Sémantické vyhľadávanie v doméne pracovných ponúk *

Ján Krausko, Michal Barla, and Anton Andrejko

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií
Slovenská technická univerzita v Bratislave
Ilkovičova 3, 842 16 Bratislava, Slovensko
dzonyk@gmail.com

Abstrakt *Pravdepodobne najčastejšie využívanou službou na webe je vyhľadávanie informácií. Prevláda fulltextové vyhľadávanie na základe výskytu kľúčových slov v dokumentoch. Avšak výsledky sú často neuspokojivé. Možným riešením čoraz komplikovanejšieho vyhľadávania informácií na webe je sémantické vyhľadávanie. Cieľom príspevku je návrh implementácie sémantického vyhľadávacieho nástroja schopného pracovať s ontológiou pracovných ponúk vytvorenou v rámci iného projektu. Bližšie sa zaoberáme dopytovaním ontológie pracovných ponúk pomocou dopytovacieho ontologického jazyka SeRQL v ontologickom úložisku Sesame. V príspevku opisujeme softvérový prototyp sémantického vyhľadávacieho nástroja založeného na rámci Apache Cocoon, predstavíme zaujímavé spojenie tohto rámca a ontologického úložiska Sesame do sémantického vyhľadávacieho nástroja. Zameriame sa tiež na rôzne možnosti reprezentácie výsledkov prostredníctvom Apache Cocoon.*

1 Úvod

Web tak, ako ho poznáme teraz je prepojením dokumentov. Iniciatíva webu so sémantikou sa pripojením metadát k publikovaným dokumentom snaží vytvoriť dátovo prepojený web. Cieľom je dosiahnuť podobu, ktorá bude ľahšie strojovo čitateľná, jednoduchšie spracovateľná a vyhodnocovaná. Vhodným prostriedkom na reprezentáciu metadát, inšpirovaným zo znalostného inžinierstva, sú ontológie.

Jednou z najčastejšie využívaných služieb na webe je pravdepodobne vyhľadávanie informácií. V súčasnosti prevláda tzv. fulltextové vyhľadávanie informácií na základe výskytu kľúčových slov. Existujú mnohé algoritmy, ktoré následne usporadúvajú nájdené dokumenty podľa relevantnosti [4]. Fulltextové vyhľadávače nepoznajú obsah dokumentov, ktoré vyhľadali a ktoré zobrazujú používateľovi, čo často vedie k zlým alebo nepresným výsledkom [6].

Možným riešením čoraz komplikovanejšieho vyhľadávania informácií na webe je sémantické vyhľadávanie. Vyhľadávaču neposkytujeme kľúčové slová, o ktorých si myslíme, že by sa mohli často vyskytovať v hľadanom type dokumentu, ale kritéria, ktoré má spĺňať nájdený obsah. Predpokladá sa teda, že vyhľadávač bude „rozumieť" obsahu, ktorý ponúka.

Existuje niekoľko typov sémantického vyhľadávania v dokumentoch [3], [7]. Základným typom je vyhľadávanie informácií (information retrieval) – identifikácia relevantných dokumentov a ich radenia podľa miery vhodnosti. Vyšším typom je vyhľadávanie, ktoré poskytuje odpovede na jednoduché otázky (simple question answering), napríklad "Kto je prezident Slovenskej republiky?". Zdokonalením by bol vyhľadávač, ktorý poskytuje odpovede na komplexné otázky (complex question answering), napríklad "Aká je súčasná situácia vysokého školstva v Slovenskej republike?". U všetkých typov je možné očakávať zvýšenú efektívnosť vyhľadávania. Súčasne platí, že u všetkých typov vyhľadávania budú používané rôzne techniky usudzovania a odvodzovania.

