SÉMANTICKÝ WEB A AUTOMATICKÉ GENEROVANIE JEHO OBSAHU

Martin Švihla
Ivan Jelínek

ČVUT FEL Praha, Karlovo náměstí 13, 121 35 Praha 2, ČR
svihlm1@fel.cvut.cz, jelinek@fel.cvut.cz

Abstrakt
Sémantický Web je iniciatíva, ktorej cieľom je pridať do webových stránok počítačmi zrozumiteľný obsah. To má umožniť počítačom rozumieť informáciám na Webe a tak zlepšiť spoluprácu strojov a ľudí. Článok uvádza do technológie Sémantického Webu, zoznamuje s problematikou generovania metadát pre Sémantický Web priamo z dátového zdroja a predstavuje *METAmorphoses*, model mapovania SQL do RDF.

1. Úvod

World Wide Web sa stal v poslednom čase obrovským zdrojom informácií. Obsahuje milióny stránok o najrôznejších témach. Toto množstvo, ktoré je jednou z najväčších predností Webu, sa paradoxne stáva problémom. Aj napriek zlepšovaniu vyhľadávacích strojov, je nájdenie relevantných informácií na Webe čoraz ťažšie. Rovnako je čoraz obtiažnejšie udržiavanie veľkých informačných skladov či efektívna navigácia v nich.

Tieto problémy komplikuje skutočnosť, že obsah Webu je určený hlavne pre ľudí. Webovým stránkam softwaroví agenti, ani vyhľadávacie stroje nerozumejú. To napríklad znamená, že nám s vyhľadávaním veľmi nepomôžu. Inými slovami, pre počítače sú webové stránky len dáta, nie informácie. Existuje niekoľko iniciatív na zlepšenie tejto situácie. Jednou z nich je Sémantický Web.

2. Sémantický Web

Myšlienkou Sémantického Webu je pridať do webových stránok štruktúru a počítačmi zrozumiteľný význam (sémantika). Sémantický Web ale nie je oddelený od toho súčasného. Je jeho rozšírením, v ktorom je informáciám priradený dobre definovaný význam, umožňujúci lepšiu spoluprácu medzi počítačmi a ľuďmi [1].

2.1 Technológie

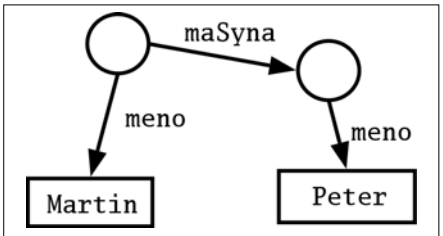
Tri základné technológie Sémantického Webu sú XML (eXtensible Markup Language) [2], RDF (Resource Description Framework) [3] a ontológie. Použitie XML pridá do dokumentov štruktúru, no táto štruktúra nevyjadruje význam obsahu.

Za účelom zachytenia významu bolo vytvorené RDF. RDF používa XML na zápis tripletov. Pomocou týchto tripletov sa význam vyjadruje ako jednoduché vety pozostávajúce z podmetu, prísudku a predmetu. Takto môžeme vyjadriť, že podmet, napr. *Martin*, má vlastnosť, napr. *maSyna*, s príslušnou hodnotou, napr. *Peter*. Zápis tohto tvrdenia by vyzeral v RDF takto:

```
<Osoba>
  <meno>Martin</meno>
  <maSyna>
```

```
      <Osoba>
        <meno>Peter</meno>
      </Osoba>
    </maSyna>
  </Osoba>
```

Vyjadrovanie v RDF sa veľmi často modeluje orientovaným grafom (obr. 1). Podmet a predmet sú uzly grafu, spojené hranou – prísudkom – smerujúcou z podmetu do predmetu [4]. Vzhľadom na to, že predmet jedného tvrdenia môže byť podmetom iného, dá sa z množstva vyhlásení vytvoriť rozsiahly súvislý graf – skutočná pavučina významov.



