantičnem spletu razumemo kot konkretno sintakso, temelječo na XML.¹ V širšem pomenu pa ga

Prejet ???

Odobren ???

¹eXtensible Markup Language - Razširljivi označevalni jezik, ki predstavlja format podatkov za izmenjavo strukturiranih dokumentov v spletu.

razumemo kot jezik z abstraktno sintakso, ki je namenjen predvsem za skupno osnovo pri opredeljevanju različnih izpeljanih jezikov za različne potrebe.

Ti jeziki morajo imeti formalno semantiko, vendar lahko kljub temu obstajajo določeni konstrukti, ki nimajo formalne semantike, izpeljane iz logike prvega reda, ampak jih vseeno potrebujemo pri reševanju določenih nalog. V zvezi s tem problemom se moramo zavedati dveh nevarnosti:

- Uporaba praktičnih konstruktorjev jezika, ki se zdijo pomembni in sicer imajo določeno intuitivno, a ne formalno, semantiko, čeprav imamo na voljo alternative, ki imajo formalno (mogoče nestandardno) semantiko.
- Posvečanje prevelike pozornosti teoretičnim vprašanjem standardne logike prvega reda, kot je izračunljivost, kompleksnost, odločljivost in jedratost.

Pri standardizaciji jezikov za predstavitev pravil na svetovnem spletu je potrebno upoštevati uporabnike SQL-a, Prolog-a in ostalih sistemov (CLIPS, JESS, ILOG itd.), saj mora obstajati povezava med različnimi pomembnejšimi koncepti omenjenih jezikov.

V prispevku bomo predstavili nekaj načinov, kako lahko sisteme za izvajanje poslovnih pravil povežemo

s semantičnim spletom. V razdelku 2 je najprej podrobneje predstavljen semantični splet in potreba po pravilih. Sledi opis predlaganega ogrodja za predstavitev znanja na semantičnem spletu, ki temelji na integraciji ontologij in pravil. Različni predlogi jezikov za predstavitev pravil na semantičnem spletu so zbrani v razdelku 3, medtem ko v razdelku 4 predstavimo podporo omenjenim jezikom v dejanskih grafičnih orodjih.

2 Semantični splet in pravila

Svetovni splet doživlja velik uspeh, kar nakazuje število uporabnikov in dokumentov, 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 povezav med elementi na spletu. Pri iskanju podatkov na svetovnem spletu moramo tako poznati bodisi natančen naslov iskanega dokumenta, bodisi ga poiščemo s pomočjo spletnega iskalnika. Problem spletnih iskalnikov pa je 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 podatki dobijo pomen, kar omogoča boljše medsebojno sodelovanje ljudi in računalnikov [6]. Moč semantičnega spleta je v združevanju tehnologij za obvladovanje znanja na področju umetne inteligence in spletnih tehnologij.

Veliko pripomb s strani širše javnosti je bilo na vse preveč standardov, ki so na voljo na svetovnem spletu. Očitki so se nanašali predvsem na ne dovolj jasne vizije posameznih predlogov. Verjamemo pa, da uporaba pravil na področju semantičnega spleta prinaša nove možnosti sodelovanja in souporabe XML, RDF,¹ OWL² in spletnih storitev, z namenom rešiti problem integracije podatkov na svetovnem spletu.

2.1 Integracija podatkov

Vedno več podatkov, ki so na voljo na svetovnem spletu, se pojavlja v XML obliki. Razlog za to je preprost - enostaven prenos delno strukturiranih podatkov, ki s shemami prinašajo interoperabilnost in z XSLT omogočajo pretvorbo med različnimi

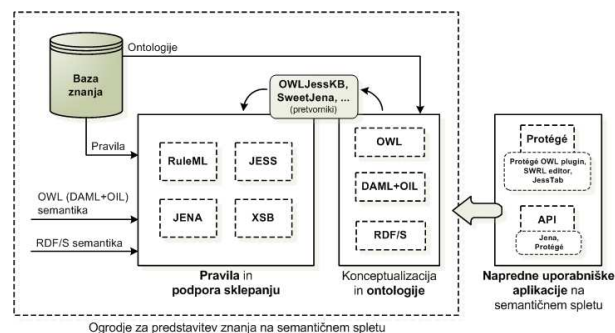
¹Resource Definition Framework - S pomočjo jezika RDF predstavimo semantične relacije na ravni primerkov v obliki trojčkov (oseba, predikat, predmet).

