Pavol Návrat, Valentino Vranić (Eds.)

WIKT 2008

3rd Workshop on Intelligent and Knowledge Oriented Technologies

Nakladateľstvo STU
Bratislava 2009

Pavol Návrat, Valentino Vranić (Eds.)

WIKT 2008

3rd Workshop on Intelligent and Knowledge Oriented Technologies

Zborník príspevkov
Proceedings

Smolenice, 6. – 7. november 2008

Nakladateľstvo STU
Bratislava 2009

Dielňu organizovali/The workshop was organized by:

Ústav informatiky a softvérového inžinierstva FIIT STU v Bratislave
Centrum pre informačné technológie
Ústav informatiky Slovenskej akadémie vied
Nadácia pre rozvoj informatiky

WIKT 2008, http://wikt2008.fiit.stuba.sk/

Ústav informatiky a softvérového inžinierstva
Slovenská technická univerzita v Bratislave
Ilkovičova 3
842 16 Bratislava

Executive editor: Anton Andrejko
Cover design: Peter Kaminský

Vydalo/Published by:
Nakladateľstvo STU
Vazovova 5, Bratislava

ISBN 978-80-227-3027-3

Predhovor

Tvorivá dielňa venovaná inteligentným a znalostne orientovaným technológiám (Workshop on Intelligent and Knowledge Oriented Technologies) sa konala tretí raz. Zdá sa, že neformálne stretávanie výskumníkov zaujímajúcich sa o túto problematiku má nejaký zmysel. Dnes, keď nie je núdza o najrôznejšie vedecké podujatia na najrôznejšie témy, samozrejme vrátane tém, patriacich do tejto problematiky, musí každý takýto projekt viac ako kedykoľvek predtým hľadať dôvody, pre ktoré by sa ho výskumníci zúčastnili.

Naštastie, prvým a najhlavnejším dôvodom je prirodzená potreba výskumníkov porozprávať svoje problémy a výsledky niekomu, kto im porozumie a prípadne zareaguje. Výmena skúseností alebo konfrontácia názorov sú pre pokrok vo výskume veľmi užitočné. Hlavnou ambíciou tvorivej dielne WIKT je umožniť takéto stretávanie sa výskumníkov. Oblasť inteligentných a znalostne orientovaných technológií je dostatočne atraktívna oblasť výskumu, ktorá priťahuje viaceré pracoviská aj jednotlivých výskumníkov, aby v nej uskutočňovali výskum v slovenskom alebo širšie slovensko-českom priestore. Na druhej strane, je pravdepodobne dobré, že existujú aj „skromnejšie“ vedecké podujatia, povedzme na regionálnej úrovni, ktoré poskytujú potrebné zázemie alebo podhubie, infraštruktúru vedeckej komunikácie pre výsledky in statu nascendi, z ktorých sa dúfajme vypracujú – aj vďaka takýmto príležitostiam – silné vedecké výsledky na špičkovej medzinárodnej úrovni.

Hlavné témy tejto pracovnej dielne sú:

- znalostné technológie a ich aplikácie
- modelovanie znalostí a ontológie
- sémantické spracovanie informačných zdrojov (dokumenty, elektronická komunikácia, databázy, znalostné procesy)
- spracovanie informačných zdrojov v slovenskom jazyku
- sémanticky a servisne orientované architektúry
- znalostné bázy a organizačné pamäte
- usudzovanie a odvodzovanie

Vítané boli príspevky v tvare rozšíreného abstraktu v slovenskom, českom alebo anglickom jazyku nasledujúcich typov:

- výskumný príspevok
- work-in-progress
- vizionársky príspevok
- znalostné praktiky
- ponaučenia a skúsenosti
- aplikačný príspevok

WIKT začal v Bratislave, pokračoval v Košiciach a teraz je späť na západe Slovenska. Vynára sa schéma západo-východnej hojdačky. Popri akejsi vnútroslovenskej politickej korektnosti má svoje hlbšie opodstatnenie. Tak v Košiciach, ako aj v Bratislave sa koncentrujú viaceré pracoviská, ktoré robia výskum a dosahujú rešpektovateľné výsledky v oblasti tejto tvorivej dielne. Preto je len prirodzené, ak sa tieto tvorivé dielne budú konať striedavo na rôznych miestach. Súčasne je veľmi dobre, že sa na dielni zúčastňujú aj mimoslovenskí účastníci. Výskumníci z Moravy či Čiech prinášajú svoje cenné skúsenosti, ktoré sú obohatením tejto dielne.

Vzájomná výmena skúseností je prvotným dôvodom, avšak WIKT sa snaží od začiatku o to, aby príspevky spĺňali rozumne náročné požiadavky na kvalitu. Preto nie každý príspevok ponúknutý na WIKT, prejde úspešne výberovým sitom posudzovania. Na túto tvorivú dielňu bolo ponúknutých 25 príspevkov. Každý z nich posudzovali dvaja posudzovatelia v zásade z programového výboru dielne. Výsledkom posudzovania bolo rozhodnutie o prijatí 23 príspevkov. I keď podiel prijatých príspevkov je veľmi vysoký, predsa len si treba všimnúť, že všetky prijaté príspevky prešli recenzným konaním a prijatie nebolo v žiadnom prípade automatické.

Za veľmi dôležitú črtu tejto dielne považujeme, že víta účasť študentov všetkých troch stupňov, ktorých projekty sa týkajú jej tém.

Ďakujeme všetkým členom programového výboru za ochotné posúdenie množstva príspevkov. Bez ochotného programového výboru sa seriózne vedecké podujatie nedá zorganizovať.

Ďakujeme všetkým členom organizačného výboru. Podarilo sa im vytvoriť veľmi príjemné a inšpiratívne prostredie pre túto dielňu. Smolenický zámok je na dielne ako stvorený – v kombinácii so šikovnými a oduševnenými organizátormi.

január 2009	Pavol Návrat predseda programového výboru	Valentino Vranić predseda organizačného výboru

Preface

The Workshop on Intelligent and Knowledge Oriented Technologies took place for the third time. It seems that informal meeting of researchers interested in this problem area makes some sense. Today when there is a variety of different scientific undertakings on different topics – including the topics that belong to this problem area, of course – each such project must more than ever look for the reasons for researchers to participate in it.

Fortunately, the first and main reason is the natural need of researchers to tell about their problems and results to someone who would understand them and possibly react. Sharing experiences or opinion confrontation are very useful for the advancement in research. The main ambition of WIKT is to enable such meeting of researchers. The area of intelligent and knowledge oriented technologies is sufficiently appealing area of research that attracts several institutions and individual researchers to realize their research in it in Slovak or wider Slovak-Czech area. On the other side, it is probably good that there are also "more modest" scientific undertakings –say at the regional level – that provide the necessary groundings, mycelium, or infrastructure of scientific communication for the results in statu nascendi from which we hope – thanks to such opportunities as well – strong scientific results at an international level will develop.

Main topics of this workshop are:

- knowledge technologies and their applications
- knowledge and ontology modeling
- semantic processing of information resources (documents, electronic communication, databases, knowledge processes)
- processing of information resource in Slovak language
- semantic and service oriented architectures
- knowledge bases and organizational memories
- reasoning and inference

Papers in the form of extended abstract in Slovak, Czech, or English language of the following types were welcome:

- research paper
- work-in-progress
- visionary paper
- knowledge practices
- lessons and experiences
- application paper

WIKT has started in Bratislava, proceeded in Košice, and it is again at the west of Slovakia. A scheme of west-east seesaw emerges. Along with some intra-Slovak political correctness, there is also a deeper justification. As in Košice, so in Bratislava

several institutions that perform research and achieve respectable results in the area of this workshop are focused. So it is just natural if these workshops are to be organized interchangeably at different places. At the same time, it is very good if the workshop is attended by non-Slovak participants. Researches from Morava or Bohemia bring their valuable experience that enriches this workshop.

Experience interchange is the primary reason, but from the beginning WIKT aims to have the papers fulfill reasonably demanding quality criteria. This is why not each paper offered to WIKT passes successfully through the selection sieve of reviewing. To this workshop 25 papers have been offered. Each one was reviewed by two reviewers mostly from the program committee of the workshop. The reviewing resulted in a decision to accept 23 papers. Even though the acceptance ratio is high, it has to be noted that all accepted papers have passed through the review process and acceptance had by no means been automatic.

We consider as a very important feature of this workshop that it welcomes the participation of students from all three levels whose projects are related to its topics.

We thank all members of the program committee for their willingness to review a number of papers. Without a willing program committee no serious scientific undertaking could take place.

We thank all members of the organizing committee. They managed to create a very pleasant and inspiring environment for this workshop. Smolenice Castle seems as if it had been created for workshops – in combination with clever and dedicated organizers.

January 2009

Pavol Návrat
Program Committee Chair

Valentino Vranić
Organizing Committee Chair

Programový výbor (Program Committee)

Predseda (Chair):

Pavol Návrat, FIIT STU Bratislava

Členovia (Members):

Mária Bieliková, FIIT STU Bratislava
Ladislav Hluchý, SAV Bratislava
Stanislav Krajči, UPJŠ Košice
Michal Láclavík, SAV Bratislava
Marián Mach, TU Košice
Ján Paralič, TU Košice
Viera Rozinajová, FIIT STU Bratislava
Peter Vojtáš, UPJŠ Košice
Valentino Vranić, FIIT STU Bratislava

Organizačný výbor (Organizing Committee)

Predseda (Chair):

Valentino Vranić, FIIT STU Bratislava

Členovia (Members):

Mária Bieliková, FIIT STU Bratislava
František Babič, FEI TU Košice
Daniela Chudá, FIIT STU Bratislava
Viera Rozinajová, FIIT STU Bratislava
Ján Suchal, FIIT STU Bratislava

Table of Contents

Web

Databázy a informačné systémy

Spracovanie prirodzeného jazyka

Nástroje

Web

Integrované prehliadanie webu a webu so sémantikou pomocou prehliadača novej generácie

Michal Tvarožek, Mária Bieliková

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií
Slovenská technická univerzita
Ilkovičova 3, 842 16 Bratislava
{tvarozek,bielik}@fiit.stuba.sk

Abstrakt Súčasné webové prehliadače neumožňujú používateľom jednotným spôsobom pristupovať k heterogénnym informáciám na webe a webe so sémantikou, pričom podpora pokročilých úloh, napr. prieskumného vyhľadávanie je značne obmedzená. V príspevku navrhujeme riešenie v podobe webového prehliadača novej generácie, ktorý používateľom transparentne umožňuje pristupovať k webu, webu so sémantikou a prepojeným dátam pomocou personalizovaného fazetového prehliadania.

1 Potreba integrovaného prehliadania informácií na webe

Súčasný web obsahuje veľa heterogénnych zdrojov informácií v podobe webových stránok, obrázkov, audio a video nahrávok, ktoré sú zvyčajne určené pre priamu "konzumáciu" používateľmi. Ďalší zdroj informácií na webe predstavujú informácie a metadáta opísané pomocou prostriedkov webu so sémantikou, ktoré môžu byť pripojené k obyčajným stránkam, dostupné v podobe tzv. "prepojených dát" (angl. Linked data) pomocou definovaného rozhrania alebo uložené v rôznych ontologických úložiskách dostupných na webe pomocou štandardizovaných dopytovacích jazykov. Tieto informácie sú však primárne určené na spracovanie strojmi a neobsahujú preddefinovaný spôsob ich vizualizácie pre ľudí.

Súvisiacim problémom je praktická nedostupnosť informácií nachádzajúcich sa v hlbokom webe (a teda databázach) koncovým používateľom. Štúdia spoločnosti BrightPlanet [1] z roku 2001 odhadla pokrytie povrchového webu (19 TB) vyhľadávacími strojmi na cca. 16-17%, čo predstavovalo len cca. 0,03% všetkých dostupných informácií po zohľadnení hlbokého webu (7500 TB).

Dôležitý je tiež priebežný posun požiadaviek a očakávaní používateľov od "jednoduchého" zodpovedania dopytov k tzv. prieskumnému vyhľadávaniu (angl. exploratory search), ktoré kladie dôraz na (interaktívne) pochopenie informácií, skúmanie, analýzu a učenie [2]. Práve táto oblasť je ešte značne neprebádaná, keďže existujúce prístupy zďaleka nevyužívajú možnosti súčasných zariadení (napr. čoraz väčšie monitory), personalizácie (zohľadnenie individuálnych preferencií používateľov), či interaktívnej grafickej vizualizácie informácií (napr. grafmi alebo zhlukmi).

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 1–4, ISBN 978-80-227-3027-3.

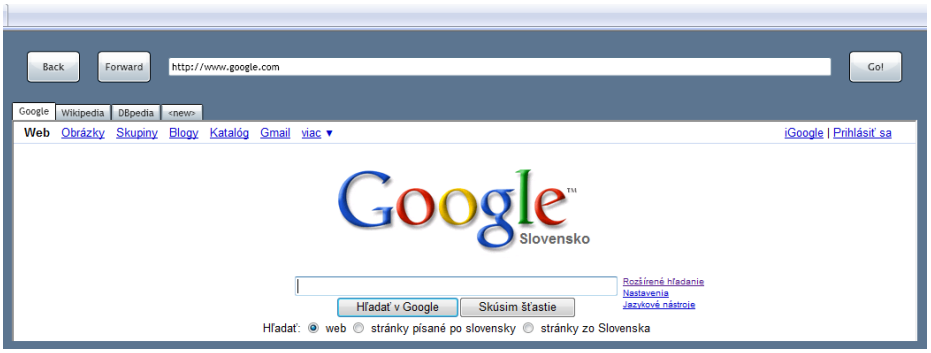
2 Integrovaný webový fazetový sémantický prehliadač

Ako jedno z riešení načrtnutých problémov sme navrhli a v podobe softvérového prototypu realizovali vybrané časti konceptu integrovaného webového prehliadača novej generácie. Náš prehliadač používateľom umožňuje konzistentným spôsobom prehliadať obsah "klasického" webu i webu so sémantikou, pričom tiež poskytuje možnosť integrovaného vyhľadávania a prehliadania informačného priestoru pomocou personalizovaného fazetového prehliadača.

Navrhnutý prehliadač interne pracuje v troch režimoch podľa typu, resp. zdroja spracúvaných informácií. Z pohľadu koncového používateľa je však použitie jednotlivých režimov plne transparentné – prehliadač mení režim práce automaticky na základe spracúvaných dát a dostupnosti príslušných metadát.

2.1 Prehliadač "klasického" webu

Režim *prehliadač "klasického" webu* vychádza zo súčasných webových prehliadačov – používateľom poskytuje typické ovládacie prvky (Obrázok 1), ktoré rozširuje o podporu personalizácie a adaptácie, a o princípy sociálneho webu zohľadnením interaktivity, kolaborácie a vzťahov medzi používateľmi [3].



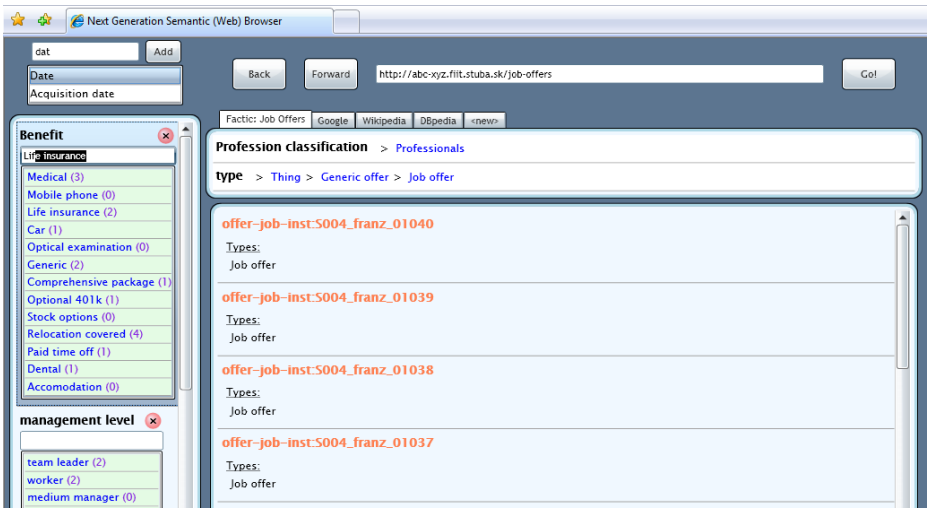
Obrázok 1. Rozhranie navrhnutého prehliadača v režime pre "klasický" web. Súčasťou rozhrania sú typické ovládacie prvky súčasných prehliadačov – "späť", "dopredu", záložky a pole na zadanie URL adresy stránky, resp. kľúčových slov pre vyhľadávanie.