Existuje viacero projektov, ktoré sa priamo zaoberajú alebo aspoň využívajú vyhľadávanie v prostredí sémantického webu. Príkladom je projekt MKSearch[1], v rámci ktorého je vyvíjaný vyhľadávací nástroj založený na indexovaní metadát vo webových dokumentoch. Nástroj vyhľadáva v naindexovaných metadátach uložených vo forme RDF v ontologickom úložisku Sesame.

Projekt Bibster[2] je nástroj na asistenciu výskumným skupinám pri manažovaní, zdieľaní a vyhľadávaní bibliografických dát. Systém pracuje v prostredí peer to peer siete. Na uloženie dát používa ontologické úložisku Sesame a dopytovanie vykonáva prostredníctvom jazyka SeRQL.

V príspevku sa venujeme návrhu sémantického vyhľadávacieho nástroja, schopného poskytnúť výsledky odpovedajúce jednoduchým otázkam v dátach opísaných ontológiami. Overenie návrhu sémantického vyhľadávača sme zrealizovali vytvorením prototypu nástroja (Semantic Search Tool – SST), ktorý umožňuje používateľovi vyhľadávanie v pracovných ponukách na základe vlastností týchto ponúk. Pracovali sme s už existujúcou databázou pracovných ponúk a s doméno-

* Táto práca bola čiastočne podporovaná štátnym programom výskumu a vývoja „Budovanie informačnej spoločnosti" na základe zmluvy č. 1025/04.

[1] MKSearch, http://www.mksearch.mkdoc.org
[2] Bibster, http://bibster.semanticweb.org

vou ontológiou vytvorenou v rámci projektu NAZOU[3] [5].

2 Princíp sémantického vyhľadávania

Vyhľadávací nástroj získa od používateľa jeho kritéria na pracovné ponuky, o ktoré by mal záujem, pomocou formulára. Ten je zostavený z viacerých rolovacích menu, v ktorých si používateľ môže zvoliť požadovanú hodnotu určitej vlastnosti ponuky. Napríklad môže vybrať konkrétnu pracovnú pozíciu, ktorej by sa mala týkať ním požadovaná pracovná ponuka. Jednotlivé zoznamy všetkých rolovacích menu sa načítavajú priamo z ontológie ešte pred samotným otvorením okna formuláru.

Používateľ podľa svojho uváženia nastaví hodnoty vybraným položkám. Na základe takto vybratých hodnôt sa vytvorí dopyt do ontologického úložiska. Výsledkom je zoznam ponúk, ktoré spĺňajú tento dopyt. Nevyplnené rolovacie menu sa pri tvorbe dopytu neuplatnia. Zoznam nájdených pracovných ponúk sa zobrazí v tabuľke, kde sú ku každej ponuke uvedené jej základné atribúty (názov ponuky a meno firmy, ktorá ponuku zadala) vrátane odkazu na zobrazenie jej detailu. Používateľ si môže zobraziť detail o konkrétnej ponuke, prípadne spresniť dopyt opätovným vyplnením hodnôt vo formulároch.

Uvedená metóda umožňuje vyhľadávať pracovné ponuky na základe vlastností, ktoré nadobúdajú hodnoty z ohraničených množín. Takéto vlastnosti smerujú väčšinou do enumerovaných tried alebo tried, ktoré sú plne definované svojimi podtriedami. Pri enumerovanej triede sú vymenované všetky možné inštancie danej triedy a nemá zmysel vytvárať iné inštancie. Pri triede definovanej podtriedami nemôže existovať taká inštancia triedy, ktorá zároveň nie je inštanciou niektorej podtriedy danej triedy.

Keďže uvedené vlastnosti sú zadefinované priamo v jazyku OWL[4] (*owl:oneOf* pre enumerovanú triedu a *owl:unionOf* pre triedu definovanú podtriedami), dá sa metóda sémantického vyhľadávania implementovať dostatočne generický tak, aby sa dala použiť pre vyhľadávanie inštancií ľubovoľnej triedy, ktorá má vlastnosti smerujúce na plne definované triedy.