²Web 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.

XML formati. Vendar nam XML sam po sebi ne ponuja rešitve integracije podatkov. Številna podjetja še vedno niso razvila svojih shem za poenostavitev izmenjave, pogosto pa izmenjava zahteva tudi združevanje podatkov iz različnih, porazdeljenih, virov, brez skupnih shem ali shem brez jasno določenih preslikav. Tako še vedno ostaja problem npr. pri iskanju odgovora na vprašanje, kdaj sta dva podatka približno enaka oz. podobna. Če omogočimo dostop do podatkov na svetovnem spletu preko URI³ naslovov, smo naredili šele prvi korak pri integraciji podatkov.

2.2 Uporaba pravil in ontologij

Rešitev prej omenjenega problema lahko najdemo v uporabi pravil na semantičnem spletu. Pravila skupaj z ontologijami prav gotovo prinašajo višjo raven spletnih storitev, kjer s konvergenco pridemo do semantičnih spletnih storitev. Primere uporabe lahko najdemo v poslovnih okoljih, še zlasti uporaba agentov⁴ v elektronskem poslovanju, ki bo temeljilo na pridobljenem znanju. Vloga agentov bo zbiranje informacij, sklepanje in optimizacija stroškov. Izvajali bodo tudi pogajanja z ustreznno mero avtonomije. Prvi korak k dosegu teh idej pa je zmožnost komunikacije z dovolj veliko mero skupno določenega pomena, seveda s pomočjo semantičnega spleta, ki vključuje uporabo pravil in ontologij.



Slika 1. Uporaba pravil in ontologij na semantičnem spletu

Slika 1 prikazuje ogrodje za predstavitev znanja na semantičnem spletu. Glavna ideja se skriva v integraciji pravil in ontologij ter uporabi baze znanja s strani naprednih uporabniških aplikacij.

Z uporabo pravil se na semantičnem spletu srečamo z novimi težavami, ki jih prej nismo imeli.

³Uniform Resource Identifier - Splošen identifikator, ki se uporablja za razlikovanje konkretnih in abstraktnih virov na svetovnem spletu.

⁴Za 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.

Zato se moramo zavedati naslednjih ključnih elementov pri vpeljavi in uporabi pravil na tem področju:

1. **Raznolikost.** Obstoječe in nastajajoče ontologije ter pravila uporabljajo heterogene oblike zapisovanja znanja, zato potrebujemo mehanizme, ki bodo sposobni obvladovati te raznolikosti.
2. **Porazdeljenost** obvladovanja pravil in ontologij pri uporabi baz znanj.
3. **Neskladnost.** Pri združevanju znanja pride do konfliktnih situacij, zato moramo imeti ustrezne mehanizme, s katerimi to obvladujemo.
4. **Dinamičnost.** Posodabljanj baz znanj bo vedno več, zato se moramo znati spopasti s to povečano frekvenco.
5. **Zakasnitev.** Razširljivost je odločitvenega pomena pri doseganju integracije znanja, kjer se želimo približati linearnemu času dostopa (podobno kot pri podatkovnih bazah).

Ob upoštevanju zgornjih izhodišč smo se odločili za pristop, kjer je baza znanja produkt integracije ontologij in pravil (glej sliko 1). Pri procesu združevanja so uporabljeni pretvorniki, ki ustrezne koncepte in relacije iz ontologij preslikajo v sisteme za izvajanje pravil, tako da lahko uporabimo obstoječe sisteme za izvajanje poslovnih pravil. Celotno ogrodje je tako na voljo naprednim aplikacij, ki uporabljajo storitve semantičnega spleta.