Obr. 1: RDF Graf

Naviac RDF označuje všetky zdroje, ktoré popisuje, jedinečným identifikátorom – URI. URI môže byť priradené nielen objektom na Webe a ich vlastnostiam, ale identifikované môžu byť aj objekty reálneho sveta.

Posledným zo stavebných kameňov Sémantického Webu sú ontológie. RDF totiž poskytuje iba všeobecný model pre popis zdrojov, neobsahuje žiadnu slovnú zásobu pre vytváranie samotných metadát. Na to slúžia ontológie, ktoré majú za úlohu formálne popisovať termíny, používané v metadátach. Ontológie definujú triedy objektov a ich vlastnosti pre špecifickú doménu. Triedy aj ich vlastnosti môžu mať rôzne hierarchické vzťahy, tak ako to ilustruje nasledujúci jednoduchý príklad.

```
<rdfs:Class rdf:about="Osoba" rdfs:label="Osoba">
  <rdfs:subClassOf rdf:resource="Clovek"/>
</rdfs:Class>
<rdf:Property rdf:about="maSyna">
  <rdfs:domain rdf:resource="Osoba"/>
  <rdfs:range rdf:resource="Osoba"/>
  <rdfs:subPropertyOf rdf:resource="maPribuzneho"/>
</rdf:Property>
```

Okrem termínov a vzťahov medzi nimi ontológie ponúkajú aj odvodzovacie pravidlá, pomocou ktorých sa dájú z faktov uvedených v metadátach odvodiť nové informácie. Tak napríklad, z predchádzajúcich dvoch príkladov dokáže odvodzovací mechanizmus dedukovať, že ak je *Martin* inštanciou triedy *Osoba* a *Peter* je jeho syn, tak sú obaja z triedy *Clovek* a zároveň sú príbuzní.

V súčasnosti existuje niekoľko jazykov na zápis ontológií. Nadstavbou RDF pre popis ontológií je jednoduché RDFS (RDF Schema), z ktorého vyšiel OIL (Ontology Interchange Language). Ten sa spojil s DAML a na čas bol DAML-OIL najpoužívanejším jazykom z tejto kategórie. Momentálne sa veľké nádeje vkladajú do práve dokončeného OWL (Web Ontology Language), ktorý sa má pod záštitou W3C stať jednotným štandardom pre zápis ontológií na webe.

2.2 Použitie

Od použitia technológií Sémantického Webu sa očakáva pokrok v niekoľkých oblastiach, na nasledujúcich riadkoch vymenujeme hlavné tri [7].

Správa znalostí (Knowledge Management):

- Vyhľadávanie a zbieranie informácií: pokiaľ budú dáta popísané strojovo spracovateľnými metadátami, nebudú vyhľadávacie stroje závislé len na kľúčových slovách a ich výsledky budú relevantnejšie. Softwarový agenti tak budú môcť vyhľadávať, zbierať a triediť informácie, ktoré bude užívateľ potrebovať.
- Správa veľkých úložísk dát: počítačom zrozumiteľné dáta môžu počítače automaticky spravovať – triediť, spájať, aktualizovať a podobne.
- Adaptácia informačných zdrojov: očakáva sa zlepšenie techník prispôsobovania informačných zdrojov (napríklad webových stránok) preferenciám užívateľa, pokiaľ dáta a užívateľský profil budú zrozumiteľne popísané pre adaptačný mechanizmus.

Integrácia podnikových aplikácií (Enterprise Application Integration):
K zlepšeniu súčasného stavu integrácie podnikových aplikácií môžu technológie Sémantického Webu prispieť možnosťou rozšíriteľnosti a znovupoužiteľnosti. Ontológie garantujú, že integrácia môže byť rozšírená podľa potrieb podniku. Štandardy, na ktorých je Sémantický Web postavený, zároveň zaručujú, že vytvorené prostriedky môžu byť znovu použité.

eCommerce:
Rovnako dobre môžu ontológie poslúžiť aj v oblasti eCommerce. Komunikujúce strany vzťahu B2B potrebujú nájsť spoločný jazyk – dostatočne flexibilný, aby postihoval neustále zmeny potrieb obchodu, no zároveň postavený na stabilných a otvorených štandardoch.