3 Architektúra nástroja

Navrhnutú metódu sme overili vytvorením prototypu webovej aplikácie SemanticSearchTool (SST), ktorá umožňuje sémantické vyhľadávanie nad ontológiou pracovných ponúk.

Aplikácia je integrovaná do prezentačného rámca Cocoon, ktorý je založený na architektúre dátovodov a filtrov [1]. Cocoon obsahuje množstvo použiteľných blokov, ktoré po správanej konfigurácii poskytujú aplikácii bohatú funkcionalitu. Pre uloženie ontológie používame ontologické úložisko Sesame[5]. Sesame štandardne poskytuje RDFSchema úložisko so základným odvodzovaním vzťahov medzi triedami a inštanciami.

Základné prepojenie jednotlivých častí riešenia je zobrazené na obrázku 1. Používateľ vidí vo svojom prehliadači stránku vygenerovanú servletom Cocoon, v ktorom je zasadený nástroj SST. Ten je pomocou Sesame Repository API prepojený na ontologické úložisko, ktoré teoreticky nemusí byť spustené v tom istom servlet kontajneri na tom istom stroji.

Sesame dovoľuje ontológie ukladať do súboru, do pamäte alebo do RDBMS (Relational DataBase Management System). Závisí od požiadaviek používateľa, ktorý spôsob uprednostní. Uchovávaním ontológie v pamäti počítača vo forme ontologického modelu dosiahneme vysokú rýchlosť prehľadávania a odvodzovania znalostí. V tomto prípade sme však obmedzení jej kapacitou, čo môže byť dôvodom použitia súboru, kde sa znižuje rýchlosť práce, alebo môžeme použiť relačnú databázu [2].

Sesame môže súčasne spravovať viac ontologických úložísk, a preto pri nadväzovaní komunikácie (pozri obr. 2, fáza 1) je potrebné identifikovať konkrétne úložisko prostredníctvom jeho identifikátora.

4 Dopytovanie nad ontológiou pracovných ponúk

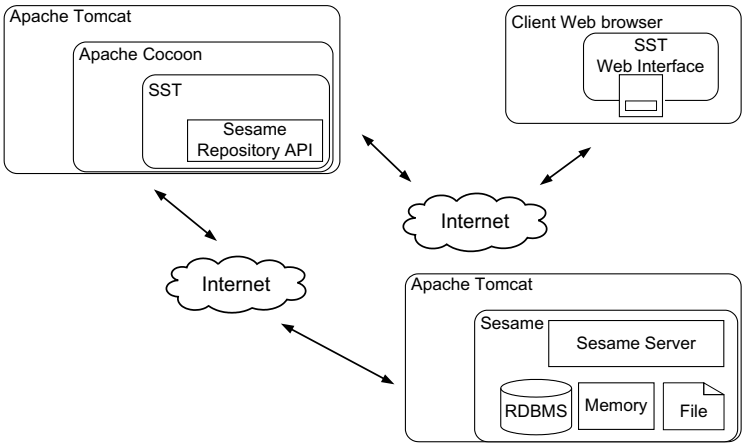
Cieľom nástroja pre sémantické vyhľadávanie je zostaviť dopyt, ktorého výsledkom bude zoznam pracovných ponúk vyhovujúcich kritériám používateľa. Nástroj SST prostredníctvom svojho formuláru (obrazovka aplikácie je na obrázku 2) umožňuje používateľovi špecifikovať tieto zúženia (kritéria) na vyhľadávanie:

- oblasť podnikania firmy, ktorá ponuku uverejnila, kde zoznam možností v rolovacom menu sa získava priamo z ontológie,
- pracovná pozícia – zoznam možností v rolovacom menu sa získava z ontológie,
- región – miesto vykonávania práce; zoznam možností v rolovacom menu sa získa z ontológie,
- druh pracovného pomeru – zoznam možností v rolovacom menu sa získa z ontológie, napr. *self employed* alebo *contract*,
- vhodnosť pre uchádzača bez praxe – k dispozícii sú predvolené možnosti: áno, nie, nezáleží.