3 Jeziki za predstavitev pravil na semantičnem spletu

V zadnjem času se je pojavilo kar nekaj predlogov jezikov za predstavitev pravil na semantičnem spletu in v nadaljevanju si bomo podrobno ogledali naslednje:

- Rule Markup Language (RuleML)¹
- OWL Rules Language (ORL)²
- Semantic Web Rule Language (SWRL)³

3.1 RuleML

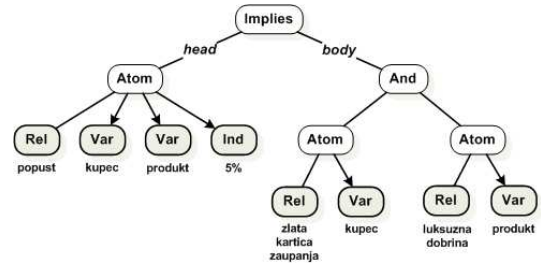
RuleML jeziki imajo potencial, da postanejo glavni akterji na področju uporabe pravil na semantičnem spletu in v porazdeljenih sistemih, saj omogočajo kreiranje, procesiranje, objavljanje in medsebojno komunikacijo pravil. Sintaksa RuleML temelji na

¹<http://www.ruleml.org>

²<http://www.cs.man.ac.uk/~horrocks/DAML/Rules>

³<http://www.daml.org/2003/11/swrl>

XML-ju, čeprav obstaja del njegove sintakse tudi v RDF obliki.



Slika 2. Primer RuleML pravila v obliki drevesa

Primer pravila v RuleML obliki si lahko ogledamo na sliki 2, kjer imamo pravilo prikazano v obliki drevesa z ustreznimi gradniki, ki jih definira RuleML. V naravnem jeziku se pomen zgornjega pravila glasi: Če ima kupec zlato kartico zaupanja in produkt spada med luksuzne dobrine, potem se kupcu pri nakupu upošteva 5% popust. XML zapis omenjenega drevesa pa lahko vidimo na sliki 3.

```
<Implies>
  <head>
    <Atom>
      <Rel>popust</Rel>
      <Var>kupec</Var>
      <Var>produkt</Var>
      <Ind>5%</Ind>
    </Atom>
  </head>
  <body>
    <And>
      <Atom>
        <Rel>zlata kartica zaupanja</Rel>
        <Var>kupec</Var>
      </Atom>
      <Atom>
        <Rel>luksuzna dobrina</Rel>
        <Var>produkt</Var>
      </Atom>
    </And>
  </body>
</Implies>
```

Slika 3. Primer RuleML pravila v XML obliki

Cilj RuleML je zagotoviti ponovno uporabljivost in izmenjavo na višji ravni, podobno kot pri MDA.⁴ Namesto da ustvarimo še en jezik za opis pravil, RuleML ponuja množico izpeljanih modularnih jezikov, ki temeljijo na skupnem podatkovnem modelu.

Pogled na pravila si lahko analogno MDA pristopu predstavljamo na treh ravneh:

1. Na **poslovni ravni** pravila v obliki izjav opredeljujejo določen del poslovnega procesa ali strukturno omejitev. Za predstavitev se

⁴Model-Driven Architecture je arhitektura 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 potrebno model PIM prevesti v model, ki je specifičen za izbrano platformo (PSM).