Vymenované oblasti použitia môžu vyzerať ako odvážna vízia, ale je potrebné uvedomiť si, že Sémantický Web je len na počiatku svojho vývoja. Aj napriek tomu však už existuje niekoľko rozšírených štandardov, ktoré sú založené na RDF a princípoch Sémantického Webu. Spomenieme len niektoré z nich.

Dublin Core: Formát metadát, pôvodne určený pre popis webových stránok. Vďaka svojim vlastnostiam ako jednoduchosť, modularita, rozšíriteľnosť a hlavne medzinárodná podpora sa stal formátom elektronického popisu zdrojov v múzeách, knižniciach a vládnych inštitúciách.

RSS (RDF Site Summary, Really Simple Syndication): Formát na publikovanie abstraktov webových stránok, noviniek a tlačových správ, v súčasnosti pomerne rozšírený. Napríklad informačné webové portály v tomto RDF formáte zverejňujú popisy svojich článkov.

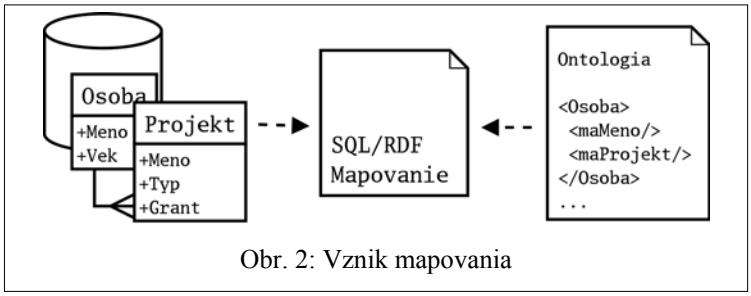
Composite Capability/Preference Profiles (CC/PP): Formát založený na RDF, ktorý popisuje softvérové a hardvérové vlastnosti webových prehliadačov. Tieto profily umožňujú lepšiu spoluprácu medzi webovým serverom a klientom. Podporované je široké spektrum prístrojov od mobilných telefónov po webové televízie.

FOAF (Friend of a Friend): Iniciatíva založená na OWL ontológiách, ktorá si rýchlo získava obľubu. Jedná sa o projekt vytvorenia Webu počítačom zrozumiteľných osobných stránok o ľudoch, vzťahoch medzi nimi a aktivitách, ktorým sa venujú.

3. Generovanie obsahu Sémantického Webu

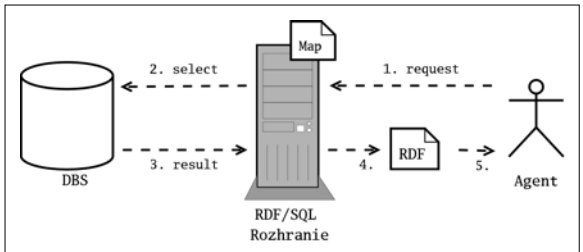
Jedným zo základných predpokladov pre rozšírenie koncepcie Sémantického Webu je vytvorenie kritického množstva metadát, ktoré spustí reťazovú reakciu tvorby ďalších metadát a aplikácií, ktoré s nimi budú pracovať.

Sú dva základné spôsoby tvorby metadát Sémantického Webu. Prvým je anotácia existujúcich alebo práve vznikajúcich webových stránok. Táto anotácia môže byť ručná, pričom sa používajú nástroje na tvorbu RDF, nie nepodobné HTML editorom, avšak anotované webové zdroje musia byť statické. Poloautomatický prístup umožňuje popisovať aj rozsiahlejšie dynamické prezentácie, pričom sa ručne vytvorí šablóna pre sadu stránok s rovnakým výzorom (napr. pre knižničný portál) a stroj (*wrapper*) potom automaticky vyberie z týchto stránok informácie a vytvorí z nich metadáta.