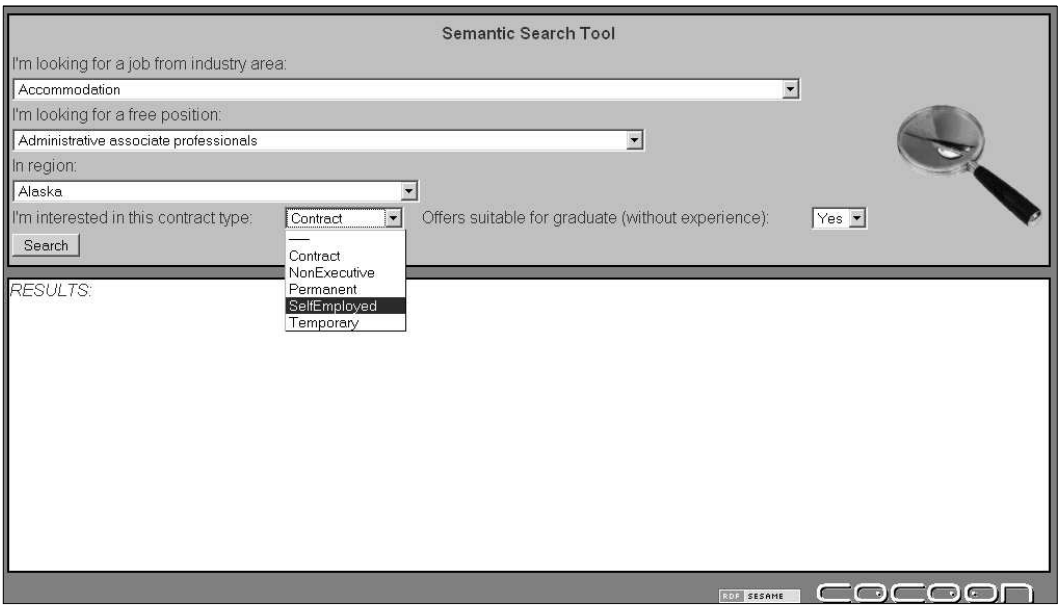
[3] Projekt NAZOU, `http://nazou.fiit.stuba.sk`

[4] OWL – Web Ontology Language, `http://www.w3.org/TR/owl-features`

[5] Sesame, `http://www.openrdf.org`



Obrázok 1. Architektúra komponentov sémantického vyhľadávacieho nástroja.



Obrázok 2. Používateľské rozhranie aplikácie.

Zaujímavým kritériom je *vhodnosť pre uchádzača bez praxe*, keďže takýto atribút ponuky sa priamo v ontológií nenachádza. Ku každej ponuke však existujú predpoklady kladené na uchádzača (*hasPrerequisites*). Tie môžu byť s ponukou spojené vlastnosťou *requires* (zamestnávateľ striktne vyžaduje splnenie predpokladu) alebo *prefers* (uchádzači, ktorí predpoklad spĺňajú sú zvýhodnení). Každý predpoklad sa môže týkať jednej z troch klasifikácií: klasifikácia skúseností, kvalifikácie a osobnostných atribútov.

Z uvedeného spôsobu modelovania vyplýva, že ponuka, ktorá je vhodná pre uchádzača bez praxe nesmie mať žiaden predpoklad prepojený s taxonómiou skúseností pomocou predikátu *requires*.

Dopyt, ktorý nám vráti zoznam ponúk nevhodných pre uchádzača bez praxe bude v jazyku SeRQL vyzerať nasledovne:

```
SELECT Offer
  FROM {Offer} jo:hasPrerequisite {P},
    {P} jo:requires {C},
    {C} rdf:type {c:ExperienceClassification}
```

Ontológia explicitne definuje jednotlivé logické časti entity (v našom prípade pracovnej ponuky) a vzťahy medzi týmito časťami. Hodnoty v každom poli formulára predstavujú možné ohraničenia priestoru ponúk podľa príslušnej logickej časti (región, pozícia a pod.). Podľa toho či používateľ dané pole vyplnil alebo nie sa vytvorením prienikov a zjednotení jednotlivých do-

pytov poskladá zložitejší dopyt zahrňujúci všetky požadované kritéria na hľadanú pracovnú ponuku.