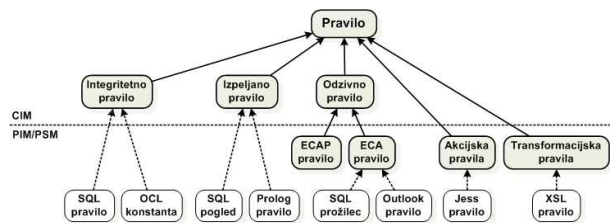
uporablja deklarativna oblika, ki je največkrat naravni jezik ali kakšna grafična opisna oblika. Primeri pravil v naravnem jeziku so:

- P_1 : Oseba, ki želi opravljati vozniški izpit, mora biti stara najmanj 18 let.
- P_2 : Dobiček na investicijo ni obdavčen, če je od nakupa delnice minilo že več kot 3 leta.
- P_3 : Ko cena delnice pade za več kot 5% in za investicijo ni potrebno plačati davka, jo prodaj.

2. Na **ravni neodvisnosti od platforme** so pravila izražena v obliki formalnih izjav, katere lahko kasneje povežemo z izvajalnimi izjavami dejanske platforme. Jeziki za zapis pravil na tej ravni so SQL:1999, OCL 2.0, ISO Prolog idr.

3. **Raven platformno odvisnih pravil** vsebuje pravila, zapisana z izjavami v določenem izvajalnem jeziku, kot so prožilci v podatkovni bazi Oracle 9, JESS ali pravila v orodju Microsoft Outlook.

Zavedati se moramo, da imamo v poslovni domeni opravka z jezikoslovno bogato in zapleteno izrazno obliko, ki jo lahko v formalni matematični obliki zapišemo le v omejenem obsegu. Pri RuleML je zato uporabljen pristop, kjer imamo množico jezikov za opis najpomembnejših tipov pravil.



Slika 4. Kategorizacija pravil

V nadaljevanju si bomo nekoliko bolj podrobno ogledali osnovne tipe pravil, ki jih predlaga RuleML in jih lahko vidimo na sliki 3: *integritetna*, *izpeljana*, *odzivna*, *akcijska* in *transformacijska pravila*.

3.1.1 Integritetna pravila

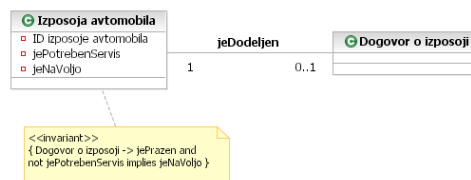
Integritetno pravilo sestavlja logični izraz, zapisan v poljubnem logičnem jeziku (npr. predikatna logika). Pravilo predstavlja izjavo, ki mora biti resnična v vseh nastalih stanjih in prehodih med stanji v diskretnem dinamičnem sistemu, za katerega je definirano. Pravilo P_1 iz razdelka 2 je primer

statičnega integritetnega pravila, medtem ko bi dinamično integritetno pravilo lahko bilo: Potrditvi rezervacije vozila mora slediti rezervacija vozila zahtevanega razreda za dan prej, kot je zahtevan datum.

Priljubljena jezika za izražanje integritete sta SQL in OCL.

3.1.2 Izpeljana pravila

Izpeljano pravilo je sestavljeno iz enega ali več pogojev in sklep. V začetku razdelka smo omenili pravilo P_2 , ki ga lahko uvrstimo med izpeljana pravila. Zelo znane oblike izpeljanih pravil lahko najdemo v obliki Prolog pravil in SQL pogledov. To vrsto pravil lahko prikazemo tudi v UML razrednih diagramih s pomočjo OCL konstant, kot je prikazano na sliki 4.



Slika 5. Opredelitev izpeljanega pravila s pomočjo OCL

V naravnem jeziku si lahko omenjeno pravilo predstavljamo kot: *Avto lahko najamemo, če ni povezan z nobenim dogovorom o izposoji in zanj ni potreben servis.*

3.1.3 Odzivna pravila

Odzivno pravilo sestavlja prožilni dogodek, pogoj, odzivna akcija in včasih tudi popogoj. Odzivna pravila delimo glede na prisotnost popogojev v dve skupini: ECA⁵, ki nimajo popogojev in ECAP⁶, kjer so popogoji prisotni. To kategorizacijo prikazuje tudi slika 4, kjer sta omenjena tudi pogosto uporabljena primera teh pravil - SQL prožilec in Outlook pravilo.