2.2 Prehliadač webu so sémantikou

V režime *prehliadania webu so sémantikou* navrhnutý prehliadač využíva dostupné metadáta a umožňuje používateľom pomocou fazetového prehliadania informačného priestoru vyhľadať relevantné informácie. Prehliadač do tohto režimu prejde automaticky, keď adresa zodpovedá ontologickému úložisku, ktoré obsahuje informácie v ontologickej reprezentácii podľa štandardu RDFS/OWL, pričom naša implementácia využíva dopytovanie pomocou jazyka SPARQL.

V prípade "novoobjaveného" úložiska prebehne najprv inicializácia používateľského rozhrania – analýza metadát v úložisku a vygenerovanie zodpovedajúcich faziet, ktoré používateľ môže následne použiť na navigáciu v informáciách dostupných v úložisku. Samotné rozhranie fazetového prehliadača predstavuje rozšírenie rozhrania z režimu prehliadania "klasického" webu o prvky fazetového prehliadača – zobrazenie faziet, vyhľadávania faziet a ohraničení, zobrazenie aktuálneho dopytu a výsledkov vyhľadávania (Obrázok 2).

Náš prehliadač rozširuje fazetové prehliadače o personalizáciu na základe implicitnej spätnej väzby [4] a nové prístupy ku kolaborácii, viacparadigmovému vyhľadávaniu a vizualizácii informácií pomocou grafov a zhlukov [5].

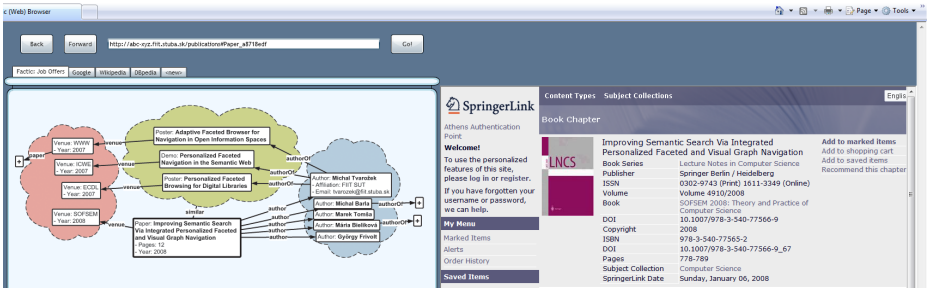


Obrázok 2. Rozhranie prehliadača pre web so sémantikou obsahuje vygenerované fazety (vľavo), zobrazenie aktuálneho dopytu a navigačné nástroje z webového prehliadača (hore), a doménovo nezávislé zobrazenie výsledkov vyhľadávania (v strede).

2.3 Prehliadač prepojených dát

Ak používateľ namiesto URL adresy, resp. adresy ontologického úložiska zadá identifikátor zdroja z prepojených dát (URI), alebo ak sú k bežnej webovej stránke pripojené dodatočné metadáta, prehliadač zmení režim na *prehliadanie prepojených dát* (Obrázok 3). V tomto režime prehliadač na pozadí sťahuje informácie o vybraných zdrojoch z ontologických úložísk, resp. adries určených na základe dereferencovania ich URI pomocou prístupov pre prepojené dáta.

Pohľad na vlastnosti (atribúty) zvoleného zdroja je ekvivalentný zobrazeniu detailov o zvolenom výsledku vyhľadávania v režime prehliadania webu so sémantikou. Zobrazenie vlastností riešime buď v podobe textovej tabuľky, alebo v podobe grafu zdrojov a ich vlastností s podporou horizontálnej inkrementálnej navigácie v informačnom priestore [5] (Obrázok 3).



Obrázok 3. Rozhranie navrhnutého prehliadača v režime pre prepojené dáta, resp. detaily inštancií informačného priestoru. Vlastnosti zvolenej inštancie sú zobrazené v podobe grafu (vľavo), resp. doménovo špecifickej vizualizácie napr. zobrazenia súvisiacich informácií o zvolenej publikácii z "klasického" webu (vpravo).

3 Zhrnutie

Navrhnutá koncepcia integrovaného webového prehliadača novej generácie predstavuje posun v prehliadaní webového obsahu, pretože umožňuje používateľom plne transparentne prehliadať obsah súčasného webu, webu so sémantikou, resp. databáz hlbokého webu, a prepojených dát cez jednotné používateľské rozhranie.

Návrh koncepcie prehliadača vychádza z adaptívneho fazetového prehliadača ontológií vyvinutého v rámci projektu NAZOU (`nazou.fiit.stuba.sk`). Vhodnosť a použiteľnosť navrhnutého riešenia overujeme realizáciou prototypu prehliadača pomocou klientskych webových technológií a vyhodnotením jeho pilotného použitia vo viacerých aplikačných doménach.

Poďakovanie. Táto práca bola čiastočne podporená Kultúrnou a edukačnou grantovou agentúrou MŠ SR, grant No. KEGA 3/5187/07 a projektom VEGA grant No. VG1/0508/09.

Referencie

1. Bergman, M.K.: The deep web: Surfacing hidden value (2001)
2. Marchionini, G.: Exploratory search: from finding to understanding. Comm. of the ACM **49**(4) (2006) 41–46
3. Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. LNCS 4321. Springer, Berlin (2007)
4. Tvarožek, M., Bieliková, M.: Personalized faceted navigation for multimedia collections. In: SMAP '07: Proc. of the 2nd Int. Workshop on Semantic Media Adaptation and Personalization, Washington, DC, USA, IEEE CS (2007) 104–109
5. Tvarožek, M., Bieliková, M.: Collaborative multi-paradigm exploratory search. In: WebScience '08: Proceedings of the hypertext 2008 workshop on Collaboration and collective intelligence, New York, NY, USA, ACM (2008) 29–33

Využitie sociálnych sietí pri inicializácii modelu používateľa pre personalizované webové systémy

Michal Barla, Mária Bieliková

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informacných technológií
Slovenská technická univerzita v Bratislave
Ilkovičova 3, 842 16 Bratislava, Slovensko
{barla,bielik}@fiit.stuba.sk

Abstrakt Personalizované systémy trpia tzv. "cold-start" problémom, keď v prípade nového používateľa systém nemá dostatok informácií na základe ktorých by dokázal účinne prispôsobovať prezentovaný obsah a/alebo navigáciu. Jedným zo sľubných zdrojov informácií o nových používateľoch sa zdajú byť sociálne siete. Tie však neposkytujú priame informácie o individuálnych charakteristikách používateľa. V tomto príspevku predstavujeme použitie rôznych typov vzťahov medzi používateľmi pre inicializáciu modelu používateľa: odhad charakteristík.

1 Úvod

Sociálne správanie ľudí je fenomén, ktorý sa okrem reálneho života prejavuje aj v našej práci s informačnými systémami na webe, ak tieto systémy takéto správanie podporujú [1]. Hovorí sa o druhom webe (Web 2.0), resp. o sociálnom webe. Modely komunít vo webových systémoch potláčajú a niekde aj nahrádzajú tradičnú úlohu modelu používateľa. Objavujú sa systémy realizujúce prispôsobovanie na základe komunít, ktoré majú vlastné, v čase meniace sa charakteristiky [2].

Tradičné prispôsobovanie na báze modelu používateľa však má svoje nezastupiteľné miesto pri efektívnej práci s informáciami. Personalizácia na základe charakteristík jednotlivca ľahko prekoná personalizáciu na základe komunít, v prípade, že model používateľa obsahuje dobré odhady charakteristík používateľa. Veríme, že tieto dva prístupy je možné spojiť a dosiahnúť tak vo výsledku ešte lepší efekt. Súčasný web obsahuje nesmierne množstvo dát, ktoré priamo alebo nepriamo prepájajú jednotlivcov do komunít. Priame prepojenia poskytujú rozhrania (API) populárnych sociálnych aplikácii ako je napr. Facebook[1] aLinkedIn[2]. Nepriame prepojenia dokážeme získať napr. analýzou domovských stránok používateľov a ich prepojení na iné sídla [3]. Sociálna sieť dokážeme extrahovať aj z verejne dostupných dátových úložísk (napr. tripartitný graf autorov, publikácií a konferencií poskytovaný v rámci ontológie SwetoDBLP[3] poskytuje

[1] Facebook, http://www.facebook.com/
[2] LinkedIn, http://www.linkedin.com
[3] SwetoDBLP ontology: http://lsdis.cs.uga.edu/projects/semdis/swetodblp/

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 5–8, ISBN 978-80-227-3027-3.

informáciu o vzťahu `spoluautorstvo`, ale spája aj ľudí, ktorí sa zúčastnili rovnakej konferencie). Dôležité je, že objavené prepojenia sú typované (čiže vieme zadefinovať ich význam) a môžu mať ďalšie atribúty (napr. vzťah `kolega` môže mať atribúty, ktoré upresnia časový interval vzťahu, prípadne inštitúciu, v ktorej obidvaja kolegovia spolu pôsobili). V tomto príspevku prezentujeme koncepciu, ktorá kombinuje komunitný a individuálny prístup k modelovaniu používateľa a využíva komunity na inicializáciu modelu používateľa a prekonanie známeho "cold-start" problému adaptívnych systémov.

2 Modelovanie používateľa s využitím sociálnych vzťahov

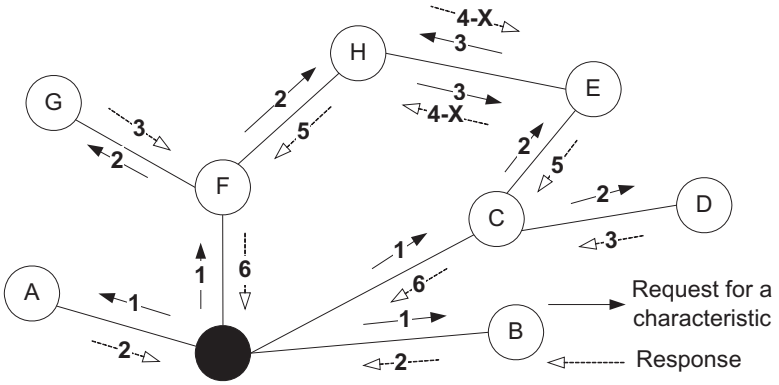
Väčšina prístupov k prispôsobovaniu založených na komunitách uvažuje iba s jedným typom vzťahu `patrí-do` medzi jedincom a príslušnou komunitou. Nami ponúkaný prístup inicializácie modelu používateľa využíva rôzne vzťahy medzi jednotlivcami danej komunity, teda skutočnú sociálnu sieť. Navyše, väčšina súčasných systémov využíva sociálne aspekty správania sa v systéme prezentujúcom informácie iba vo forme anotácie obsahu, aby používateľ videl "navigačné stopy" komunity [4]. Model používateľa sa tým, na rozdiel od nášho prístupu, priamo nemení.

Hlavná myšlienka nášho prístupu inicializácie modelu používateľa je zobrazená na obr. 1. Šírime požiadavku na hodnotu charakteristiky používateľa do používateľovej sociálnej siete tvorenej určitým vzťahom a kombinujeme odpovede do prvotného odhadu charakteristiky, ktorý sa ukladá do modelu. Takáto požiadavka sa vyšle pre každý známy typ vzťahu a pre každú modelovanú charakteristku (napr. preferencie jednotlivých doménových konceptov). Počas spracovania každý uzol siete čaká na odpoveď od všetkých uzlov, ktoré s požiadavkou oslovil (uzol C z obrázka 1 prešíril v kroku 2 požiadavku na uzly D a E, prijal odpoveď od uzla D v kroku 3, ale čaká aj na odpoveď uzla E, ktorá prichádza v kroku 5). Aby sa zabránilo slučkám, každý uzol môže participovať v spracovaní jednej požiadavky iba raz. To je dôvod, prečo si uzly E a H z obrázka navzájom požiadavku odmietnu (v obrázku zobrazené ako X)

Pri prijatí odpovede má uzol nasledujúce možnosti:

- ak sa prijatá hodnota charakteristiky zhoduje s jeho vlastnou, preposiela túto hodnotu s vyššou dôverou;
- ak sa prijatá hodnota vyhodnotí ako podobná k jeho vlastnej, preposiela sa hodnota, ktorá reprezentuje spoločné vlastnosti prijatej a vlastnej hodnoty (typicky spoločný nad-koncept, napr. v doméne publikácií by sa záujem o témy z *Int. Semantic Web Conference* a *European Semantic Web Conference* odvodil záujem o témy nad-konceptu: *Semantic Web Conference*);
- ak je prijatá hodnota príliš vzdialená vlastnej hodnote, preposiela iba svoju hodnotu, s nižšou dôverou.

Porovnávanie charakteristík využíva spôsob opisu charakteristík pomocou metadat z príslušných doménových ontológií, ako bolo ukázané v [5]. To nám umožňuje zapojiť do procesu rôzne prístupy porovnávania ontologických konceptov a inštancií, ako napr. [6].



Obrázok 1. Inicializácia modelu nového používateľa (čiernym) charakteristikami skombinovanými od ostatných používateľov (bielym) spojených určitým typom vzťahu. Niektoré uzly (A,B) odpovedajú priamo, niektoré (F,C,E) požiadavku preposielajú ďalej a zdroju odpovedajú až potom, čo dostanú a spracujú odpovede od ďalších uzlov v ich sociálnej sieti. Každý uzol odpovedá na požiadavku iba raz (uzly H a E si z tohto dôvodu navzájom vymenia odmietavú odpoveď).

3 Trénovanie atribútov modelu šírenia

Kombinácia charakteristík viacerých používateľov prepojených soc. vzťahmi je ovplyvnená niekoľkými parametrami:*(i)šíriteľnosťou* charakteristiky cez určité typy vzťahov (napr. vzťah `spoluator` je vhodný pre šírenie charakteristiky vyjadrujúcej záujem o témy súvisiace s konferenciami o webe so sémantikou, zatiaľ čo vzťah `spolužiak` by nám v tomto prípade nedal žiadne zmysluplné informácie), *(ii)hĺbkou šírenia* (do akej hĺbky sa od nového používateľa bude šíriť požiadavka), *(iii)tranzitivitou vzťahov* a *(iv)ohodnotením prispievajúceho uzla* v rámci siete, získané zo štandardných metód analýzy soc. sietí (zistíme dôležitosť, populárnosť a pod. jednotlivých uzlov, ktoré majú vplyv na jeho váhu pri šírení).

Tieto parametre môžeme určiť metódou pokus – omyl, môžeme využiť znalosti doménových expertov, fokusných skupín a pod. Väčšinu parametrov sa však dokážeme naučiť z dát, ktoré už máme k dispozícii – z modelov tých používateľov, ktorí systém už dlhšie používajú (a už prekonali "cold-start" problém). Spôsobov ako získavať model používateľa je mnoho, jeden príklad založený na analyzovaní správania používateľov sme ukázali v [7].

Váha vzťah – charakteristika. Pre vyššie opísaný model inicializácie modelu používateľa sme navrhli spôsob trénovania váh, ktoré vyjadrujú nakoľko je charakteristika šíriteľná cez určité typy vzťahov. Vychádzame z predpokladu, že ak veľa používateľov spojených určitým typom vzťahu zdieľa hodnotu charakteristiky, môžeme tento vzťah využiť pre odhad tejto charakteristiky pri novom používateľovi. Trénovanie je teda založené na koncepte podobnosti vypočítanej porovnávaním metadát jednotlivých charakteristík používateľov [6].

Podstatou je výpočet (čiastkovej) podobnosti charakteristiky istého typu medzi všetkými dvojicami spojenými vzťahom určitého typu. Výsledná váha je potom získaná ako celková podobnosť (suma čiastkových podobností) podelená počtom dvojíc. Ak sa tento postup zopakuje pre všetky známe typy charakteristík a známe typy vzťahov, dostaneme výslednú maticu váh.

Tranzitivita a hĺbka šírenia. Výpočet hĺbky šírenia pre dvojicu (typ charakteristiky, typ vzťahu) je podobný ako v prípade vyššie opísaného výpočtu váh. Namiesto dvojíc sa postupne vyberajú $n-tice$ (trojice, štvorice atď.) a skúma sa celková podobnosť ich charakteristík. Cieľom je nájsť také maximálne n, pri ktorom sa podobnosť udrží nad stanovenou hranicou. Ak je n väčšie ako dva, vzťah môže byť pre daný typ charakteristiky považovaný za tranzitívny.