5 Implementácia nástroja

Pre zostavovanie dopytov na ontologické úložisko sme použili dopytovací jazyk SeRQL[6], ktorý vychádza z jazykov RDQL a RQL a čiastočne aj jazyk SPARQL[7] s ktorého plnou podporou sa v budúcnosti počíta po ukončení procesu štandardizácie organizáciou W3C. Okrem jazyka SeRQL rámec Sesame podporuje aj jazyk RDQL a RQL.

Vytvorená klientská aplikácia sa pripája cez Sesame Repository API k Java Servlet aplikácii umiestnenej na vzdialenom webovom serveri Apache Tomcat[8]. Sesame Repository API je jedným z dvoch Sesame Access API rozhraní umožňujúcich vysokoúrovňový prístup do úložiska s funkciami, ako dopytovanie či vkladanie RDF súborov. Druhé Graph API rozhranie poskytuje "jemnejšiu" prácu nad úložiskom, ako je vytváranie malých RDF modelov priamo v kóde či rôznu manipuláciu s RDF. Implementácia klientskej aplikácie je založená na rámci Apache Cocoon[9] určenom pre vytváranie webových aplikácií založených na XML od Apache Foundation.

Spolupráca všetkých komponentov zahŕňa 3 navzájom nadväzujúce fázy, ktoré sú zobrazené na obrázku 3. Prvá fáza zabezpečuje nadviazanie spojenia so serverom a prvotnú komunikáciu servera s klientskou aplikáciou a servera s ontologickým úložiskom. Nástroj SST sa pokúsi pripojiť na vzdialené ontologické úložisko Sesame prostredníctvom prihlasovacieho mena a hesla. Po úspešnej autentifikácii je nutné identifikovať konkrétne úložisko pomocou *repositoryID*, s ktorým bude nástroj pracovať.

V druhej fáze je zostavený dopyt v jazyku SeRQL. Získané dáta odpovedajúce výsledku dopytu Sesame posiela vo forme tabuľky do ďalšej fázy. Dopyt môže byť zadaný priamo v kóde ako znakový reťazec dodržujúci syntax SeRQL alebo môže byť vygenerovaný na základe volieb používateľa zadaných prostredníctvom ponúknutých formulárov.

O tretiu fázu sa takmer výhradne stará rámec Cocoon, ktorý na základe šablón a výsledkov získaných z ontológie generuje, transformuje a nakoniec serializuje dáta do požadovaného formátu. V Apache Cocoon používame implementovaný JX generátor, ktorý má na vstupe šablónu, do ktorej na určené miesta doplní aktuálne hodnoty premenných. Šablóna spolu s CSS[10] určuje celkové rozloženie elementov na stránke. Cocoon vo fáze transformácie umožňuje tiež použitie XSL[11] štýlov. Rovnako je tiež možné využiť vlastný transformátor, ktorý relatívne ľahko naprogramujeme v jazyku Java alebo niektorý zo vstavaných transformátorov, napríklad i18n umožňujúci internacionalizáciu vytvorenej aplikácie. Súbor sa nakoniec serializuje do (X)HTML. Cocoon však ponúka oveľa viac možných výstupných súborov. Môžeme napríklad využiť WAP/WML Serializer na vytvorenie stránky pre mobilné zariadenia alebo pomocou PDF Serializer vygenerovať PDF dokument vhodný pre tlač.