3.1.4 Akcijska pravila

Akcijsko pravilo preprosto sestavlja pogoj in akcija, ki se ob izpolnjenem pogoju izvede. S to vrsto pravil lahko implementiramo tudi izpeljana pravila. Na področju poslovnih pravil so akcijska pravila ena najpomembnejših in tudi najpogosteje uporabljenih ter predstavljajo ključno komponento v RuleML družini jezikov. Primeri znanih izvajalnih okolij akcijskih pravil so JESS, iLOG Rules/JRules, Fair Isaac/Baze Advisor itd.

⁵Event-Condition-Action

⁶Event-Condition-Action-Postcondition

3.2 ORL

Številne pomankljivosti jezika OWL izhajajo iz dejstva, da je jezik relativno bogat pri izražanju razredov, medtem ko se veliko slabše odreže pri lastnostih. Preprosto ne obstaja konstruktor, s katerim bi lahko določili relacije med sestavljenimi lastnostmi in drugimi, mogoče celo sestavljenimi, lastnostmi. Enostaven primer, ki opisuje ta problem, je relacija med lastnostmi starš, brat in stric, ki jo v osnovni obliki OWL jezika ne moremo zapisati.

Eden od načinov reševanja omenjenega problema je razširitev jezika OWL z bolj izraznim jezikom za opis lastnosti. Dodajanje pravil jezikom za predstavitev znanja, ki temeljijo na opisni logiki, ni ravno nov pristop. Kar nekaj predhodnjih sistemov (npr. Classic), ki temeljijo na opisni logiki, so svojo izrazno moč povečali z uporabo jezika za predstavitev pravil [9]. Vendar so pravila v teh sistemih imela šibkejšo semantično moč, kot jo imamo pri vpeljavi relacij med razredi in nadrazredi, saj so bile relacije prisotne le med posameznimi primerki in niso vplivale na sklepanje na podlagi razredne hierarhije.

ORL razširja OWL v sintaktičnem in semantičnem smislu: osnovna sintaksa ORL pravil je razširitev abstraktne sintakse OWL DL in OWL Lite, pravila pa imajo tudi formalen pomen v povezavi z razširitvijo OWL DL.

Ko se odločamo za uporabo OWL pravil, moramo izpostaviti naslednje prednosti [4]:

- Glede na OWL ne predstavljajo bistvenih sprememb, tako da so tudi relativno enostavna za uporabo.
- OWL pravila imajo veliko izrazno moč, saj temeljijo na izrazni moči OWL-ja pri obeh elementih pravil: vzroku (telo) in posledici (glava)
- Obljubljajo zelo elegantno rešitev raznovrstnega semantičnega procesiranja. Raziskovalnim skupinam, ki se ukvarjajo s semantičnim spletom, je jasno, da pri realizaciji ideje semantičnega spleta ne bo dovolj zgolj ena tehnika, model ali metoda. Potreba po različnih načinih uporabe ne bo predstavljala izbire, ampak bo nujno prisotna.

3.3 SWRL

Čeprav z uporabo OWL jezika pridobimo veliko izrazno moč na semantičnem spletu, vseeno obstajajo določene omejitve pri izražanju. Te omejitve pa lahko odpravimo z uporabo **SWRL** jezika za predstavitev pravil na semantičnem spletu. SWRL je nastal na temeljih ORL, ki smo ga predstavili v prejšnjem razdelku. Ideja za SWRL se je pojavila

še leta 2004 in je eden novejših predlogov jezika za opis pravil na semantičnem spletu.