4 Záver

Opísali sme nový prístup k riešeniu *cold-start* problému pomocou vzťahov medzi jednotlivými používateľmi systému. Model nového používateľa môže byť inicializovaný s použitím charakteristík ostatných používateľov, znalostí o vzťahoch a ich význame. Zdieľanie *individuálnych* modelov *reálnych* používateľov prístup odlišuje od iných, ktoré používajú buď staticky definované alebo dynamicky vyvíjajúce sa profily komunít. Náš prístup umožňuje použitie štandardných techník pre ďalšiu prácu s modelom používateľa, ktorý, po inicializácii, obsahuje iba hrubý odhad charakteristík a počas používateľovej práce so systémom je ďalej spresňovaný aby lepšie odrážal konkrétneho jednotlivca s jeho potrebami.

Poďakovanie. *Táto práca bola čiastočne podporená agentúrou KEGA, grant 3/5187/07 a Vedeckou grantovou agentúrou VEGA, grant VG1/0508/09.*

Referencie

1. Brusilovsky, P.: Social Information Access: The Other Side of the Social Web. In Geffert, V., et al., eds.: SOFSEM 2008. LNCS 4910, Springer (2008) 5–22
2. Atzenbeck, C., Tzagarakis, M.: Criteria for Social Applications. In Vassileva, J., et al., eds.: Socium: Adaptation and Personalisation in Social Systems: Groups, Teams, Communities. Workshop held at UM 2007. (2007) 45–49
3. Bekkerman, R., McCallum, A.: Disambiguating Web appearances of people in a social network. In Ellis, A., et al., eds.: WWW 2005, ACM (May 2005) 463–470
4. Farzan, R., Brusilovsky, P.: Community-based Conference Navigator. In Vassileva, J., et al., eds.: Socium: Adaptation and Personalisation in Social Systems: Groups, Teams, Communities. Workshop held at UM 2007. (2007) 30–39
5. Andrejko, A., Barla, M., Bieliková, M.: Ontology-based User Modeling for Web-based Information Systems. In: Advances in Information Systems Development. Springer (2007) 457–468
6. Andrejko, A., Bieliková, M.: Investigating Similarity of Ontology Instances and its Causes. In Kurkova, V., et al., eds.: ICANN 2008. LNCS 5164, Springer (2008) 1–10
7. Barla, M., Bieliková, M.: Estimation of User Characteristics using Rule-based Analysis of User Logs. In: Data Mining for User Modeling, Proc. of Workshop held at UM2007. (2007) 5–14

Personalized Recommendation Scheme Based on Content and Knowledge Hierarchy

Zdeněk Velart[1], Petr Šaloun[2]

[1] DCS, FEECS, VSB Technical University Ostrava
zdenek.velart.fei@vsb.cz
[2] DIC, Faculty of Science, University of Ostrava
petr.saloun@osu.cz

Abstract. Navigating students to desired information in the information space is task of many web-based systems. Our solution of navigation and recommendation is based on interconnection of existing concepts space which represents structure of the domain with a space of existing learning objects. For every student we recommend suitable learning objects according the students model and rating of the learning object. Traditional approach which is using prerequisities and gains is replaced with metrics which are calculated from positions of associated concepts in the concepts space. The currently developed web-based system where we test an automatized course preparation based on our proposed approach is discussed.

1 Introduction

The student should be directly navigated to desired information, which is seeking or wants to learn. We base our navigation scheme on existing concepts space of the domain, where relationships between knowledge is captured. Interconnection of the concepts space with a space of existing learning objects is accomplished by the use of the keywords of the domain.

The basic idea was presented in [3, 4]. Based on feedbacks we revised and extended our ideas to provide automation of processing of learning objects and metrics which summarizes learning object rating (complexity) based on content of the learning object and positions of associated concepts in the concepts space and their dependent concepts and their positions. Our problem domains are the domain of Programming in C++ and Electronical publishing. The courses we decided to reuse (result from the run were presented in [2]) were sequential with hyperlinks from previous to next chapter and integrated table of contents. The division is no longer desired and the sequential navigation of the course was stripped down leaving thus a space of separate learning objects.

2 Concepts space based recommendation

Our concepts space based recommendation uses an algorithm which recommends learning objects, which the algorithm evaluates as “simple” to learn for particular

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 9–12, ISBN 978-80-227-3027-3.

student in a particular time. Learning objects are evaluated according their added attributes and actual students model. Every learning object (which can be defined and explained in paragraph, page or set of pages) is represented by the learning material of the domain. The structure of the domain is captured in the concepts space, where relationships between concepts are stored.

The algorithm for processing the space of learning objects and recommendation can be divided into to parts – *pre-run computing* where the space of learning objects is processed and every learning object is enhanced with attributes and *recommendation* where actual navigation between the learning objects for every student is evaluated on the fly.

2.1 Pre-run computing

In this phase every learning object from the space of learning objects is enhanced with attributes: *concepts set*, *depth*, *summarized depth*, *average depth*, *median depth*, *average summarized depth* and *median summarized depth*.

1. For every concept the following attributes are computed – *depth* of the concept in the concepts space and *summarized depth* which is computed as sum of depths of concepts which current concept depends. If concept has not yet defined keywords of the domain, then they are also specified.
2. For every learning object:
   - keywords of the domain are searched in the learning text and examples associated with the text, for better processing we are testing use of Wordnet (http://wordnet.princeton.edu/) relational dictionary and we are currently searching for an alternative for czech language;
   - according the found keywords corresponding concepts are added into *concepts set*;
   - the attributes are computed – *depth* is defined as sum of depths of the concepts in the *concepts set*, *summarized depth* is defined as sum of summarized depths of the concepts in the *concepts set*, *average depth* as average of depths of the concepts in the *concepts set* and likewise *median depth*, *average summarized depth* and *median summarized depth*.

Learning objects can be ordered by different metrics such as – size of *concepts set*; *depth* and *summarized depth* of learning object; size of the *concepts set* together with *median depth*. This pre-run computing phase represents extensive computation for every learning object and concept. All the results are stored into persistent storage for later use in the recommendation process.

As optional information which can be used in interconnection of concepts space with space of learning objects similarity of concepts can be used. In the article [1] authors presents methods for similarity evaluation.

2.2 Recommendation

Information which learning objects student already visited is stored in student's personal profile. When the student learns a new learning object all the concepts

form the *concepts set* are added to his *achieved knowledge set*, which is used for recommendation. For every visited learning object there is also a timestamp stored. It can be later used for purposes of path reconstruction.

Presentation of course menu is based on the preferences and concepts stored in students *achieved knowledge* set. The navigation and menu updating is described by Figure 1 – used numbers corresponds with the description in list:

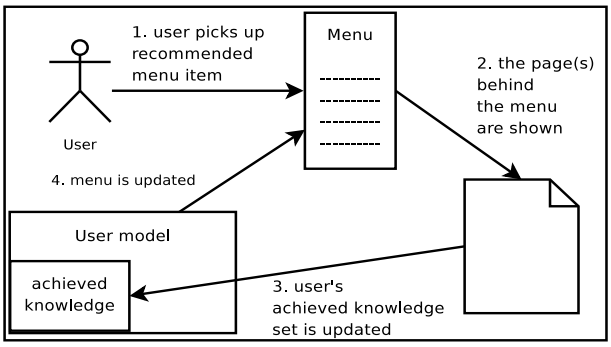


Fig. 1. Course recommendation.

1. Student chooses a learning object from recommended learning objects in the menu, which is presented to him.
2. According the chosen menu item the student is presented with learning material of the particular learning object.
3. When student finishes the learning material(s) from shown learning object, the system can test the students knowledge – this step is optional. The system then updates student's *achieved knowledge* set with the concepts from the *concepts set* of the learning object.
4. Student's menu is updated according his actual *achieved knowledge* set, *concepts set* of unvisited learning objects and recalculated attributes. The actual construction of the menu for the student is following: For every not visited learning object a complement from *concepts set* and *achieved knowledge set* is created. From the result all attributes of the learning object are recalculated in the same way as was described previously. Learning objects for the menu are ordered according the chosen metrics. The number of shown learning objects in the menu is restricted by value of a *threshold attribute*. Navigation continues with the step 1.

2.3 Delivering the course

Currently we are developing a web-based delivery system, which is able to use such enhanced space of learning objects and provide described navigation scheme. Some of the key features of the system are:

Threshold attributes – currently the system defines one threshold attribute – *number of entries*, which defines the amount of menu entries, which should be presented to the student. Valid entries for this attribute are numerical value, which directly represent the number of shown entries in menu or a constant representing value all entries – the system then show all actually available entries.

The attribute can be defined on system level as default value. If a new course is created the default value is copied into the course settings. The teacher of the course can change this attribute thus overriding the default value. If a student is enrolled into the course, the course value is copied into his profile. Optional force attribute can be specified by the teacher, which then disables the changes of value by the student.

Testing knowledge – is a key variable in the feedback from the student. The student is bound to take some tests during the course to prove his gained knowledge. We would like to use the result of the test in the process of adaptation and to update achieved knowledge set.

3 Future work and conclusions

In this paper we presented the idea of concepts space based course recommendation scheme. Ongoing research and current development of a web-based system was discussed.

Our future work in this area will focus on extending the definition with situations where the course is enhanced with additional learning objects or on the other hand some are removed. Our immediate goal is to finish the implementation of the web-based delivery system and provide our students with the course as soon as possible. The results from the run of such delivered course we would like to compare with result from previous years (presented in [2]), where the course was distributed to students as sequential.

Acknowledgement: This work was partially made with connection to the project Kontakt MEB080878.

References

1. Andrejko A., Bieliková M. Investigating Similarity of Ontology Instances and Its Causes. *ICANN 2008 – 18th International Conference on Artificial Neural Networks*, Prague, Czech Republic 2008. pp. 1-10. ISBN 978-3-540-87558-1
2. Šaloun P., Velart Z. Adaptive Hypermedia as a mean for learning programming. *In Workshop proceedings of the* 6^{th} *ICWE 2006.* ACM Digital Library, 2006.
3. Velart Z., Šaloun P. Ontology Based Course Navigation. *In HT'07 Proceedings.* Manchester, UK, 2007, pp. 151-152.
4. Velart Z., Šaloun P., Kalinská M. Znalostmi řízený průchod kurzem. *1st Workshop on Intelligent and Knowledge oriented Technologies – WIKT 2006 Proceedings.* 2006. Bratislava, Slovakia. ISBN 978-80-969202-5-9

Podpora kolaborácie používateľov rozšírením komunikačného nástroja na báze sémantického spracovania

Ivan Kapustík, Viera Rozinajová, Peter Bartalos

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií
Slovenská technická univerzita v Bratislave
Ilkovičova 3, 842 16 Bratislava, Slovakia
{kapustik,rozinajova,bartalos}@fiit.stuba.sk

Abstrakt. Článok zachytáva aktuálny stav rozpracovania čiastkovej úlohy Kolaborácia v projekte SEMCO-WS. Venuje sa podpore zdieľania informácií, znalostí a skúseností jednotlivých používateľov. Stručne v ňom predstavujeme sémantické rozšírenie nástroja na komunikáciu používateľov. V závere sú námety na ďalšie možnosti podpory kolaborácie používateľov.

Kľúčové slová: kolaborácia, chat, ontológia, sémantické spracovanie dokumentov

1 Úvod

Príspevok sa venuje analýze možností podpory kolaborácie používateľov pri riešení komplexných problémov. Dôležitým aspektom je možnosť zdieľania informácií, znalostí a skúseností jednotlivých používateľov.

Jednou z ciest zabezpečenia spomenutej funkcionality je hľadanie možností čo najefektívnejšieho rozšírenia bežne zaužívaných komunikačných nástrojov o možnosť spracovania a následného sprístupnenia informácií zachytených v komunikácii. Ďalšou možnosťou je poskytnutie doplnkových informácií, ktoré môžu výrazne urýchliť nájdenie optimálneho riešenia v danej situácii. Tieto vylepšenia chceme realizovať pomocou sémantického spracovania komunikácie, ktorá prebieha medzi používateľmi v rámci chatu.

Uvedenej problematike sa venujeme v rámci výskumného projektu SEMCO-WS (Sémantická kompozícia webových a gridových služieb), kde sa sústreďujeme na vytvorenie nástrojov na kolaboráciu používateľov. Výskum sa zameriava najmä na možnosti efektívnej komunikácie skupiny používateľov systému a na metódy na zachytávanie a zdieľanie skúseností používateľov, ktoré chceme následne ponúknuť používateľom, aby sme čo najefektívnejšie podporili ich prácu.

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 13–16, ISBN 978-80-227-3027-3.

2 Kolaborácia

Kolaborácia je proces, kde dvaja alebo viac ľudí pracuje na dosiahnutí spoločného cieľa, využívajúc pritom zdieľanie znalostí. Vychádzajúc z takto definovanej kolaborácie, hľadáme spôsob, ako ju podporiť tak, aby v čo najväčšej miere urýchlila a zefektívnila dosiahnutie daného cieľa [1]. Aplikačná oblasť, v ktorej pracujeme, je oblasť krízového manažmentu. Táto oblasť, ako je známe, vyžaduje obzvlášť dobre zohratú tímovú prácu. Členovia tímu sú pozorne vyberaní a často pracujú s utajenými informáciami. Ich komunikácia je neverejná a môže obsahovať informácie, ktoré môžu byť v budúcnosti pri podobných situáciách nápomocné. Ak by sa podarilo nejakým spôsobom komunikáciu tímu vhodne uložiť a v prípade potreby on-line sprístupniť, mohlo by to znamenať efektívny posun v podpore kolaborácie.

V súčasnosti existuje množstvo nástrojov, ktoré podporujú priamu alebo nepriamu komunikáciu používateľov, správu a rozdeľovanie úloh, či iné kolaboratívne činnosti [2]. Analýzou našej špecifickej úlohy sme dospeli k záveru, že najväčším prínosom by bola podpora kolektívneho hľadania riešenia problému. Túto sme sa rozhodli realizovať prostredníctvom priamej komunikácie používateľov s možnosťou rozšírenia o odkazy na iné zdroje znalostí.

Samotné použitie nástroja na elektronickú komunikáciu – chat – je však len základnou požiadavkou – aby sme mohli zachytiť prebiehajúcu komunikáciu. Na efektívnejšiu podporu hľadania riešenia problémov zo zvolenej domény ešte potrebujeme vhodnú formu automatického porozumenia komunikovaného textu. Od tohto porozumenia sa očakáva rýchla reakcia na písaný text. To znamená, že do niekoľkých sekúnd je treba vyfiltrovať písaný text, nájsť jeho významové ohodnotenie a získať zodpovedajúcu informáciu na podporu rozhodovania. Dlhšie časy môžu znamenať neaktuálnosť reakcie systému, lebo komunikácia sa už presunula ďalej. Viaceré realizácie predspracovania a ohodnocovania textu však tieto požiadavky len ťažko spĺňajú [3, 4]. Rozhodli sme sa preto využiť postup, ktorý bude vo všetkých fázach vychádzať z doménovej ontológie.

3 Komunikačný nástroj

Naším prvým cieľom bolo vytvoriť komunikačný nástroj, ktorý poskytuje štandardné možnosti zdieľania informácií – ako napr. bežne používaný chat. Rozhodli sme sa vytvoriť z existujúcich komponentov [5] vlastnú implementáciu nástroja, aby sme ho mohli prispôsobiť špecifickým požiadavkám a aby sme do neho mohli dorábať potrebnú funkcionalitu. Následne bolo našou snahou rozšíriť tento nástroj tak, aby poskytol možnosti sprístupnenia informácií zaznamenaných v rámci jednotlivých komunikačných celkov ako aj poskytnutie ďalších relevantných informácií k diskutovaným témam.

Náš prístup sme založili na sémantickom spracovaní, keďže práve v tejto oblasti vidíme možnosti hľadania nových riešení v oblasti spracovania a sprístupnenia informácií. V projekte SEMCO-WS, ktorého zameranie je podstatne širšie a rieši problémy sémantickej kompozície webových a gridových služieb, sa vytvárajú ontológie, ktoré budú slúžiť pre potreby tejto kompozície. Aj z tohoto aspektu sa

ponúka možnosť využiť existujúcu ontológiu z domény krízového manažmentu a pokúsiť sa namapovať zaznamenaný text na pojmy z nej, resp. určiť príslušnosť pojmov z textu k význačným konceptom z ontológie. Tým sa otvárajú možnosti spoľahlivejšieho určenia témy komunikácie a zvýši sa možnosť poskytnutia relevantných dokumentov.