Priebeh klientskej požiadavky je zobrazený na obrázku 4. HTTP požiadavku prichádzajúcu z webového prehliadača klientskeho počítača zachytáva aplikačný server (Apache Tomcat). Cocoon sa riadi konfiguračným súborom SiteMap (sitemap.xmap) vo formáte XML a dátovodmi (*pipelines*), ktoré sú v ňom nakonfigurované. Dátovody pracujú obdobne ako to poznáme pri dátovodoch v systéme UNIX (výstup jedného komponentu sa predáva na vstup druhého komponentu). V konfiguračnom súbore je tak isto namapovaná virtuálna adresárová štruktúra (používaná v HTTP požiadavkách) na skutočnú použitú na strane servera. Klientska požiadavka sa na základe regulárneho výrazu namapuje na príslušný dátovod. Riadenie a logiku nástroja zabezpečuje flowscript (vychádzajúci zo syntaxe jazyka Javascript), ktorého funkcie sa volajú z jednotlivých častí dátovodov. Na komunikáciu a prácu s ontologickým úložiskom Sesame Flowscript využíva metódy triedy MySesame, ktorá je vytvorená v jazyku Java a vložená do rámca Cocoon. Výsledky sú spätne odovzdané konfiguračnej mape, ktorá pre vytvorenie HTTP odpovede následne vyberá príslušný dátovod.

Prepojenie Cocoonu a aplikačnej logiky

Flowscript predstavuje jedno možné miesto prepojenia Cocoonu a aplikačnej vrstvy. Jeho funkcionalita je rozdelená do dvoch častí. Funkcia *vstup* zabezpečuje logiku pre úvodný formulár, zostavenie dopytu pomocou triedy *MySesame*, vyhľadávanie a zobrazenie nájdených ponúk. Funkcia *detail* zabezpečuje logiku pre zobrazenie detailu konkrétnej ponuky.

Prvá funkcia je v činnosti od spustenia nástroja SST. Pripojí sa na Sesame, získa všetky údaje potrebné na naplnenie rolovacích menu formulára a prostredníctvom bloku CocoonForms a generátora *JXGenerator* vygeneruje HTML dokument s formulárom.

[6] SeRQL – Sesame RDF Query Language

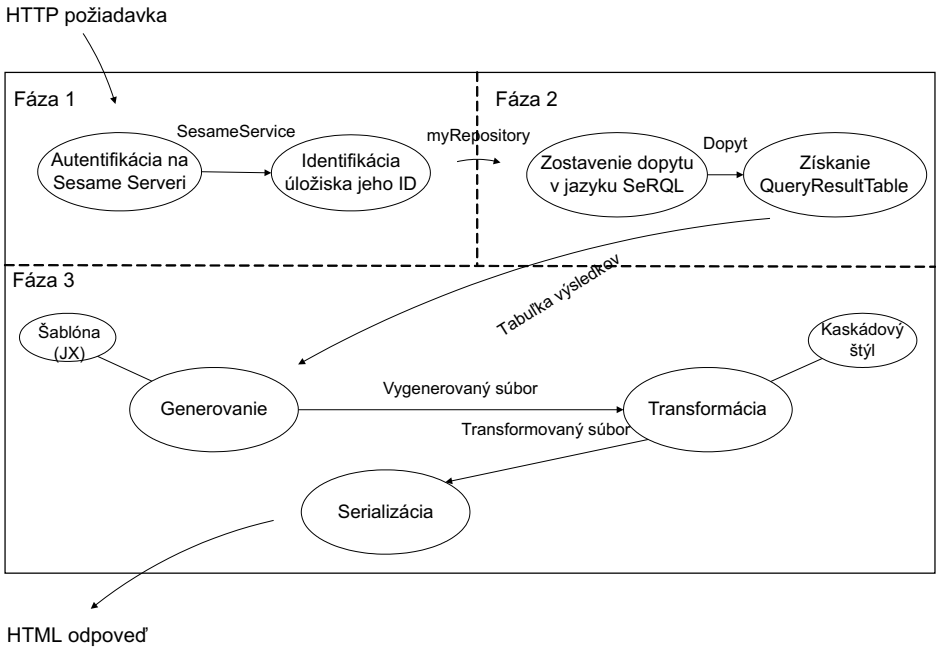
[7] SPARQL – SPARQL Protocol and RDF Query Language,`http://www.w3.org/TR/rdf-sparql-query/`