Glavna ideja, ki stoji za SWRL, je razširitev OWL DL s pravili, kjer želimo obdržati čimvečjo združljivost nazaj z obstoječo sintakso in semantiko. Pristop uvaja Hornovo obliko pravil, ki razširjajo OWL sintakso. Pravila so preprosto v obliki implikacije med vzroki (telo) in posledicami (glava). Pomen pravila pa lahko preberemo kot: vedno ko je zadoščeno pogojem v telesu, morajo držati tudi posledice v glavi.

SWRL zagotavlja zelo uporabno raven v hierarhiji semantičnega spleta, vendar se moramo zavedati, da bodo nekatere aplikacije potrebovale še večjo izrazno moč. Pri nekaterih aplikacijah je recimo pomembno izražanje aritmetičnih relacij med posameznimi vrednostmi (npr. da osebo, ki ima prihodke enake ali višje kot odhodke, označimo kot srečno, medtem ko pri osebah, kjer odhodki presegajo prihodke, načeloma velja, da so nesrečne). Še vedno ni jasno, ali bi bilo mogoče to funkcionalnost doseči z nadaljnjo razširitvijo funkcionalnosti jezika SWRL ali z razširitvijo OWL podatkovnih tipov [9].

4 Podpora jezikom v orodjih

Podpora jezikom za predstavitev pravil na semantičnem spletu v razvojnih orodjih je zelo slaba. Razlogi za to so prav gotovo v problemu zrelosti, saj so rešitve oz. predlogi še zelo novi.

4.1 Protégé

Eno redkih orodij, s pomočjo katerega lahko v grafičnem načinu obvladujemo pravila, zapisana v SWRL obliki, je SWRL editor.¹ Na voljo je kot vključek za Protégé² in nam omogoča enostavno obvladovanje pravil v okviru OWL ontologije. Koncepte, lastnosti in primerke ontologije lahko intuitivno vključujemo v pravila, ki jih ustvarjamo ali urejamo, prav tako pa lahko vse dele ustrezno opišemo. V SWRL editorju je prisotno tudi osnovno sintaktično in semantično preverjanje, vendar moramo za integriteto pravil skrbeti sami, saj nam sistem ne more preprečiti pisanja nesmiselnih pravil. Pravila so na koncu shranjena v OWL/RDF obliki baze znanja, tako da je možna uporaba teh pravil in spremljajoče ontologije tudi v drugih orodjih. Za večjo razširljivost je na voljo API, s pomočjo katerega lahko iz programskega jezika Java dostopamo do baze znanja in shranjenih pravil. Največja negativna lastnost je to, da trenutno ne obstaja sistem za izvajanje

¹<http://protege.stanford.edu/plugins/owl/swrl>

²<http://protege.stanford.edu>

teh pravil, ampak je na voljo zgolj enostaven mehanizem za sklepanje, ki deluje na podlagi OWL konceptov in medsebojnih relacij, ne pa pravil, zapisanih v SWRL obliki. Vendar pa lahko pravila vseeno uporabimo s pomočjo pristopa, ki smo ga omenili v razdelku 2.2.

4.2 SweetRules

Zanimiv pristop predstavlja rešitev SweetRules³, ki jo avtorji označujejo kot prvo odprtokodno platformo za poslovna pravila na semantičnem spletu. Paket SweetRules prinaša številne uporabne lastnosti:

- pretvorba med različnimi heterogenimi sistemi za izvajanje pravil (forward in backward chaining)
- sklepanje s povezavo z različnimi obstoječimi sistemi
- obvladovanje in testiranje baze pravil

S pomočjo SweetRules lahko obvladujemo različne tehnologije na področju pravil in semantičnega spleta, kot so RuleML, XSB, Jess, IBM Common-Rules, Knowledge Interchange Format (KIF), OWL, Process Handbook, Smodels, Jena-2, SWRL.