4 Sémanticky rozšírený chat

Úlohou navrhnutého nástroja je počas online diskusie používateľov zaznamenávať obsah jednotlivých príspevkov a rozčleniť ich podľa zadaného kritéria na jednotlivé diskusné jednotky. Rozdelenie na takéto jednotky možno realizovať viacerými spôsobmi – napr. zohľadniť časové hľadisko (každý príspevok oddelený určitou nastavenou prestávkou bude považovaný za novú jednotku). Diskusné jednotky budú ďalej oddeľovať texty, kde sa nenájde príslušnosť k doménovej ontológii – používatelia diskutujú o niečom, čo sa netýka známej oblasti krízového manažmentu.

Všetky ukončené diskusné jednotky sa archivujú spolu s ich automatickým ohodnotením významu a prípadným ohodnotením relevantnosti od používateľov. Aktuálna diskusná jednotka sa priebežne vyhodnocuje a hľadá sa jej význam. Pretože sa typicky jedná o krátke texty, ktoré môžu rýchlo meniť význam, ohodnotenie významu prebieha na dvoch úrovniach – získanie kontextu diskusie a získanie významu poslednej vety alebo príspevku do diskusie. Systém následne dokáže vyhľadať súvisiace dokumenty so zhodným alebo podobným významom a zobraziť používateľovi ich zoznam.

V ďalšom rozšírení prístupu sa využijú informácie z predošlých diskusií a korešpondujúcich prehliadaných dokumentov používateľmi v danom čase pre vybrané kategórie. Pomôže mu to pri kvalitnejšom určovaní relevantnosti prehliadaných dokumentov, čo môže bližšie špecifikovať dokumenty, ktoré boli nápomocné pri riešení určitej situácie. Nástroj teda zahŕňa automatické sémantické ohodnotenie práve prebiehajúcej diskusie a výber archivovanej diskusie s podobným ohodnotením. Používateľ má zároveň možnosť označiť, nakoľko mu poskytnutá informácia pomohla pri hľadaní riešenia jeho problému. Informácie, ktoré sú sémanticky podobné, ale majú nízky prínos, budú mať výrazne zníženú preferenciu zobrazenia.

Samotné sémantické ohodnocovanie bude založené na vyhľadaní slov z diskusie v ontológii a vytvorenie vektora priradení k existujúcim ontologickým konceptom a ich vzťahom [6]. Miera podobnosti medzi týmito vektormi určuje základnú prioritu zobrazovania nájdenej informácie.

5 Záver a námety pre ďalšiu prácu

Výhody nášho riešenia oproti známym existujúcim riešeniam sú hlavne v efektívnosti riešenia, lebo spracovávame len relevantnú komunikáciu a systém sa neodkláňa od definovanej témy. Používateľ sa prostredníctvom poskytovanej informácie môže „zúčastniť“ na relevantnej diskusii, ktorá prebehla v jeho neprítomnosti. Nevýhodou

nášho prístupu je nutnosť úpravy ontológie, ak chceme podchytiť komunikáciu aj v inej problémovej oblasti.

V článku sme predstavili návrh na sémantické obohatenie komunikácie používateľov, ktorá prebieha pomocou štandardných nástrojov komunikácie ako napr. chat. Keďže návrh metódy výberu relevantných informácií zo zachytenej komunikácie ako aj nástroj, ktorý uvedenú metódu bude implementovať, sú v štádiu rozpracovania, v článku ešte neprezentujeme konkrétne výsledky. Zamerali sme sa skôr na myšlienky vedúce k návrhu metódy a opis návrhu funkcionality implementovaného nástroja.

Ďalšou možnosťou rozšírenia funkcionality navrhnutého nástroja ja zabudovanie metód a techník, ktoré nielen umožňujú hladkú komunikáciu a prístup ku zaznamenaným diskusiám, ale snažia sa aktívne podporiť proces vyriešenia diskutovaného problému. Príkladom môže byť Delfská metóda [7]. Možno ju charakterizovať ako metódu na štruktúrovanie procesu skupinovej komunikácie tak, aby sa tento proces zefektívnil a našlo sa riešenie komplexného problému. Na dosiahnutie tejto „štruktúrovanej komunikácie“ je potrebné zabezpečiť: spätnú väzbu na individuálne príspevky, posúdenie skupinového názoru, možnosť revidovať názory jednotlivcov a určitý stupeň anonymity. Uvedenej problematike sa plánujeme venovať v našej ďalšej práci.

Poďakovanie: Práca na projekte bola podporená grantovou agentúrou APVV (projekt 0391 06 SEMCO-WS).

Literatúra

1. Laclavik, M., Seleng, M., Hluchy, L.: User Assistant Agent (EMBET): Towards Collaboration and Knowledge Sharing in Grid Workflow Applications. In Cracow '06 Grid Workshop: K-Wf Grid. Cracow, Poland, 2007, ISBN 978-83-915141-8-4, pp. 122-130
2. Special Issue on Collaboration Support Systems. IEEE Transactions on Systems, Man and Cybernetics, Part A. Vol. 36. Issue 6. Nov. 2006
3. Douglas W. Oard and D. W. Oard.: The state of the art in text filtering. In UMUAI, Vol. 7, pp. 147-178. 1997.
4 Bos, J.: Three Stories on Automated Reasoning for Natural Language Understanding. in: Proceedings of ESCoR (IJCAR Workshop): Empirically Successful Computerized Reasoning. pp. 81-91. 2006
3. Saggion, H., Funk, A., Maynard, D., Bontcheva, K.: Ontology-based Information Extraction for Business Applications. In Proceedings of the 6th International Semantic Web Conference (ISWC 2007), Busan, Korea, November, 2007.
4. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*. Philadelphia, July 2002.
6. Maynard, D., Peters, W., Li., Y.: Metrics for Evaluation of Ontology-based Information Extraction. In Proc of WWW 2006 Workshop on "Evaluation of Ontologies for the Web" (EON 2006), Edinburgh, Scotland, 2006.
7. Linstone, H.A., Turoff, M.: The Delphi Method: Techniques and Applications, 2002, http://is.njit.edu/pubs/delphibook/index.html (Accessed August 2008) ISBN 0-201-04294-0

Zisťovanie vzájomnej podobnosti výučbových konceptov

Marián Šimko, Mária Bieliková

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií, Slovenská technická univerzita
Ilkovičova 3, 842 16 Bratislava, Slovensko
{simko,bielik}@fiit.stuba.sk

Abstrakt. Súčasné systémy pre e-vzdelávanie stále viac používajú dômyselné mechanizmy prispôsobovania, ktoré pracujú so znalosťami o príslušnej doméne. Opísanie doménovej oblasti tak, aby sme mali dostatok metadát pre personalizáciu, je však pre tvorcu obsahu namáhavá a časovo náročná činnosť. Problematické je najmä získavanie vzťahov medzi jednotlivými znalosťami, ktoré sa prezentujú študentovi a ktoré sú pre prispôsobovanie nevyhnutné. V tomto príspevku prezentujeme metódu pre zisťovanie vzájomnej podobnosti výučbových konceptov, na základe ktorej je možné automaticky identifikovať vzťahy medzi konceptmi v doménovom modeli výučbovej aplikácie.

1 Prispôsobovanie v kontexte e-vzdelávania

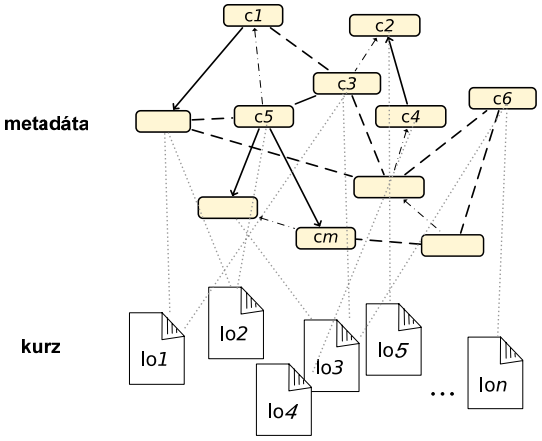
Personalizácia, jedna z čŕt súčasného trendu označovaného aj ako druhý web alebo Web 2.0, zohráva obzvlášť dôležitú rolu pri vzdelávaní. Predkladanie študijného materiálu šitého na mieru, používateľsky-špecifická navigácia v informačnom priestore, to sú len niektoré z výhod adaptívnych výučbových systémov, ktoré umožňujú študovať dôslednejšie a efektívnejšie.

Pri prispôsobovaní sa využíva model používateľa, ktorý sa najčastejšie realizuje prekryvne: kopíruje doménový model, pričom nesie informácie o študentových znalostiach jednotlivých konceptov [1]. Koncepty doménového modelu sú navzájom prepojené vedomostné elementy, fragmenty informačného priestoru, ktoré z pohľadu cieľov výučby predstavujú pre študenta určitú pridanú hodnotu. Použijúc metaforu, doménový model je kognitívna mapa kurzu, ktorú štúdiom prenášame do nášho vlastného kognitívneho systému. V procese tvorby kurzu takéto mapy zvyčajne nie sú explicitne dostupné. Pre potrebu prispôsobovania ich treba vytvoriť. To je však pre autora kurzu netriviálna úloha. Počet konceptov sa v priemernom kurze ráta v desiatkach, počet relácií až v stovkách čo prakticky znemožňuje manuálnu tvorbu doménového modelu.

Problémom je aj samotná forma výučbových materiálov, ktoré sú najčastejšie v textovej podobe. Časť výpovednej hodnoty strácajú už v momente svojho vzniku, kedy sú „zakódované“ do – z pohľadu strojov – prúdu znakov. Autor kapitoly či lekcie ani pri najlepšej vôli do textu nezapíše všetko to, čo chce odkázať svojim študentom. A len málokedy explicitne vzájomne previaže súvisiace kapitoly, resp. výučbové materiály, ktoré sú dostupné.

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 17–20, ISBN 978-80-227-3027-3.

Opísaný problém sa dá sformulovať ako potreba „rekonštrukcie“ znalostí obsiahnutých v texte do podoby čitateľnej strojom. Cieľom je automatické vytvorenie doménového modelu reprezentujúceho metadáta kurzu len z dostupných výučbových materiálov. Na základe metadát sa usudzuje pre realizáciu prispôsobovania (obr. 1).



Obr. 1. Pohľad na doménový model adaptívneho kurzu. Kurz je zostavený z objektov výučby (lo*1*, lo*2*, ...), nad ktorými sa nachádza metamodel domény – priestor konceptov (c*1*, c*2*, ...) pripomínajúci ontológiu. Notácia relácií medzi konceptmi vyjadruje ich odlišnú sémantiku.

2 Východiská pre rekonštrukciu znalostí

Zdroje pre rekonštrukciu znalostí o doméne sú obmedzené na výučbové materiály, ktoré sa prezentujú v procese výučby. Všetky informácie, ktoré máme v čase tvorby kurzu k dispozícii sa týkajú *obsahu* objektov výučby (ten môžeme rozdeliť na textový obsah a formu) a *hierarchických vzťahov* medzi objektmi výučby.

Východiskom pre spracovanie textového obsahu sú v tomto prípade metódy a techniky dolovania v texte (angl. text mining). V praxi sa využívajú kategorizačné, klasifikačné a zhlukovacie algoritmy [2, 3] a algoritmy spracovania prirodzeného jazyka. Nezanedbateľnú sémantiku, ktorá môže byť využitá pre získanie metadát, majú vizuálne prvky ako nadpisy, odseky, fonty a pod. Spracovanie týchto atribútov závisí spravidla od reprezentácie objektov výučby [4]. Hierarchické vzťahy medzi objektmi výučby zahŕňajú explicitné členenie textu na kapitoly/podkapitoly a vzájomné odkazy v texte. Tieto informácie môžu pomôcť k štrukturalizácii metadát.

Všetky uvedené aspekty využívame pri automatizovanom generovaní metadát kurzu, čím vytvárame časť doménového modelu [5]. Nosnou časťou generovania a zároveň ťažiskom nášho výskumu je odhaľovanie vzájomnej podobnosti konceptov.

3 Odhaľovanie vzájomnej podobnosti konceptov

Odhaľovanie vzájomnej podobnosti konceptov je jedným z krokov metódy automatizovaného generovania metadát kurzu [5]. Prichádza na rad po analýze

a spracovaní obsahu objektov výučby, kedy sa vygenerujú kľúčové slová, ktoré majú „potenciál“ stať sa konceptmi. Vo fáze finalizácie doménového modelu o tom rozhodne samotný používateľ, ktorý vyhodnotí správnosť automatického generovania.

Doménový model na začiatku odhaľovania podobnosti konceptov obsahuje ohodnotené väzby medzi objektmi výučby (učebnými textami) a kľúčovými slovami, ktoré predstavujú kandidáty na koncepty (situácia je podobná ako na obr. 1, ale metadáta nie sú vzájomne prepojené). Váha väzby, ktorú zistíme analýzou textu, vyjadruje mieru príslušnosti konceptu k danému objektu výučby. Na základe týchto väzieb a väzieb medzi jednotlivými objektmi výučby hľadáme vzťahy medzi konceptmi. Pre každý koncept zostrojíme zoznam susedov, z ktorého vyberieme *k* najpodobnejších a vytvoríme relácie v doménovom modeli. Voľba premennej *k* je jedno z najdôležitejších miest celej metódy a sama osebe je predmetom širšieho výskumu. Zatiaľ najsľubnejšie výsledky priniesla metóda výpočtu využívaná v kategorizačných algoritmoch [2].

Pre výpočet podobnosti konceptov sme navrhli niekoľko variantov, ktoré sa odlišujú pohľadom na aktuálny stav doménového modelu:

- vektorový prístup,
- šírenie aktivácie,
- analýza založená na algoritme PageRank.

Pri *vektorovom prístupe* vytvárame vnútornú reprezentáciu konceptov ako váhovú sumu vektorov objektov výučby, ku ktorým prislúchajú. Podobnosť medzi konceptmi je vyjadrená kosínusovou vzdialenosťou. Tento variant sa spolieha na dôkladnú lexikálnu analýzu zdrojových textov a presnú vnútornú reprezentáciu objektov výučby. Jeho úspešnosť je najviac spätá s úspešnosťou metód a techník spracovania prirodzeného jazyka. Stretávame sa tu s problémami homonymie, polysémie, a pod.

V prípade využitia *šírenia aktivácie* hľadíme na doménový model ako na kontextovú sieť. Máme dva typy uzlov: objekty výučby a koncepty. Informácie, ktoré sú implicitne prítomné pre objekty výučby (tu sú nám najviac nápomocné hierarchické vzťahy medzi nimi), sa snažíme „namapovať“ do priestoru konceptov. Každý koncept postupne aktivujeme počiatočnou energiou, ktorú prešírime do celého grafu cez relácie príslušnosti. V ostatných konceptoch sa postupne naakumuluje energia, ktorá vyjadruje podobnosť voči aktivačnému konceptu.

Tretím variantom je analýza založená na algoritme *PageRank***.** Ten stavia na analógii s webom. Z aktuálneho stavu doménového modelu zostrojíme dočasný graf, ktorý obsahuje iba koncepty. Váhy a orientácia dočasných relácií (ktoré vznikli prepojením objektov výučby incidujúcich s rovnakým objektom výučby) chápeme ako počet odkazov vedúcich na jednotlivé stránky na webe. Pre každý koncept potom aplikujeme rozšírený algoritmus PageRank [6], ktorého výsledkom je implicitné ohodnotenie prestíže každého konceptu. Pomocou pomeru prestíží vyjadríme vzájomnú podobnosť konceptov.

4 Experiment a zhodnotenie

Uvedenú koncepciu odhaľovania podobnosti konceptov premietnutú do troch variantov sme otestovali na kurze funkcionálneho programovania, ktorý sa prednáša

na FIIT STU v Bratislave. Vygenerované relácie sme porovnávali s referenčnými vytvorenými manuálne pre potreby experimentu. Úspešnosť generovania sme vyhodnocovali použitím miernej modifikácie známych metrík úplnosť (angl. recall) a presnosť (angl. precision). Výsledky sú zobrazené v tabuľke 1.

Tabuľka 1. Výsledky experimentu.

Variant	Úplnosť (P)	Presnosť (R*)	Metrika F (F*)
Vektorový prístup	0.509	0.793	0.620
Šírenie aktivácie	0.443	0.784	0.566
PageRank-analýza	0:569	0.741	0.652

Je ťažké zhodnotiť, či možno výsledky považovať za úspech alebo nie. V oblasti výučbových systémov ich nevieme porovnať, pretože nám nie sú známe práce s podobným zameraním, a pre úroveň spracovania textu sú výsledky príliš špecifické. Isté je, že experiment poukázal na možnosti viacerých vylepšení. Náš výskum sa bude v najbližšej dobe sústreďovať najmä na analýzu možnosti použitia zhlukovacích algoritmov, ktoré môžu pomôcť usporiadať zostrojený priestor vedomostí, a na overenie v spojitosti s využitím takto vytvoreného doménového modelu pri výučbe pomocou adaptívneho systému pre vzdelávanie [7].