Obr. 2: Vznik mapovania

Množstvo prezentácií na Webe je však vytvárané dynamicky z dát, ktoré sú obvykle uložené v relačných databázach. Logicky sa preto ponúka myšlienka vytvárať RDF dokumenty priamo z databázy, tak ako sú generované dynamické HTML stránky. Takéto riešenie je oproti manuálnej a poloautomatickej anotácii HTML dokumentov rýchlejšie, lacnejšie a zaručuje aktuálnosť RDF prezentácie.



Obr. 3: Použitie mapovania

Všeobecný model generovania RDF priamo z databázy predpokladá mapovanie konkrétnej štruktúry databázy do tried konkrétnej ontológie, ktorú bude výsledný RDF dokument používať (obr. 2).

Vzniknuté mapovanie potom použije mechanizmus publikovania RDF metadát (webový server a pod.), aby vytvoril z databázy RDF dokument (obr. 3).

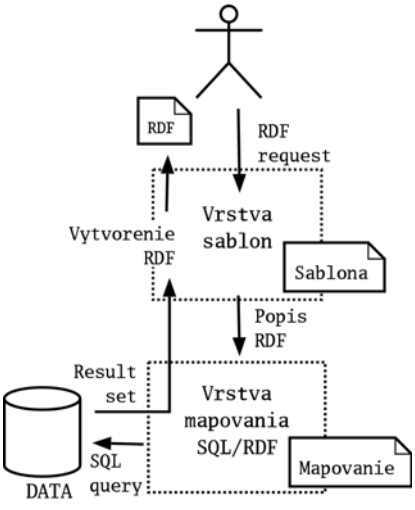
4. Dvojstupňový model mapovania SQL do RDF

V tejto kapitole by sme chceli predstaviť *METAmorphoses* – našu koncepciu mapovania obsahu relačnej databázy do RDF. Tento model bol vytvorený s dôrazom na flexibilnosť SQL/RDF mapovania a na jednoduchosť použitia.

4.1 Úvod do problematiky

Existuje niekoľko projektov [5, 6 a iné], ktoré sa mapovaním SQL do RDF zaoberajú. Tieto systémy majú rôzne zamerania, no žiadny z nich príliš neuľahčuje vytváranie RDF prezentácií bežným programátorom. Ideál, že tvorba RDF by mala byť rovnako jednoduchá ako tvorba HTML [4], je stále veľmi vzdialený. Ako odpoveď na túto výzvu sme navrhli model mapovania *METAmorphoses*. Naším cieľom bolo navrhnúť jednoduché programátorské rozhranie, aby programátori mohli vytvárať RDF fragmenty tak jednoducho, ako tvoria HTML fragmenty pomocou JSP, ASP alebo PHP.

Dva základné predpoklady našej práce sú: dáta sú uložené v „klasickej“ databázovej schéme (teda nie sú v databáze uložené vo forme RDF tripletov) a máme k dispozícii ontológiu (alebo viac ontológií), ktorá špecifikuje formát generovaného RDF.



Obr. 4: Architektúra *METAmorphoses*

4.2 Architektúra systému

Aby sme dosiahli flexibilné mapovanie a vysokú použiteľnosť, rozdelili sme logiku systému do dvoch vrstiev (obr. 4). Vrstva mapovania spája RDF triedy s SQL štruktúrami. Vrstva šablón vytvára podľa tohto mapovania inštancie RDF na základe programátorom definovaných šablón. Obidve vrstvy sú podrobnejšie popísané v nasledujúcej podkapitole.

4.3 Mapovanie a šablóny

Jadrom mapovacej vrstvy je jazyk mapujúci SQL do RDF. Jazyk v základe vychádza z D2R [5], no je pozmenený vzhľadom na architektúru a potreby nášho modelu. Jazyk je založený na XML a navrhnutý tak, aby mapoval triedy konkrétnej ontológie do databázovej štruktúry. To okrem iného znamená, že nemusíme vytvoriť mapu pokrývajúcu celú databázu, ale vyberieme len tie jej časti, ktoré budeme používať. Na nasledujúcom úryvku kódu je ukážka tohto jazyka.

```
<Mapping>
 <Class templateName="person" rdfLabel="foaf:Person"
        sql="select * from person">
  <ClassCondition templateName="username" whereString="username ="/>
  <Variable templateName="personId" sqlName="id"/>
  <Property templateName="firstName" rdfLabel="foaf:firstName"
            sqlName="first_name"/>
  <Property templateName="surname" rdfLabel="foaf:surname"
            sqlName="family_name"/>
 </Class>
</Mapping>
```