5 Sklep

Z razvojem semantičnega spleta se je razširila tudi uporaba ontologij, kot formalizma za opis znanja in podatkov. Le te so predstavljene v takšni obliki, da jih lahko souporabljamo na svetovnem spletu. Širša uporaba jezika OWL za predstavitev znanja na semantičnem spletu prinaša nove možnosti za razvoj zahtevnejših aplikacij na različnih problemskih področjih. Vendar je koristnost ontologij omejena z mehanizmi procesiranja, ki so integrirani v takšno obliko predstavitve. Ravno zaradi tega se veliko odvija na področju formalizacije logičnega sloja ontologij. Jezik za predstavitev pravil na semantičnem spletu (SWRL) je pomemben korak v tej smeri in temelji na predhodnjih raziskavah RuleML in ORL. Navsezadnje bomo šele z razpoložljivostjo standardiziranega jezika za zapis pravil lahko učinkovito uporabljali ontologije in pravila pri razvoju inovativnih aplikacij, ki uporabljajo semantični splet. Za uspešen razvoj in uporabo teh tehnologij je ključnega pomena razumevanje zmožnosti in medsebojnega delovanja pravil in ontologij.

³SWEET = Semantic WEB Enabling Tools, <http://sweetrules.projects.semwebcentral.org>

6 Literatura

- [1] H. Halpin, Rules: Enabling Data Integration using Semantic Web, *W3C Rule Workshop Proposal*, Marec 2005.
- [2] E. Miller: Rules and Semantic Web, *W3C Workshop on Rule Languages for Interoperability*, Washington, DC, ISA, April 2005.
- [3] I. Horrocks, P. F. Patel-Schneider: A Proposal for an OWL Rules Language, <http://www.cs.man.ac.uk/~horrocks/DAML/Rules/>, Oktober 2003.
- [4] K. Grant Clark: A Web of Rules, <http://www.xml.com/pub/a/2003/10/23/iswc.html>, 2003.
- [5] K. Taveter, G. Wagner: Agent-Oriented Enterprise Modeling Based on Business Rules, *Proc. of 20th Int. Conf. on Conceptual Modeling (ER2001)*, Springer-Verlag, LNCS 2224, pp. 527-650, 2001.
- [6] M. C. Daconta, L. J. Obrst, T. K. Smith: The Semantic Web: A Guide to the future of XML, Web Services, and Knowledge Management, Wiley publishing, 2003.
- [7] B. Groszof, M. Dean: Semantic Web Rules with Ontologies, and their E-Business Applications, *3rd International Semantic Web Conference*, Hiroshima, November 2004.
- [8] M. Hatala, R. Wakkary, L. Kalantari: Rules and ontologies in support of real-time ubiquitous application, *Journal of Web Semantics: Science, Services and Agents on the World Wide Web, volume 3, issue 1*, Elsevier, pp. 5-22, 2005.
- [9] I. Horrocks, P. F. Patel-Schneider, S. Bechhofer, D. Tsarkov: OWL rules: A proposal and prototype implementation, *Journal of Web Semantics: Science, Services and Agents on the World Wide Web, volume 3, issue 1*, Elsevier, pp. 23-40, 2005.

Dejan Lavbič je leta 2004 diplomiral na področju računalništva in informatike na Univerzi v Ljubljani, Fakulteta za računalništvo in informatiko, kjer je trenutno tudi zaposlen kot mladi raziskovalec. Na raziskovalnem področju se ukvarja z inteligentnimi agenti, večagentnimi sistemi, ontologijami, odkrivanjem zakonitosti v podatkih in razvojem mobilnih aplikacij.

Marko Bajec je docent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Diplomiral je iz računalništva leta 1996. Isto leto se je vpisal na podiplomski študij računalništva. Magistriral je leta 1998, doktoriral pa leta 2001. V okviru Katedre za informatiko se ukvarja z razvojnimi tehnologijami in razvojem ter prenovitvijo informacijskih sistemov. Marko Bajec je član društva Slovenska informatika, združenja AIS (Association for Information Systems) in član programskega odbora posvetovanja Dnevi slovenske informatike.

Marjan Krisper je docent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani in predstojnik katedre za informatiko. 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.