Poďakovanie. Táto práca bola čiastočne podporená Kultúrnou a edukačnou grantovou agentúrou MŠ SR, grant č. KEGA 3/5187/07.

Referencie

1. Brusilovsky, P. Developing adaptive educational hypermedia systems: From design models to authoring tools. In Murray, T. et al. (eds.): *Authoring Tools for Advanced Technology Learning Environment*. Kluwer, 2003, pp. 377-409.
2. Diedrich, J., Balke, W-T. The Semantic GrowBag Algorithm: Automatically Deriving Categorization Systems. In *Proc. of the 11th European Conf. on Research and Advanced Technology for Digital Libraries*, 2007, pp 1-13.
3. Fortuna, B., Grobelnik, M., Mladenic, D. OntoGen: Semi-automatic Ontology Editor. In Smith M. J., Salvendy G. (Eds.): *Human Interface, Part II, HCII 2007, LNCS 4558*, pp. 309–318.
4. Roberson, S., Dicheva, D. Semi-automatic ontology extraction to create draft topic maps. In *Proc. of the 45th Annual ACM Southeast Regional Conf.*, Winston-Salem, North Carolina, 2007, pp. 100-105.
5. Šimko, M. *Odhaľovanie vzťahov v obsahu výučbového kurzu*. Bratislava, 2008. 58s. Diplomová práca, FIIT STU v Bratislave. Vedúca práce: prof. M. Bieliková.
6. White, S., Smith, P. Algorithms for estimating relative importance in networks. In *Proc. of the 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data mining*. ACM Press, 2003, pp. 266-275.
7. Šimún, M., Andrejko, A., Bieliková, M.: Maintenance of Learner's Characteristics by Spreading a Change. In *WCC 2008, IFIP, Volume 281, Learning to Live in the Knowledge Society*, M. Kendall et al., Boston: Springer, pp. 223–226.

Znalostný priestor so sémantikou pre zjednodušenie sprístupňovania informácií

Tomáš Kuzár

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava, Slovensko
kuzar@fiit.stuba.sk

Abstrakt. Rozvoj webových technológií, ktorý umožnil bežným používateľom internetu podieľať sa na tvorbe jeho obsahu prostredníctvom blogov, wiki portálov alebo profilov v rámci sociálnych sieti, mal za následok zahltenie webového priestoru nepreberným množstvom informácií. Do popredia sa preto dostava otázka vyhľadávania a sprístupňovania informácií v tomto priestore. V článku je prezentovaná myšlienka sprístupňovania informácií s ohľadom na sémantický význam sociálnych väzieb používateľa.

1 Úvod

Ešte nedávno boli bežne dostupné stránky na internete implementované ako jednoduché statické prezentácie. Obsah portálov bol výhradne v rukách ich tvorcov. Situácia sa zmenila s príchodom webu s prívlastkom Web 2.0. Ten umožnil aj bežným používateľom editovať obsah webových portálov a tak vznikli projekty, ktoré nazývame sociálne média: Wikipédia, blogy, sociálne siete, diskusné fóra, videá. Rýchly nárast objemu publikovaného obsahu má za následok zahltenie používateľa, ktorý sa v veľkom množstve informácií stráca. Nevyhnutnosťou sa tak stávajú pokročilé metódy vyhľadávania a sprístupňovania informácií na webe, napr. použitie sémantického webu a sémantického anotovania.

Snaha o sprístupnenie veľkého množstva informácií je podnetom pre vytvorenie znalostného priestoru, ktorý by aktívne podporoval prácu a učenie sa v prostredí webu. V tomto smere môžeme spomenúť prístupy, ktoré môžu byť použité na podporuru vyhľadávania na webe: sémantický web, filtrovanie aktivít používateľa alebo aktívne sémantické štruktúrovanie webu [1].

2 Inteligentné vyhľadávanie v prostredí internetového denníka

Našim cieľom je navrhnúť systém, ktorý umožní používateľom využívať internetové noviny obohatené o funkcionalitu znalostného priestoru. Bežne dostupné internetové noviny však používateľovi poskytujú len funkcionalitu primárnej úrovne (sufrovanie po stránke, čítanie článkov, prispievanie do diskusií, publikovanie článkov na blogu)

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 21–24, ISBN 978-80-227-3027-3.

a tak internetový denník nie je možné považovať na znalostný priestor. Články sú na pridávané priebežne a manuálne triedené do jednotlivých kategórií.

V prípade, že by sa používateľ pohyboval v prostredí znalostného priestoru, systém by mu automaticky sprístupnil články reflektujúce jeho záujem. Odporúčanie článkov by záviselo od sociálnej komunity, do ktorej patrí a sufrovacích vzorov, ktoré v danej komunite prevládajú. Domnievame sa totiž, že ľudia s podobnými záujmami majú podobné vyhľadávania informácií na webe.

3 Aktivity používateľa na webe doplnené o sémantický kontext

Na základe analýzy logu z webového vieme popísať spôsob, ako sa používatelia na portáli pohybovali. Dokážeme určiť chronológiu navštevovania jednotlivých článkov, teda v istom zmysle sme od používateľov získali informáciu o náväznosti jednotlivých článkov, tém a slov.

Pri manuálnom a semi-automatickom sémantickom anotovaní sa vo veľkej miere spoliehame na činnosť používateľa, ktorý jednotlivé webové stránky anotuje. Na druhej strane pri automatickom sledovaní aktivity používateľa, počítač sám sleduje prácu používateľa bez toho, aby to používateľ zaregistroval.

Nami navrhované priebežné automatické anotovanie spája sledovanie pohybu používateľa na webe so sémantikou stránok alebo príspevkov, ktoré si prezrel. Každú stránku vieme sémanticky určitým spôsobom reprezentovať – napr. výberom niekoľkých kľúčových slov. Na základe sledovania pohybu používateľa medzi článkami získame sémantické vzťahy medzi kľúčovými slovami, ktoré sa v článkoch vyskytujú. Týmto spôsobom sa stáva bežný používateľ spolutvorcom sémantickej anotácie.

Ako príklad použitia aktívneho sémantického anotovania je možné uviesť internetové noviny. Články v internetových novinách sú pridávané denne, sú členené do rubrík a najčítanejšie články patria väčšinou medzi odporúčané. Tento spôsob odporúčania je založený na štatistikách čítanosti jednotlivých článkov. V žiadnom prípade však neberie do úvahy širšie súvislosti – sémantický význam článkov, sémantická podobnosť medzi po sebe navštívenými článkami, čas strávený používateľom na jednotlivých stránkach a podobne.

4 Sociálna sieť blogera doplnená o sémantický kontext

Na druhej strane máme používateľov blogu – blogerov. Blogeri tvoria sociálnu sieť, lebo sa navzájom sa seba odkazujú. Aktívny bloger nám poskytuje veľa informácií o ňom samom – osobný profil, publikované články, príspevky v diskusiách a odkazy na svojich obľúbených blogerov [2]. Z týchto informácií je možné extrahovať sémantický profil blogera.

Našou snahou vytvoriť znalostný priestor, ktorý bude reagovať na potreby blogerov. Našim cieľom je vytvoriť prepojenie medzi profilom blogera a článkami, ktoré spadajú do jeho záujmu. Za významnú časť profilu blogera pokladáme jeho odkazy na ostatných blogerov, teda sieť jeho sociálnych kontaktov. V prípade, že

vygenerujeme blogerom vhodné internetové odkazy, zvýšime pridanú hodnotu sociálnej siete medzi blogermi a tiež postavíme základy znalostného priestoru v prostredí internetového denníka.

5 Analýza surfovacích vzorov a sociálnej siete

Naša predstava znalostného priestoru pozostáva z dvoch základných častí: analýza webového logu servera a analýza profilov členov sociálnej siete. Z logu získame vzťahy medzi článkami na základe analýzy sekvencií používania, z profilov používateľov získame vzťahy medzi článkami na základe analýzy sociálnych väzieb používateľov.

Každý webový server má logový súbor, v ktorom je zaznamená aktivita používateľov daného webu: identifikácia používateľa, čas prístupu, navštívená stránka. Na základe týchto údajov dokážeme odmerať čas, ktorý strávil na jednotlivých stránkach, vieme používateľovi priradiť obsah, ktorý si na portáli pozeral. Obrátene tiež vieme určiť stránky, ktoré majú rovnakého čitateľa. Tým pádom získame prepojenia medzi stránkami, ktoré budú dané používateľom. Analýzou správania viacerých používateľov, dokážeme popísať významné časté surfovacie vzory v rámci portálu. Za významný vzor budeme považovať taký, ktorý sa veľmi často vyskytuje u konktrétneho používateľa. V celej vzorke častý byť však nemusí.

Analýzou webového logu získame niekoľko rôznych sekvenčných vzorov. Tieto vzory nebudú obsahovať iba sekvenciu navštívených odkazov, ale aj ďalšie informácie ako napríklad: dĺžka článku, releventnosť zdroja, komplexnosť článku a ďalšie meta-informácie. Sekvenčný vzor bude mať teda pomerne vysokú výpovednú hodnotu.

Okrem odhalenia relevantných vzorov, chceme tiež určiť relevantných používateľov – autority. Správanie autorít môžeme tiež použiť ako vzor pri ďalšom spracovaní. Za autoritu môžeme považovať používateľa, ktorý týka veľký počet článkov týkajúcich sa jednej témy (sémanticky podobné) , následne prispieva do diskusií a jeho diskusné príspevky často pozitívne komentované. Okrem toho autoritu môžeme objaviť aj v sociálnej sieti – napr. bloger, na ktorého sa mnohí odkazujú. Významného blogera môžeme považovať za autoritu a pri vizualizácii sociálnej siete sa javí ako centrálny uzol. Táto časť bola aj experimentálne overená. Napríklad v testovacej vzorke sme objavili dva zaujímavé centrálne uzly – blogeri Cerven a Hermana. Obaja sú často odkazovaní. Obaja blogeri sa venujú duchovnej tématike, ich tvorba je sémantiky podobná, a je to podložené aj tým, že niektorí používatelia sa odkazujú na Cervena a zároveň aj na Hermanu.

6 Vytvorenie a nastavanie modelu

Našim cieľom je odhaliť prepojenie medzi používaním webu a sociálnymi väzbami používateľov. Toto prepojenie vytvoríme na základe sémantickej podobnosti článkov – na jednej strane ide o podobnosť vyplývajúcu zo sekvenčných vzorov, na druhej

strane ide o podobnosť vyplývajúcu zo sociálnej siete používateľa. Budeme reprezentovať dve skutočnosti: sekvenčné vzory a profily používateľov.

V prvom rade potrebujeme vhodne reprezentovať jednotlivé sufrovacie vzory. Predpokladané parametre pre model sekvenčného vzoru:

- Dĺžka vzoru (počet stránok)
- Frekvencia výskytu vzoru
- Výskyt konceptov

Predpokladané parametre pre reprezentáciu používateľa:

- Doménová oblasť používateľa (bloggera)
- Počet publikovaných článkov
- Témy článkov – koncepty
- Veľkosť komunity bloggera a počet jeho sociálnych väzieb
- Komplexnosť vyjadrovania (priemerná dĺžka slov, odborné výrazy)

7 Záver

Za podstatu našej metódy sme si zvolili kombináciu dvoch prístupov – analýza aktivity používateľa pri pohybe medzi článkami a sociálna sieť blogera. Chceme overiť, či existuje vzťah medzi používaním webu a sociálnou sieťou. Predpokladáme, že používanie webu spája sémanticky podobné články. Predpokladáme tiež, že aj sociálna sieť dáva dokopy sémanticky podobné články. Ak sa nás predpoklad potvrdí, segmentujeme populáciu do malých skupín a v rámci skupiny použijeme odporúčanie na základe preferencií ostatných členov v skupine a zároveň na základe sekvenčných vzorov používania.

Referencie

1. Geissler S., Hampel T.: Static, Dynamic and Active Interaction Structures in the Web: Providing Semantic Annotations in the Virtual Knowledge Space
2. Yao L. et. al.: A Unified Approach to Researcher Profiling

Databázy
a informačné systémy

Použitie MapReduce architektúry na spracovanie veľkých informačných zdrojov

Martin Šeleng, Michal Laclavík, Ladislav Hluchý

Ústav Informatiky, Dúbravská cesta 9, 845 07 Bratislava, Slovensko
{Martin.Seleng,Michal.Laclavik,Ladislav.Hluchy}@savba.sk

Abstrakt. V príspevku opisujeme architektúru MapReduce vhodnú na spracovanie veľkých informačných zdrojov. Zároveň opisujeme dostupnú open source implementáciu tejto architektúry – Hadoop. Hadoop je javovskou implementáciou frameworku MapReduce navrhnutého a vyvinutého spoločnosťou Google na spracovanie a generovanie veľkých objemov dát. Výhoda MapReduce spočíva v jednoduchom použití a správe distribuovaného systému, kde vývojár aplikácie nemusí dokonale poznať aricihitektúru systému. Stačí ak naprogramuje Map a Reduce metódy ktoré si môže odladiť aj na svojom PC.

Kľúčové slová: MapReduce, Hadoop, HBase, distribuované spracovanie dát, distribuovaný súborový systém, BigTable, HDFS

1 Úvod

Ak chceme v dnešnej dobe spracovávať veľké množstvá dát, ktoré dnes poskytuje internet, potrebujeme na to množstvo počítačov a spôsob ako paralelne a rýchlo spracovávať tieto informácie. Je preto potrebné správne navrhnúť architektúru celého systému, a to hlavne:

- spôsob distribuovania dát,
- množstvo a spôsob replikácie dát,
- spôsob (framework) paralelného spracovania,
- množstvo a typ uzlov, na ktorých bude spracovanie prebiehať,
- jednoducho škálovateľný systém, ktorý by bol efektívny a spoľahlivý.

Na paralelné a distribuované spracovanie dát existuje množstvo prístupov cez použitia super počítačov ako aj architektúrach založených na bežne dostupných PC ako clustrové počítanie založené na PVM [12], MPI [13], Condor [14] alebo integrácia distribuovaných dátových a výpočtových zdrojov v prostredí Gridu [15]. V prostredí paralelného a distribuovaného počítania sa však v poslednej dobe presadzuje architektúra MapReduce [1]. Táto architektúra vyšla z komerčného prostredia, kde je potrebné spracovávať rozsiahle informačné zdroje – Google, a je denne overovaná v praxi.

Existujú tri voľne dostupné implementácie MapReduce architektúry:

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 27–34, ISBN 978-80-227-3027-3.

- Hadoop [8], vyvinutý spoločnosťou Apache Foundation spolu s projektami Lucene a Nutch. Spoločnosť Yahoo! má momentálne Hadoop spustený na 10 000 jadrách [10] v produkčnom prostredí [11].
- Phoenix [9], vyvinutý na Stanfordskej Univerzite, implementovaný v C++.
- Disco [17], vyvinutý spoločnosťou Nokia, implementovaný vo funkcionálnom jazyku Erlang.

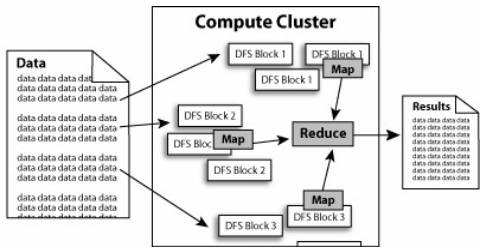
V príspevku predstavíme dva projekty spoločnosti Apache Software Foundation[1]: Hadoop[2] a HBase[3]. Hadoop je projekt, ktorý bol pôvodne vyvinutý ako infraštruktúra pre ďalší projekt spoločnosti Apache Nutch[4], ktorý sťahuje a indexuje stránky z webu. Obidva tieto projekty sú súčasťou projektu Lucene[5]. Hadoop je javovskou implementáciou frameworku MapReduce [1] navrhnutého a vyvinutého spoločnosťou Google na spracovanie a generovanie veľkých objemov dát a zahŕňa tiež implementáciu GFS (Google Files System) [3] nazývanú HDFS (Hadoop Distributed File System) [2]. HBase je podprojekt projektu Hadoop. Frameworku Hadoop poskytuje podobnú funkcionalitu ako BigTable [4] distribuovanému súborovému systému GFS. V mnohých veciach je podobný klasickým databázovým systémom. V príspevku zároveň predstavíme architektúru uvedených systémov, distribuovaný súborový systém a demonštrujeme jednoduchú aplikáciu a jej portovanie do frameworku MapReduce.