Každá mapovaná trieda obsahuje SQL príkaz, ktorým sa získajú dáta pre inštanciu. Naviac môže obsahovať zoznam svojich vlastností, podmienok a premenných. Vlastnosti v triede v mapovania zodpovedajú vlastnostiam RDF triedy. Podmienky a premenné sú konštrukcie, ktoré sa používajú v šablónach na riadenie generovania RDF dokumentu. Každý element v mapovaní obsahuje aj meno príslušného RDF tagu (`rdfLabel`). To znamená, že na tvorbu RDF nemusí byť použité žiadne RDF API, čo je významné zvýšenie výkonnosti systému. Atribút `templateName` spája mapovanie so šablónou.

Šablóny píše programátor, ktorý vytvára dynamickú prezentácie Sémantického Webu. Šablóna riadi generovanie inštancií RDF podľa príslušného mapovania, môže volať všetky elementy v definované mapovaní. Šablóna obsahuje tagy `putInstance` a `putProperty` na pridanie RDF kódu do výsledného RDF dokumentu. No pravidlá ako sa tieto tagy používajú sú uložené v mapovaní. Tagy `Condition` a `Variable` len kontrolujú použitie predchádzajúcich dvoch elementy. Napríklad, nasledujúci kód vytvorí v spojení s vyššie uvedeným mapovaním RDF dokument obsahujúci všetky osoby v databáze s ich krstným menom.

```
<putInstance name="person">
  <putProperty name="firstName"/>
</putInstance>
```
Pridaná podmienka v nasledujúcom príklade obmedzí výber osôb na tie, ktoré majú príslušné užívateľské meno.

```
<putInstance name="person">
  <Condition name="username">svihlm1</Condition>
  <putProperty name="firstName"/>
  <putProperty name="surname"/>
</putInstance>
```

Okrem literálu môže byť obsahom podmienky aj premenná a tak môžeme vytvárať zložitejšie konštrukcie. Výsledkom použitia predchádzajúcej šablóny bude nasledujúce RDF:

```
<foaf:Person>
  <foaf:firstName>Martin</foaf:firstName>
  <foaf:surname>Svihla</foaf:surname>
</foaf:Person>
```

Vzhľadom na to, že šablóny sú tiež zapisované v XML, programátor môže jednoducho používať elementy šablón akoby to boli fragmenty RDF. To napríklad znamená, že šablóna môže byť prostredníctvom užívateľských JSP tagov (*JSP custom tags*) súčasťou JSP stránky. Tak môžu byť tagy šablóny voľne kombinované s JSP tagmi za účelom vytvárania dynamickej RDF prezentácie.

4.4 Generovanie RDF

Mapovanie aj spracovanie šablón obstaráva *METAmorphoses procesor*, napísaný v jazyku Java. Procesor je tiež rozdelený do dvoch častí – spracovanie mapovania a spracovanie šablóny.

Keď príde požiadavka na RDF, procesor nájde príslušnú šablónu, načíta zodpovedajúce mapovanie, pripojí sa do databáze a vytvorí zoznam používaných tried. V ďalšom kroku prechádza postupne šablónu, vyhodnocuje podmienky a premenné a vytvára z nich SQL príkazy. Výsledky týchto príkazov sú premieňané na RDF fragmenty. V tejto fáze sa RDF elementom priraďuje URI. Keď je šablóna spracovaná, výsledné RDF sa pošle späť ako odpoveď.