[8] Apache Tomcat, `http://tomcat.apache.org`

[9] V čase písania príspevku bol rámec Apache Cocoon vo verzii 2.1.8, `http://cocoon.apache.org`

[10] CSS – Cascading Stylesheet, `http://www.w3.org/Style/CSS`

[11] XSL – The Extensible Stylesheet Language Family, `http://www.w3.org/Style/XSL`



Obrázok 3. Znázornenie priebehu spracovávania.

Druhá funkcia dostane na svoj vstup ID konkrétnej ponuky a na jeho základe vyhľadá všetky dostupné informácie spojené s touto ponukou. Výsledný HTML dokument sa vygeneruje použitím *JXTemplate* šablóny, do ktorej sa vložia získané dáta.

Definícia formulára

Jednotlivé prvky formuláru (tzv. *widgety*) vyžadujú definíciu vo vstupnom súbore formou XML štruktúr. Napríklad pre rolovacie menu lokalít pracovných ponúk vyzerá takto:

```
<fd:field id="lokalita">
  <fd:label>In region: </fd:label>
    <fd:datatype base="string"/>
    <fd:selection-list type="flow-jxpath"
      list-path="List"
      value-path="value" label-path="label"/>
</fd:field>
```

Takto definovaný *widget* hovorí, že prvok formuláru *lokalita* bude ponúkať možnosť výberu jednej hodnoty zo zoznamu hodnôt, čo bude reprezentované formou rolovacieho menu (*selection-list*). Hodnoty jeho zoznamu v tomto prípade nie sú dopredu staticky nastavené, ale mu budú odovzdané z flowscriptu prostredníctvom premennej *List*.

Zobrazenie výsledkov

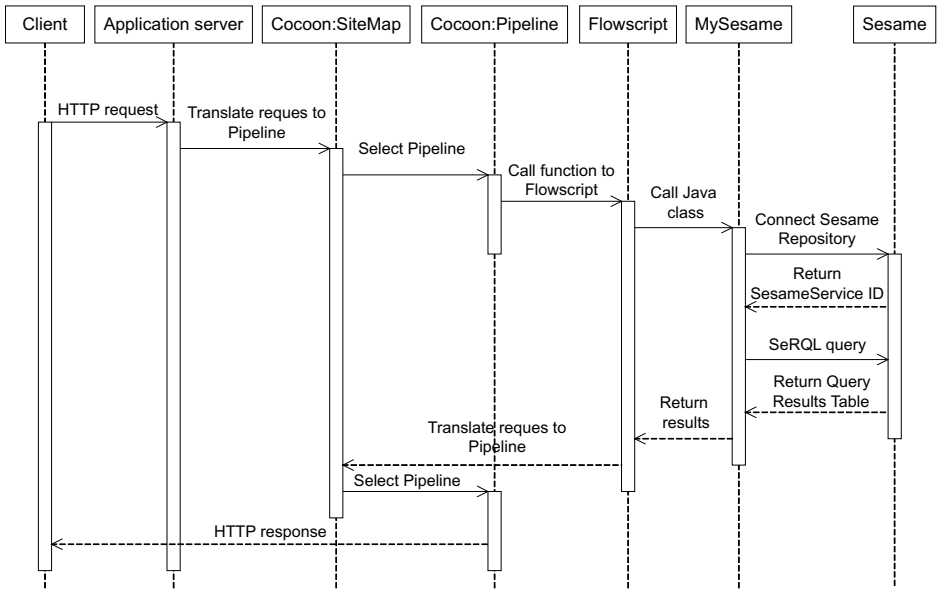
Zobrazenie výsledkov dopytu je možné viacerými spôsobmi – buď sa znova použije rámec CocoonForms a výsledky sa zobrazia vo formulári alebo sa použijú šablóny pre zobrazenie výsledkov.