2 Architektúra

Prvým, kto takúto relatívne jednoduchú a hlavne funkčnú architektúru navrhol, bola spoločnosť Google. Prostredie, v ktorom sa v spoločnosti Google vykonávajú všetky operácie s dátami, sa dá opísať v nasledujúcich piatich bodoch (platných v čase zverejnenia článku [1]):

1. Jednotlivé pracovné stanice (uzly) majú štandardne dvojjadrové x86 procesory, beží na nich operačný systém Linux a majú 2-4GB pamäte (vo väčšine prípadov ide o bežné pracovné stanice, a nie servery s vysokým výkonom).
2. Používa sa bežné sieťové pripojenie s rýchlosťou 100Mb/s, resp. 1Gb/s na úrovni pracovnej stanice.
3. Kluster pozostáva zo stoviek až tisícok pracovných staníc (z toho dôvodu sú výpadky pracovných staníc bežné).
4. Používané disky sú vo väčšine prípadov lacné, s IDE rozhraním, priamo pripojené k pracovným staniciam. Používajú distribuovaný súborový systém GFS [3]. Systém štandardne používa replikáciu, aby zabezpečil dostupnosť a spoľahlivosť systému postaveného nad nespoľahlivým hardvérom.
5. Používateľ posiela do systému procesy, ktoré plánovač rozdelí na voľné pracovné stanice a spustí [1].

[1] http://www.apache.org/
[2] http://hadoop.apache.org/
[3] http://hadoop.apache.org/hbase/
[4] http://lucene.apache.org/nutch/
[5] http://lucene.apache.org/



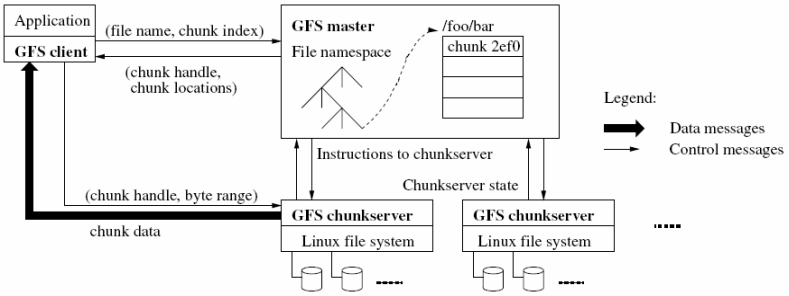
Obr. 1. MapReduce architektúra[6].

Základná idea je teda distribuovať dáta na jednotlivé nody v rámci distribuovaného file systému (pozri ďalšiu časť) a nad týmito dátami zabezpečiť spustenie Map metódy. Výsledky Map metódy sú vstupom pre metódu Reduce ktorá spracuje výstup Map metód do požadovaného výsledku.

2.1 Distribuovaný súborový systém

Na to, aby sme mohli spúšťať procesy v paralelnom prostredí, potrebujeme mať aj distribuovaný súborový systém, nad ktorým budeme tieto procesy spúšťať.

Na nasledujúcom obrázku sa nachádza architektúra a tok informácií v distribuovanom súborovom systéme GFS.



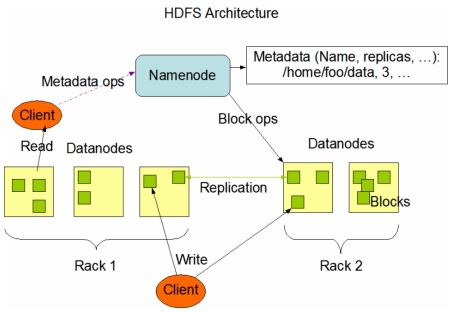
Obr. 2. Architektúra, dátový a riadiaci tok informácií v GFS [3].

Táto centralizovaná architektúra pri veľkých objemoch dát vyzerá v prípade jedného mastera veľmi zúžená. Avšak v GFS klient nepristupuje pri čítaní a zápise informácií uložených na pracovných staniciach cez GFS mastra, ale pri prvom prístupe dostane priamo meno GFS chunkservera, ku ktorému má pristúpiť, toto meno si zapamätá na určitú dobu a ďalej komunikuje priamo s chunkserverom.

Podobne aj Hadoop využíva 5 vyššie spomínaných bodov ako Google a založil svoju distribuovanú súborovú architektúru HDFS na princípoch GFS (master/slave). Na nasledujúcom obrázku je zobrazená architektúra systému HDFS.

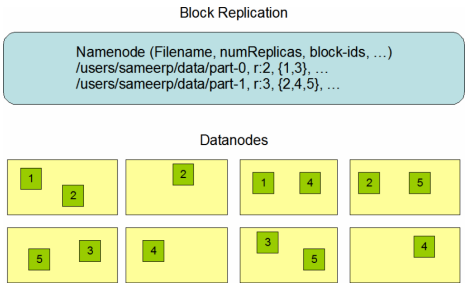
[6] http://hadoop.apache.org/core/

HDFS kluster pozostáva z jedného uzla NameNode zodpovedného za manažovanie prístupu k súborom a adresárom a za všetky blokové operácie, ako je otváranie, zatváranie a premenovanie súborov a adresárov. Uzly DataNode sú zodpovedné za čítanie a zapisovanie súborov a adresárov pre potreby súborových klientov. Aj v prípade HDFS neprechádzajú cez uzol NameNode žiadne používateľské dáta okrem metadát systému HDFS.



Obr. 3. Architektúra, dátový a riadiaci tok informácií v HDFS [2].

Najväčšou výzvou v distribuovaných súborových systémoch je replikovanie blokov dát. Na nasledujúcom obrázku je ukázaný systém replikovania v súborovom systéme HDFS (podobne je to aj v GFS).



Obr. 4. Spôsob replikovania blokov v HDFS klusteri [2].

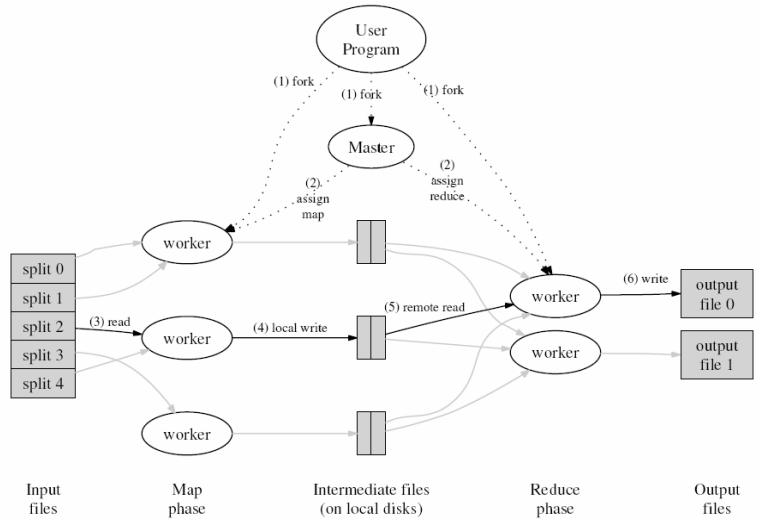
2.2 Framework MapReduce

Aby sme mohli využiť distribuovaný systém opísaný v časti 2.1, potrebujeme mať framework, ktorý bude zodpovedný za manažovanie procesov (rozdeľovanie, spúšťanie, atď.). V tejto časti si opíšeme framework MapReduce [1] navrhnutý spoločnosťou Google, ktorý umožňuje spúšťať paralelené procesy bez znalosti paraleleného programovania. Programovací model prostredia MapReduce je opísaný nižšie.

Na vstupe výpočtu je množina párov kľúč/hodnota. Vývojár musí naimplementovať dve funkcie: `Map` a `Reduce`. Funkcia `Map` dostane na vstupe pár kľúč/hodnota a vyprodukuje preň dočasný pár kľúč/hodnota. Prostredie MapReduce

zoskupí dočasné hodnoty priradené tomu istému kľúču *K* a posunie ich do funkcie `Reduce`. Funkcia `Reduce` má na vstupe dočasný kľúč *K* a množinu hodnôt k nemu priradených. Tieto hodnoty spojí dokopy, pričom je predpoklad, že množina hodnôt po ukončení funkcie `Reduce` bude menšia. Dočasné hodnoty sú poskytované do funkcie `Reduce` ako iterátor. To umožňuje zvládnuť zoznam hodnôt, ktoré sú príliš veľké na načítanie do pamäte.

Na nasledujúcom obrázku je opísaný spôsob spustenia procesu v prostredí MapReduce od spoločnosti Google.



Obr. 5. Spustenie a vykonanie procesu v prostredí MapReduce (Google) [1].

Výpočtový kluster MapReduce (v implementácií Hadoop) pozostáva z jedného uzla JobTracker (v menších klusteroch sa používa na tom istom uzle ako uzol NameNode) zodpovedného za manažovanie spúšťaných procesov. Uzly TaskTracker (uzly TaskTracker a DataNode musia byť totožné) sú zodpovedné za priame vykonávanie zverených úloh.

V nasledujúcich krokoch si popíšeme spustenie procesu v prostredí Hadoop:

1. Na uzle JobTracker sa spustí požadovaný proces, ktorý má naimplementované funkcie `Map` a `Reduce`.
2. JobTracker preskúma voľné uzly (musí na nich bežať TaskTracker) a podľa potreby (v závislosti od veľkosti vstupných dát) pridelí potrebné množstvo výpočtových uzlov (TaskTracker v závislosti od počtu jadier zvládne počítať 2 až 4 úlohy naraz). Súčasne je spustená aj úloha `Reduce` (v závislosti od množstva dát a uzlov sa môže spustiť aj viac úloh `Reduce`).
3. Po dokončení niektorej z `Map` úloh sa jej výsledky prekopírujú na niektorý z uzlov, kde beží úloha `Reduce`. Výsledky sa utriedia a čaká sa na ukončenie všetkých úloh `Map`.
4. Po dokončení všetkých úloh `Map` sa spustia úlohy `Reduce` a po ich ukončení dostaneme utriedený zoznam párov kľúč/hodnota.

V prostredí MapReduce (v implementácii Hadoop) sú aj ďalšie funkcie, ktoré môže vývojár naimplementovať. Sú to funkcie:

- `Combiner` – spustí funkciu `Reduce` hneď po úlohe `Map` na danom uzle (pomáha znížiť množstvo prenesených dát cez internet),
- `Partitioner` – používateľ prostredia MapReduce môže špecifikovať počet úloh `Reduce` spustených v jednej úlohe, pričom funkcia `Partitioner` je zodpovedná za rozdelenie dočasných kľúčov *K* na jednotlivé úlohy `Reduce`. Niekedy sú napríklad výsledné kľúče typu URL a vtedy je dobré mať výsledky z jednej URL v jednom súbore (spracované jednou úlohou `Reduce`),
- `Reporter` – je funkcia, ktorá oznamuje stav procesu a úloh.

2.3 Distribuovaná databáza

Keďže väčšina procesov spúšťaných vo frameworku MapReduce pracuje s dátami z internetu (sťahovanie stránok, logy serverov, a pod.), je žiaduce ukladať dáta do podobných štruktúr, aké nám dnes poskytujú relačné databázy. Spoločnosť Google navrhla a implementovala vlastnú databázu v mnohom podobnú relačným databázam a nazvala ju BigTable [4]. BigTable je distribuovaný ukladací priestor, ktorý slúži na ukladanie štruktúrovaných dát veľkého rozsahu (v rozsahu petabajtov) na tisícky komoditných počítačov. Podobne ako Hadoop je implementáciou frameworku MapReduce a HDFS je implementáciou GFS, aj HBase je implementáciou BigTable. HBase používa dátový model podobný tomu, ktorý používa BigTable. Aplikácie ukladajú dáta do tabuliek. Dátový riadok má triediaci kľúč (je možné podľa neho triediť zoznamy párov kľúč/hodnota) a ľubovoľný počet stĺpcov. Jednotlivé tabuľky sú väčšinou veľmi riedke, takže riadky v jednej tabuľke môžu mať rôzne počty stĺpcov. Každý stĺpec má meno v tvare: „<family>:<label>“.

Ak sa na tabuľku uloženú v systéme HBase pozrieme z konceptuálneho pohľadu, ide o kolekciu riadkov, ktoré sú vyhľadávané podľa kľúčov jednotlivých riadkov (voliteľne aj podľa časovej značky), a kde stĺpce nemusia obsahovať žiadnu hodnotu pre určité riadky (tabuľka je riedka). V nasledujúcej tabuľke je ukážka tabuľky uloženej v systéme HBase.

Tabuľka 1. Ukážka tabuľky uloženej v systéme HBase – konceptuálny pohľad [16].

Row Key	Time Stamp	Column *"contents:"*	Column *"anchor:"*		Column *"mime:"*
"com.cnn.www"	t9		"anchor:cnnsi.com"	"CNN"	
	t8		"anchor:my.look.ca"	"CNN.com"	
	t6	"<html>..."			"text/html"
	t5	"<html>..."			
	t3	"<html>..."			

Treba si uvedomiť, že v prípade fyzického uloženia v HDFS systéme prázdne miesta nie sú uložené, pretože HBase (BigTable) sú fyzicky uložené po stĺpcoch, ako to ukazuje nasledujúca tabuľka.

Tabuľka 2. Ukážka tabuľky uloženej v systéme HBase – fyzické uloženie v HDFS [16].

Row Key	Time Stamp	Column *"contents:"*
"com.cnn.www"	t6	"<html>..."
	t5	"<html>..."
	t3	"<html>..."

Row Key	Time Stamp	Column *"anchor:"*	
"com.cnn.www"	t9	"anchor:cnnsi.com"	"CNN"
	t8	"anchor:my.look.ca"	"CNN.com"

Row Key	Time Stamp	Column *"mime:"*
"com.cnn.www"	t6	"text/html"

3 Praktická ukážka

V tejto časti si ukážeme a detailnejšie rozoberieme jednoduchý program na zisťovanie výskytu slov v dokumente (WordCount[7]). Nižšie je uvedený listing (funkcií `Map` a `Reduce`) tohto programu v Jave[8].

Funkcia `Map`:

```
public void map(LongWritable key, Text value, OutputCollector<Text,
IntWritable> output, Reporter reporter) throws IOException {
  Text word = new Text();
  String line = value.toString().toLowerCase();
  StringTokenizer tokenizer = new StringTokenizer(line);
  while (tokenizer.hasMoreTokens()) {
    word.set(tokenizer.nextToken());
    output.collect(word, one);
  }
}
```

Funkcia `Reduce`:

```
public void reduce(Text key, Iterator<IntWritable> values,
OutputCollector<Text, IntWritable> output, Reporter reporter) throws
IOException {
  int sum = 0;
  while (values.hasNext())
    sum += values.next().get();
  output.collect(key, new IntWritable(sum));
}
```

Funkcia `Map` dostane časť súboru (riadok) obsahujúceho text, ktorému priradí token a pre každý token (dočasný kľúč) priradí hodnotu „one“. Funkcia `Reduce` následne dostane dočasné páry kľúč/hodnota. Pre dočasný kľúč iteruje cez všetky jeho hodnoty, pričom ich počet sumuje. Keď prebehne cez všetky hodnoty, sumu zapíše ako výslednú hodnotu pre kľúč, ktorý vlastne tvorí token (slovo).

4 Záver

V príspevku sme predstavili framework MapReduce pre vývoj a spúšťanie paralelných procesov bez znalosti paralelného programovania. Predstavený program WordCount bol len jedným z mnohých, pre ktoré je táto architektúra vlastne

[7] Originálny zdrojový kód je prebratý z: http://hadoop.apache.org/core/docs/current/mapred_tutorial.html

[8] http://java.sun.com/

vytvorená, ale aj on ukázal možnosti a rýchlosť pri spracovávaní petabajtov informácií.

Poďakovanie. Táto práca je vyvíjaná a podporovaná projektmi Commius FP7-213876, SECRICOM SEC-2007-4.2-04, SEMCO-WS APVV-0391-06, AIIA APVV-0216-07, VEGA 2/7098/27.