4.5 Výsledky a poznámky

V tejto kapitole sme len stručne načrtli vlastnosti konceptu *METAmorphoses*, modelu mapovania SQL do RDF. Hlavné výhody navrhovaného riešenia sú flexibilita mapovania databázy a jednoduché programátorské rozhranie. Flexibilita znamená, že mapovací jazyk je schopný zachytiť akúkoľvek ontológiu a že jazyk šablón zodpovedá svojou štruktúrou všeobecnému modelu RDF. RDF model vychádza z tripletov, kde objekt má vlastnosť, ktorej hodnota je literál alebo iný objekt. Túto požiadavku spĺňajú šablóny, v ktorých môže byť inštancia hodnotou vlastnosti inej inštancie.

Jednoduché programátorské rozhranie je realizované šablónami. Tie sú založené na XML a tak môžu byť napríklad časťou JSP stránky.

Ďalšou výhodou je, že pokiaľ je vytvorené správne mapovanie podľa danej ontológie, je zaručovaná platnosť výsledného RDF. To znamená, že programátor, ktorý píše šablóny nemusí o ontológiách nič vedieť.

Samozrejme, *METAmorphoses* majú aj slabé miesta. Vzhľadom na dvojvrstvovú štruktúru sa môže stať, že po zmene ontológie nebude nutné zmeniť len mapovanie, ale aj šablóny postavené na tomto mapovaní. Ďalšou slabinou je, že počas návrhu prvej verzie mapovacieho jazyka sme nebrali do úvahy optimálnosť SQL príkazov.

Testy, ktoré sme doteraz so systémom urobili, potvrdili funkčnosť koncepcie. V blízkej budúcnosti by sme chceli *METAmorphoses* testovať v reálnom nasadení, aby sme získanými výsledkami mohli zlepšiť návrh modelu aj implementáciu procesora.

5. Záver

Sémantický Web je v súčasnosti veľmi diskutovaná a prudko sa rozvíjajúca technológia. Štandardy, na ktorých je postavený, sú takmer hotové. Objavujú sa prvé úspešné dátové formáty a aplikácie, ktoré s nimi pracujú. Jedným z hlavných úloh pri rozšírení tohto konceptu je vytvorenie metadát, ktoré by pokryli súčasné webové stránky metadátami zrozumiteľnými pre počítače.

V tomto článku sme predstavili Sémantický Web a možné spôsoby tvorby jeho obsahu a uviedli sme *METAmorphoses*, naše riešenie mapovania obsahu relačnej databázy do RDF na základe danej ontológie. Náš model pozostáva z dvoch vrstiev, čo zabezpečuje flexibilnosť mapovania a hlavne umožňuje jednoduché použitie pre programátora RDF prezentácie.

Na záver by som si dovolil citovať iniciátora myšlienky Sémantického Webu, Tima Berners-Leeho [9]: „Najzaujímavejšie na sémantickom webe nie je to, čo si vieme predstaviť s ním robiť, ale to, čo si predstaviť nevieme. Rovnako ako sme si pred desiatimi rokmi nevedeli predstaviť možnosti súčasného webu.“

Literatúra:

1. Berners-Lee, T., Hendler, J., Lassila, O. The Semantic web. Scientific American, May 2001
2. Extensible Markup Language (XML). March 2004, http://www.w3.org/XML
3. Resource Description Framework (RDF). March 2004, http://www.w3.org/RDF/
4. Hjelm, J. Creating the Semantic Web with RDF. Wiley Computer Publishing, New York, 2001
5. Bizer, Ch.: D2R MAP - A Database to RDF Mapping Language. Proceedings of the 12th International World Wide Web Conference, WWW 2003, Budapest, Hungary, 2003
6. Stojanovic, N., Stojanovic, L., Volz, R.: A reverse engineering approach for migrating data-intensive web sites to the Semantic Web, IIP-2002, Montreal, 2002
7. Fensel, D. Semantic Web Services: A communication Infrastructure for eWork and eCommerce. Proceedings of ICWE 2003, Oviedo, Spain, 2003, s. 1-7
8. McBride, B. Four Steps Towards the Widespread Adoption of a Semantic Web. International Semantic Web Conference 2002
9. Berners-Lee, T., Miller, E. The Semantic Web lifts off. ERCIM News No. 51, October 2002, s. 9-11