Rámec Cocoon obsahuje silný šablónový mechanizmus *JXTemplate*, pomocou ktorého sa dá nadefinovať výzor stránok spolu s pripojenými súbormi kaskádových štýlov. Šablóny používa *JXGenerator*, ktorý na určených miestach rozvinie zadefinované makrá a nahradí premenné skutočnými hodnotami.

Pri použití rámca CocoonForms sa dá využiť mapovanie medzi modelom formulára a dátami vo forme XML alebo Java bean. Toto mapovanie zadefinujeme pomocou pravidiel, ktoré priraďujú *widgety* z modelu formulára *JXPath* výrazom, ktoré adresujú polia v XML alebo premenné v Java bean objekte. Následne stačí, aby metóda, ktorá vykonáva dopyt vrátila výsledok vo forme XML alebo Java bean objekt. Takéto riešenie oddelí metódu sémantického dopytu od samotného prezentačného rámca, čo umožňuje jeho ľahkú výmenu. Zároveň je takáto metóda dopytovania flexibilná voči zmenám samotného rámca.

6 Zhodnotenie

Z viacerých možností riešení celkovej koncepcie sémantického vyhľadávacieho nástroja sme sa venovali spojeniu aplikácie postavenej na báze rámca Apache Cocoon a sémantického úložiska Sesame. Myšlienky použité pri návrhu takejto architektúry webovej aplikácie a spôsobu reprezentácie získaných výsledkov sú použiteľné všeobecnejšie.



Obrázok 4. Sekvenčný diagram priebehu klientskej požiadavky.

Dôležitou časťou nášho sémantického vyhľadávacieho nástroja je jeho logika, ktorú tvorí flowscript napísaný v jazyku Javascript, konfiguračný súbor s dátovodmi pre Cocoon vo formáte XML a Java trieda pre komunikáciu s úložiskom Sesame. Používateľské rozhranie webovej aplikácie sa generuje na základe šablón vo formáte XML. Celkový vzhľad aplikácie je postavený na kaskádových štýloch CSS, čo umožňuje ľahkú zmenu celkového výzoru.

Riešenie založené na využití ontológií je oproti relačnej databáze flexibilnejšie, umožňuje niektoré zmeny v štruktúre dát (napr. zmenu či pridanie novej položky v pracovnej ponuke) bez nutnosti zásahu do aplikácie a tiež odvodzovanie znalostí. Dopytovanie v jazyku SeRQL je aktuálne závislé na konkrétnej doménovej ontológii. Nasadenie nástroja do inej oblasti by vyžadovalo zásah do zdrojových kódov. Skúmame možnosti generalizácie nástroja tak, aby sa pomocou neho dalo sémanticky dopytovať aspoň čiastočne nezávisle od použitej ontológie.

Referencie

1. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice. Second Edition, Addison-Wesley, 2003.
2. Chong, I. E., Das, S., Eadon, G., Srinivasan, J.: Supporting Keyword Columns with Ontology-based Referential Constraints in DBMS. In 22nd International Conference on Data Engineering (ICDE'06) (2006) 95
3. Finin, T., et al.: Information Retrieval and the Semantic Web. In Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) – Track 4. IEEE Computer Society (2005)
4. Fürnkranz, J.: Web Mining. In The Data Mining and Knowledge Discovery Handbook. Springer (2005) 899-920
5. Návrat, P., Bieliková, M., Rozinajová, V.: Methods and tools for acquiring and presenting information and knowledge in the web. In International Conference on Computer Systems and Technologies – CompSysTech' 2005, Varna, Bulgaria (2005)
6. Roush, W.: Search Beyond Google. MIT Technology Review (2004) 34-45
7. Sklenák, V.: Sémantický web. In Inforum 2003 (2003)