Literatúra

1. Dean, J., Ghemawat, S.: MapReduce: Simplified DataProcessing on Large Clusters, http://labs.google.com/papers/mapreduce.html (2008)
2. http://hadoop.apache.org/core/docs/current/hdfs_design.html (2008)
3. Ghemawat, S., Gobioff., H., Leung, S-T.: The Google File System http://labs.google.com/papers/gfs.html (2008)
4. Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes A., Gruber, R. E.: Bigtable: A Distributed Storage System for Structured Data, http://labs.google.com/papers/bigtable.html (2008)
5. Borthakur, D.: Hadoop Distributed File System, http://docs.huihoo.com/apache/hadoop/HDFSDescription.pdf (2008)
6. Borthakur, D.: Hadoop Distributed File System, http://wiki.apache.org/hadoop-data/attachments/HadoopPresentations/attachments/hdfs_dhruba.pdf (2008)
7. Bieliková, M., Návrat, P., Barla, M., Bartaloš, P., Ciglan, M., Hamar, J., Kiselkov, M., Laclavík, M., Mažgut, J., Máté, J., Suchal, J., Šeleng, M., Tvarožek, M., Vojtek, P.: Štúdie vybraných tém softvérového inžinierstva (3) : pokročilé metódy navrhovania programových systémov : pokročilé metódy získavania, vyhľadávania, reprezentácie a prezentácie informácie. Bratislava : Slovenská technická univerzita v Bratislave, 2007. Edícia výskumných textov informatiky a informačných technológií, 3. ISBN 978-80-227-2701-3.
8. Lucene-hadoop Wiki, HadoopMapReduce, http://wiki.apache.org/lucene-hadoop/HadoopMapReduce (2008)
9. The Phoenix system for MapReduce programming, http://csl.stanford. edu/~christos/sw/phoenix/. (2008)
10. Open Source Distributed Computing: Yahoo's Hadoop Support, Developer Network blog, http://developer.yahoo.net/blog/archives/2007/07/yahoo-hadoop.html, (2007)
11. Yahoo! Launches World's Largest Hadoop Production Application, Yahoo! Developer Network, http://developer.yahoo.com/blogs/hadoop/2008/02/yahoo-worlds-largest-production-hadoop.html, (2008)
12. Parallel Virtual Machine (PVM), http://www.csm.ornl.gov/pvm/pvm_home.html (2008)
13. Message Passing Interface Forum (MPI), http://www.mpi-forum.org/ (2008)
14. Condor - High Throughput Computing (HTC), http://www.cs.wisc.edu/condor/ (2008)
15. LCG Deployment, http://lcg.web.cern.ch/LCG/activities/deployment.html (2008)
16. Hbase wiki page, http://wiki.apache.org/hadoop/Hbase/HbaseArchitecture (2008)
17. Disco page, http://discoproject.org/ (2008)

Integrácia a zobrazenie výstupov modulov spracovávajúcich emailové správy

Emil Gatial, Michal Laclavík, Ladislav Hluchý

Ústav informatiky
Slovenská akadémia vied
Dúbravská cesta 9, 845 07 Bratislava
emil.gatial@savba.sk

Abstrakt. Tento článok sa zaoberá integráciou výstupov z rôznych modulov, ktoré automatizovane spracovávajú a extrahujú informácie z emailových správ tak, aby výsledná emailová správa bola obohatená o relevantné informácie. V emailovej správe sú tieto informácie priložené k textu originálnej správy v textovom alebo HTML formáte. V prípade aktívnych komponentov sú priložené hypertextové odkazy na interaktívne webové rozhranie. Hlavná úloha navrhovaného prístupu je poskytnúť konfigurovateľné a intuitívne používateľské rozhranie zobrazujúce výstupy zo všetkých modulov extrahujúcich informácie z emailových správ. V úvode je rozobraná problematika integrácie výstupov z modulov. Ďalej jednotlivé podkapitoly rozoberajú návrh architektúry a technologicé možnosti implementácie. Záver je venovaný ďalším plánom pri návrhu a implementácii. Táto práca vznikla v rámci projektu COMMIUS.

1 Úvod

Tento článok sa zaoberá integráciou výstupov z rôznych modulov, ktoré automatizovane spracovávajú a extrahujú informácie z emailových správ tak, aby výsledná emailová správa bola obohatená o relevantné informácie. V emailovej správe sú tieto informácie priložené k textu originálnej správy a v prípade aktívnych komponentov sa tu nachádzajú hypertextové odkazy na webové rozhranie.

V súčasnosti je mnoho webových emailových klientov založených na AJAX[1] metodológii ako napríklad Google Mail[2], Zimbra[3], 6Zap[4], Foldera[5] and BlueTie[6]. Tieto webové aplikácie sú určené na bežnú emailovú komunikáciu spojenú s manažmentom emailových správ a dokumentov. Spomínané webové aplikácie využívajú možnosti webového prehliadača komunikovať asynchrónne na pozadí webovej aplikácie. Pre účely jednoduchšieho a transparentnejšieho vývoja takých

[1] Základné informácie o metodológii AJAX, http://en.wikipedia.org/wiki/AJAX
[2] Gmail, http://mail.google.com
[3] Zimbra, http://www.zimbra.com/products/collaboration.html
[4] 6Zap, http://www.6zap.com/product/
[5] Foldera, http://www.techcrunch.com/2006/02/20/foldera-never-organize-your-inbox-again/
[6] BlueTie ,http://www.bluetie.com/

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 35–39, ISBN 978-80-227-3027-3.

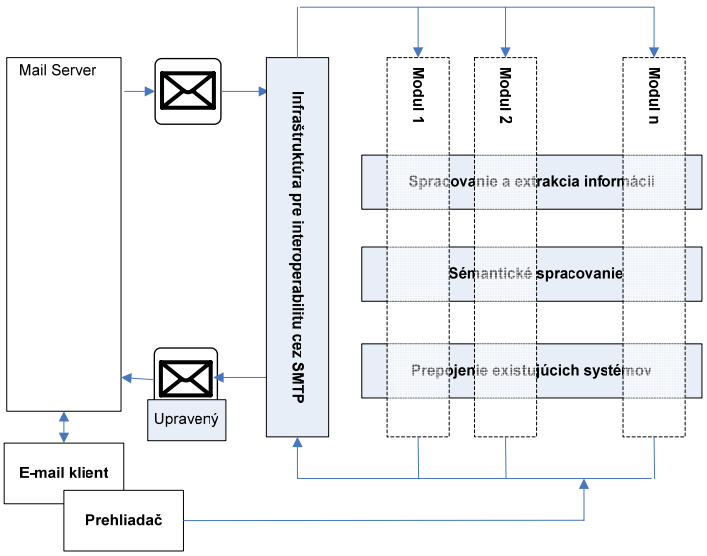
aplikácii bol vyvinutý nástroj GWT[7] [2], ktorý umožňuje prekompilovať aplikáciu vytvorenú a odladenú v Java jazyku do JavaScript jazyka umožňujúceho bežať rovnako dobre na webových prehliadačoch.

Hlavná úloha nami navrhovaného prístupu je poskytnúť konfigurovateľné a intuitívne používateľské rozhranie zobrazujúce výstupy zo všetkých modulov extrahujúcich informácie z emailových správ. To znamená, že používateľ bude môcť automatizovať činnosti, ktoré budú aktivované na základe novej emailovej správy pomocou zásuvných modulov.

Architektúra systému COMMIUS[8] je načrtnutá na obrázku 1, kde nová emailová správa prechádza rôznymi modulmi zloženými s komponentov:

- Spracovanie a extrakcia informácií (systémová úroveň)
- Sémantické spracovanie (sémantická úroveň)
- Prepojenie existujúcich systémov (procesná úroveň)

Jednotlivé komponenty sú navzájom poprepájané do logického bloku zvaného modul tak, aby výstup z neho prezentoval relevantné informácie získané na základe textu emailu. V rámci takéhoto systému môže súbežne pracovať viacero modulov, ktorých výstupy majú byť integrované do prílohy emailovej správy a navyše majú byť prístupné aj pomocou interaktívneho webového rozhrania. Webové rozhranie má zároveň slúžiť aj na konfigurovanie modulov.



Obr. 1. Architektúra systému na spracovanie emailových správ.

Prichádzajúca emailová správa je najskôr spracovaná jednotlivými modulmi, ktoré z nej vyextrahujú relevantné informácie. Tieto informácie sú rozložené a uložené do dočasnej pamäte, kde sú uložené a pripravené na zobrazenie. Jeden typ zobrazenia je

[7] Google Web Toolkit, http://code.google.com/webtoolkit/
[8] COMMIUS projekt, http://www.commius.eu/

statický text, ktorý je pripojený k originálnej emailovej správe. Tento popis môže obsahovať aj odkaz na webové rozhranie, ktoré dokáže informácie zobraziť aj pomocou dynamických komponentov.

COMMIUS systém počíta s dvomi scenármi použitia:

- Serverové riešenie. Na spracovanie emailových správ je použitý centrálny emailový server, ktorý spravuje všetku emailovú komunikáciu pre každý emailový účet. Tu je nutné zabezpečiť dôveryhodnosť a bezpečnosť algoritmov extrahujúcich informácie.
- Lokálne riešenie. Na spracovanie emailových správ je použitá inštalácia, ktorá je oddelená od emailového serveru a správy sú spracovávané lokálne. V tomto scenári nie je nutné riešiť dôveryhodnosť ani bezpečnosť algoritmov (za predpokladu, že počítač je zabezpečený voči útokom a vírom). Nevýhodou však je to, že webové rozhranie je prístupné len z lokálneho počítača.

2 Architektúra spracovania správ

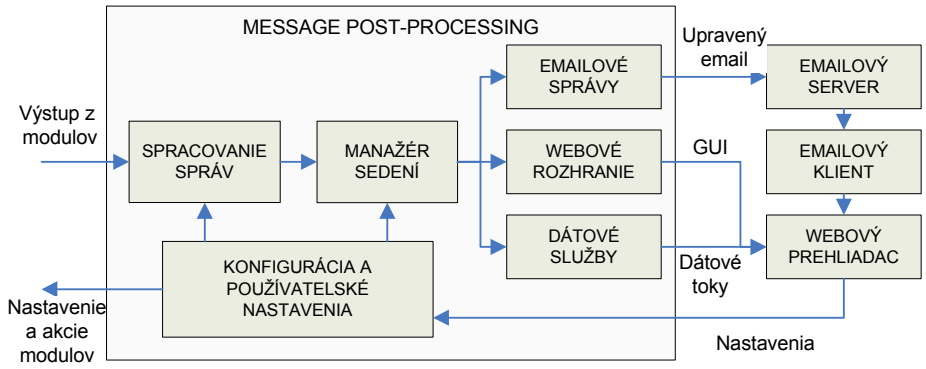
Doterajší návrh architektúry bol zameraný na zobrazenie jednoduchých správ obohatených o informácie, ktoré boli nájdené pri analýze emailovej správy modulom ACoMa [1]. Súčasný návrh architektúry MPP[9] komponentu je postavený na nasledujúcich komponentoch:

- Spracovanie správ. Tento komponent spracúva správy v rôznych formátoch (xml, eml, atď.) a extrahuje informácie a meta-informácie použité pri zobrazení.
- Manažér sedení. Integruje údaje na základe vytvoreného sedenia a poskytuje dočasnú pamäť.
- Spracovanie emailových správ. Na základe konfigurácie systému sa k originálnej emailovej správe pripoja výstupy z modulov v textovom alebo HTML formáte. Takto modifikovaný email sa pošle prijímateľovi emailu.
- Spracovanie webového rozhrania. Tento komponent má za úlohu vytvoriť webovú stránku, kde budú informácie integrované do interaktívneho rozhrania. Tento komponent vygeneruje počiatočnú stránku pomocou Javascript komponentov.
- Dátové služby. Tento komponent len registruje služby v rámci aplikačného portálu (služba v tomto kontexte predstavuje servlet prijímajúci akcie a jemu zodpovedajúca implementácia dynamických komponentov v jazyku Javascript).
- Používateľské nastavenia a konfigurácia modulov. Umožňuje modifikovať a ukladať nastavenia modulov ako aj nastavenia používateľského rozhrania.

3 Technologická platforma

Z dôvodu overenia navrhnutej metódy bola implementovaná demonštračná verzia MPP komponentu. Pri jej realizácii bol kladený dôraz na overenie architektúry a použiteľnosti vybratých implementačných technológií a nástrojov.

[9] Message Post-Processing komponent je zodpovedný za spracovanie a integráciu výstupov z COMMIUS modulov.



Obr. 2. Bloková schéma MPP komponentu.

Keďže sa jedná hlavne o prezentáciu výstupov, pri realizácii bol zvolený nástroj GWT [2], ktorý umožňuje vývoj a ladenie webových aplikácií založených metodológii AJAX v jazyku Java a následné prekompilovanie kódu do jazyka Javascript bežiaci na webových prehliadačoch. GWT nástroj bol rozšírený o sofistikovanejšie komponenty ExtGWT[10]. Bol zvolený aplikačný server Jetty[11], ktorý slúži ako kontajner pre webové aplikácie a navyše je veľmi dobre použiteľný pre lokálne riešenie.

Komponenty vytvárajúce používateľské rozhranie sú spúšťané v rámci Jetty aplikačného serveru, kde komponent na správu sedení je implementovaný ako servlet, ktorým sa inicializuje generovanie používateľského rozhrania a komponent na dátové služby je registrovaný ako GWT služby. Oba komponenty používajú ten istý kontajner sedení.

4 MPP komponent

Komponent integruje informácie extrahované z emailovej správy. Implementovaná verzia zatiaľ implementuje len grafické rozhranie. Textová časť emailovej správy je obohatená tak, že kľúčové slová ku ktorým boli nájdené relevantné poznámky sú zvýraznené symbolom, ktorý reprezentuje význam poznámky. Tieto následne umožnia vyvolanie akcie. Používateľské rozhranie zároveň umožňuje úpravu konfiguračných súborov a spúšťanie nástroja ACoMa.

V súčasnosti boli práce na MPP komponente zamerané na overenie použiteľnosti komponentov GWT a ExtGWT. Plány na ďalšiu implementáciu zahŕňajú definovanie spoločného formátu výstupných správ z modulov, ktorý bude založený na XML formáte a špecifikovať spôsob komunikácie medzi modulmi a MPP komponentom a overiť navrhnuté riešenie na jednoduchom príklade.

[10] Domovská stránka ExtGWT, http://extjs.com/products/gxt/
[11] Jetty aplikačný server, http://www.mortbay.org/jetty/

5 Záver

Síce komponent na spracovanie výstupov je len vo fáze návrhu a implementácie demonštračnej aplikácie, tento komponent prešiel overovaním technologickej platformy GWT a návrhu zásuvných modulov vizuálnych aplikácií pre jednotlivé moduly. Tento článok sa snaží opísať doterajšie výsledky pri návrhu a implementácií webovej aplikácie založenej na AJAX metodológii určenej na spracovanie emailových správ.

Poďakovanie

Táto práca bola podporená SECRICOM SEC-2007-4.2-04, FP7-ADMIRE FP7-215024, AIIA APVV-0216-07, Commius FP7-213876, SEMCO-WS APVV-0391-06, VEGA 2/7098/27.

Literatúra

1. Laclavík, M., Šeleng, M., Hluchý, L.: ACoMA: Network enterprise interoperability and collaboration using E-mail communication. In Expanding the knowledge economy: Issues, applications, case studies. - Amsterdam : IOS Press, 2007. ISBN 978-1-58603-801-4. ISSN 1574-1230, part 2, S. 1078-1084.
2. Hanson, R., Tacy, A.: GWT in Action, Easy Ajax with the Google Web Toolkit, Manning Publiccations Co., June, 2007, 632 pages, ISBN: 1-933988-23-1.

Marketing Segmentation of Bank Retail Portfolio

Jozef Kováč

Faculty of Electrical Engineering and Informatics
Technical university in Košice, Letná 9, Košice
Adastra, s.r.o., Francisciho 4, Bratislava
jozo.kovac@gmail.com

Abstract. Segmentation of client portfolio is a powerful customer intelligence tool for better understanding of companies' client base. There are many different approaches for creation of segmentation. But what segmentation is really good and how to develop it? It is important to keep in mind not only mathematical aspects of segmentation, but primary business context and adjust used methodology and whole creation process to satisfy the business user's needs. In this article there are discussed practical issues what appeared in retail portfolio segmentation project for international bank in Slovakia.

Keywords: segmentation, marketing, retail, CRISP DM, k-means

1 Introduction

Banks usually use several segmentations of their client portfolio. The most usual are segmentations by financial data (e.g. balance up to 1 mil., up to 5 mil., more than 5 mil.), by market segment (private individuals, VSE/SME, corporate, municipality), by risk rating. These basic segmentations are useful for a majority of bank's internal departments despite of fact that each department has different view over client portfolio, uses segmentations for different purposes and has different key performance indicators (KPI) for evaluation of own success.

Particular departments are motivated and trying to better know clients relevant for them, because they already realized it is way for better KPI values. One of approaches for achieving a goal is to use tailored portfolio segmentation, identify important segments and focus on them. Literature cites several algorithms and approaches suitable for solving the segmentation problem. From experience the successful segmentation solution may require more that just adoption of right algorithm. Even more important parts are deep domain knowledge, ability to listen to business user needs, data manipulation and knowledge extraction skills.

2 Methodology

CRISP DM methodology is an industry standard for development of data-mining models. It is sufficiently general methodology covering development of predictive,

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 40–44, ISBN 978-80-227-3027-3.

descriptive or segmentation models [1]. Authors [2, 3, 4] suggest also different methodologies using whose may lead to comparable results. CRISP DM consists of six basic phases:

1. Business understanding
2. Data understanding
3. Data preparation
4. Modeling
5. Evaluation
6. Deployment

These phases are closely related and it is essential to fulfill all requirements from a phase before passing to following one.

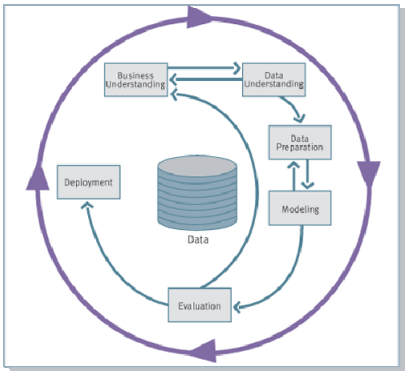


Fig. 1. CRISP DM

2.1 Business understanding

In methodology this initial phase focuses on understanding the project objectives and requirements from a business perspective, then on converting this knowledge into a data mining problem definition, and finally on a preliminary plan designed to achieve the objectives [1].

Marketing department needs segmentation for better understanding of their client portfolio and wants to use gathered knowledge for improvement of own marketing activities whose include targeting of direct mail campaigns, selection and monitoring of strategic segments, product development, service levels management.

It is hard to convert these broadly specified business requirements into definition of single data-mining problem. There is no target variable, no explicit function to validate quality of model. Success heavily depends on ability of analyst to find out right segmentation rules and persuade business users to adopt them and use in daily work. What other, primary technical, parameters should that segmentation have?

1. Reasonable number of well described and sustainable segments
2. Be based on data available in data-warehouse so clients could be scored periodically
3. Segments stabile over time (not changing too quickly)

2.2 Data understanding

There are always two basic kinds of data to be considered for segmentation project:

- "Hard" data – data periodically loaded from primary systems into data-warehouse what are provided by ETL process. Advantage of choosing these data is that because they are periodically updated, clients can be also periodically scored using segmentation rules based on these data. And even at no additional costs. Disadvantage of this data is limited information value about portfolio and so a lot of knowledge stays hidden. Segmentation project requires several monthly snapshots to evaluate stability of segmentation rules.
- "Soft" data – data about client's life style, personal preferences, life situation, relationship with competitors, etc. These data are not commonly stored and updated in data-warehouse. They are gathered using surveys on the sample of portfolio. So these data are expensive and may be biased. Not all segments respond to surveys equally, not all clients answer questions responsible and then gathered results have to be taken with some reserve.

2.3 Data preparation

Hard data driven segmentation requires client level data mart with periodical client profile snapshots. Initial segmentation requires a snapshot with complete behavioral client profile. Behavioral client profile consist of client accounts data, balances, client behavior – how he uses products, delinquencies, seniority, and some geographic and demographics attributes are included too. It is also possible to use different trends and summary multi month variables and gain better segmentation rules over time stability. Generally data mart could contain few hundred attributes.

Another part is the data transformations. Algorithms for segmentation require data in right form. Distance based algorithms work the best with data standardized to interval <0,1> or if data are categorical then common transformation is a binary encoding.

Standardization of numeric attributes slightly complicates evaluation of segmentation model, because description of segments has to use real values, evident for domain experts.

2.4 Modeling

There is long list of segmentation algorithms to choose from. Including distance based algorithms, probability based, hierarchical or Kohonen networks. For example let's try to use k-means distance based algorithm. There are a lot of questions to consider even before the first iteration.

There is no one simple and correct answer for these questions and every decision has huge impact on performance of the algorithm and on the final results. Creation of segmentation on large portfolio with lot of attributes and different parameters leads to random searching in nearly infinite state space.

To deliver segmentation in time and in requested quality we propose following approach. First gather and summarize business user's expectation during interview with them. It is commonly expected to find some well known segments in data. The first task is to find these popular segments. Then to explore the rest of portfolio and identify, name and describe all other important segments.

Using this approach segmentation is born on the paper first. Design should include number of segments, basic description of each segment and initial rules for segments definition. Design should meet business user's expectation, be extract of expert's experience, and possibly include some frontier ideas.

Second step is to check, if design suits with the data. This is iteration process and request some time – distributions of segments has to be computed, then some segments definition need to be revised, some segments are usually not found and some new will appear. Whole segmentation is set of if – then – else rules what can be expressed in SQL or nearly every other computer language. Analyst has great control over these rules. Has to check how his manual changes in rules reflect into distributions of segments. If closer detail is needed, hierarchical segmentation with more sub-segments can be created.

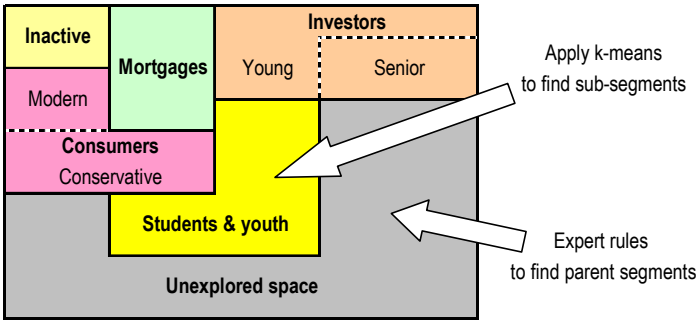


Fig. 2. Segmentation of retail portfolio.

There is a place for application of standard segmentation algorithms. When parent segment is already defined, there are fewer options for definition of sub-segments: smaller data width and height, and usually small number of sub-segments to be found. K-means algorithm has proved to be very useful here. It provides proposals for sub-segments definitions and automatically finds the splitting points for numerical values. These points are hard to find manually and k-means find them easily in a short time.

2.5 Evaluation

Segmentation model is evaluated with set of reports. These reports have to be periodically updated so customer could watch the segmentation changes over time. Set of reports includes:

– Segment distribution in current month

- Summary and average client profile over segments (balances, products penetration, debt, delinquencies, client activity, transaction, etc.)
- Segment saturation
- Segment trends over past 2 years
- Transitions matrixes with highlighted common paths between segments

Most of issues with stability of segmentation have same solution – use correct basic data for segmentation rules. It is not important to transform data into the best possible form. It is better to return back to data preparation stage and reload correct data from data-warehouse.

2.6 Deployment

Deployment is divided into two parts: technical and business.

Technical part of deployment consists of automation of periodical scoring process into data-warehouse. Segmentation rules are written into SQL procedures and provide actual information about segment for every client every month. Information about segment also appears in reporting system of department and is provided to CRM system to points-of-sales (POS).

Business part is a job for business users. In case of marketing segmentation it is marketing analysts who prepares special campaigns for individual segments, chooses strategic segments, make actions which increase their size in portfolio and periodically reports own progress. Another important user is product design department what creates customized products for important segments.

3 Conclusion

Experts say that the best segmentation is one what suits the business needs the bests. In real world it is hard to describe these needs in metrics and math, so standard approach may lead to success in very hard and long way. Abilities to listen, to understand problem details from both business and data side, and to deliver a complex solution, not just mathematics or statistics output are very important for the final success. Usually simpler, logical, understandable and sustainable solution is what business users want and will use daily.

References

1. CRISP DM homepage, http://www.crisp-dm.org/index.htm
2. Michel J. A. Berry, Gordon S. Linoff, Data-mining techniques, 2004, Willey publishing, ISBN: 0-471-47064-3
3. Milan Meloun, Jiří Militký, Martin Hill: Počítačová analýza více rozmerných dát v príkladech, 2005, Academia, Praha, ISBN: 80-200-1335-0
4. Olivia Parr Rud, Data Mining Cookbook, Juhn Wiley & Sons, ISBN 0-471-38564-6

Zvýšenie robustnosti relačnej klasifikácie v slabo homofílnych dátach

Peter Vojtek, Mária Bieliková

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií
Slovenská technická univerzita
Ilkovičova 3, 842 16 Bratislava, Slovakia
{pvojtek,bielik}@fiit.stuba.sk

Abstrakt Relačná klasifikácia obohacuje atribútovo-viazané zatrieďovanie o informácie plynúce zo vzťahov medzi objektami. Táto práca sa venuje relačným klasifikátorom, ktoré využívajú šírenie informácií pomocou viacerých typov vzťahov medzi rôznorodými objektmi v grafoch. Na základe analýzy kvality šírených informácií je navrhnutý a experimentálne overený mechanizmus na obmedzenie výmeny nevhodných informácií, ktorý sa aplikuje keď nie je splnený predpoklad homofílie susedných objektov, čo prispieva ku zvýšeniu presnosti klasifikácie.

1 Úvod

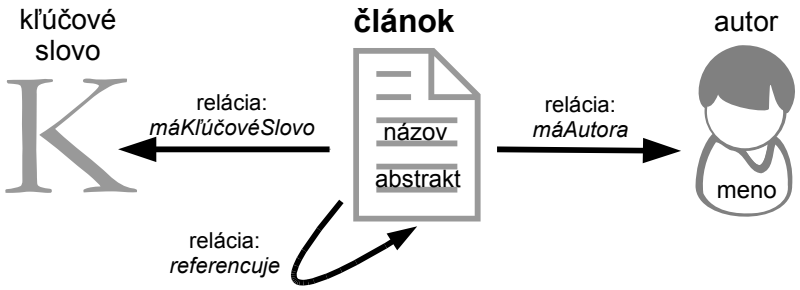
Klasifikácia, jedna z tradičných techník dolovania v dátach, využíva atribúty (tiež *vlastnosti, črty*) zatrieďovaných inštancií (objektov) ako vstupné parametre, na základe ktorých sa klasifikátor naučí rozlišovať daný vzor. Tento prístup používajú známe metódy ako rozhodovacie stromy, či SVM (Support Vector Machines) [1] a hovoríme o atribútovo-viazanom prístupe.

Odlišný prístup ku klasifikácii predstavuje relačný pohľad [2], kedy sú pre klasifikátor ústredné vzťahy medzi inštanciami (pričom sa však nevylučuje ani využitie znalostí z vlastných atribútov). Relačná klasifikácia je výhodná a dosahuje nadštandardné výsledky v oblastiach, kde sú objekty medzi sebou výrazne prepojené vzťahmi, vytvárajúcimi grafovú štruktúru. Príkladom je oblasť klasifikácie vedeckých článkov, kde článok má atribúty ako názov a abstrakt, prepojenia na iné články (cez referencie), prepojenia na osoby (autorov), či kľúčové slová. Obr. 1 zobrazuje túto doménu, pričom klasifikačnou úlohou je napr. rozhodnúť, ktoré články spadajú do triedy *Hardvér*.

Iteratívna výmena informácií v grafe

Novšie a pokročilejšie relačné klasifikátory využívajú mechanizmus spoločného odvodzovania (*collective inference*), ktorý umožňuje iteratívne šírenie informácií v grafe objektov [3]. Všeobecná schéma je takáto:

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 45–48, ISBN 978-80-227-3027-3.



Obrázok 1. Graf z domény vedeckých článkov.

1. každému objektu sa na základe jeho vlastných atribútov nastaví príslušnosť ku triedam (ďalej sa nazýva *stav*) pomocou niektorého atribútovo-viazaného klasifikátora.
2. Každý objekt z testovacej množiny si postupne v iteračných krokoch obnovuje svoj stav na základe stavov susedných objektov (susednosť = priame prepojenie reláciou medzi objektami).
3. Mechanizmus aktualizácie stavu objektu je špecifický pre zvolenú klasifikačnú metódu, zvyčajne sa zohľadňuje vlastný stav z predošlého kroku a stavy susedných objektov majú rôznu váh podľa toho, či sú z trénovacej alebo testovacej množiny.
4. Spôsob aktualizácie stavu je navrhnutý tak, aby stavy objektov konvergovali. Po ustálení stavov v grafe sa každému klasifikovanému objektu priradení tá trieda, ktorá získala najvyššie ohodnotenie (t.j. mala najvyššiu hodnotu v stavovom vektore).

Princíp spoločného odvodzovania do veľkej miery pripomína stavové automaty, s tým rozdielom, že susedstvo zvyčajne nie je pravidelné a stav je vektor reálnych čísel.

Obmedzovanie výmeny informácií

Všeobecný postup popísaný v predošlej časti vychádza z tzv. predpokladu homofílie:

> Susedné, resp. blízke objekty v grafe prislúchajú k rovnakej triede s väčšou pravdepodobnosťou ako objekty, ktoré sú si v grafe vzdialené.

Uvedený predpoklad však mnoho dátových množín spĺňa len v slabej miere a v prípade, že nie je vopred známy charakter vstupných dát, klasifikátor nie je dostatočne robustný a nie je zaručené, že výsledok zatrieďovania bude na očakávanej úrovni. Tento nedostatok sa dá ošetriť zavedením moderovania (obmedzovania) výmeny informácií v grafe tak, aby k výmene informácií dochádzalo len keď je podmienka homofílie splnená v zvolenej intenzite.

V súčasnosti nám žiaden známy relačný klasifikátor nemá podporu pre moderované šírenie informácií – stavy sa medzi objektmi vymieňajú nech je ich obsah akýkoľvek. Nami navrhnutý spôsob využíva predpoklad, že objekty ktoré výraznejšie prislúchajú k určitej triede (teda ich stavový vektor preferuje jednu triedu) prinášajú zmysluplnejšie informácie než ostatné objekty. Ohodnotenie stavu je možné pomocou entropie nasledovným spôsobom:

$$H(n) = -\sum_{i=1}^{z} p(c_i|n) \log p(c_i|n) \tag{1}$$

kde n je aktuálny objekt, z je počet tried klasifikácie a c_i sú príslušnosti ku jednotlivých triedam. Napr. objekt so stavovým vektorom $[c_1 = 0.5, c_2 = 0.5]$ je ohodnotený ako nevhodný na zdieľanie svojho stavu, keďže neposkytuje žiadnu dodatočnú informáciu o tom, ku ktorej triede patrí a naopak, objekt so stavom napr. $[c_1 = 0.8, c_2 = 0.2]$ je zrejme vhodné využiť pri šírení informácií v grafe, keďže jeho príslušnosť ku triede c_1 je výrazná. Voľba prahu, oddeľujúceho *užitočné* a ostatné stavy, závisí od konkrétnych dát – v ďalšej časti uvádzame experimentálne zistenie najvhodnejšej hranice v dátovej vzorke vedeckých publikácií.

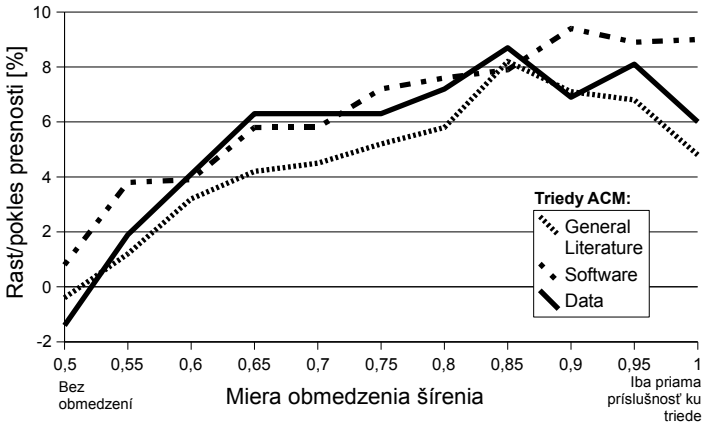
2 Experimentálne overenie

V experimente sme klasifikovali vedecké články z dátovej množiny MAPEKUS[1] s využitým vybraných tried klasifikácie ACM[2]. Klasifikovaný graf má rovnakú štruktúru ako príklad na Obr. 1, počet objektov je vyše 20 000. Ako relačný klasifikátor sme zvolili IRC (Iterative Reinforcement Classifier) [4]. Postupne sme menili parameter moderácie a sledovali rast/pokles metriky *presnosť* (accuracy) v porovnaní s atribútovo-viazaným klasifikátorom Naive Bayes. Výsledky sú na Obr. 2. Z uvedených výsledkov vyplýva, že bez obmedzenia šírenia môže relačný klasifikátor dokonca znížiť presnosť klasifikácie. S rastúcim stupňom moderácie sa však presnosť klasifikácie zvyšuje, pričom nárast presnosti je až 9.6%. Pri najvyššej možnej miere obmedzenia, kedy sa grafom šíria len vektory s najnižšou možnou entropiou (napr. $[c_1 = 1.0, c_2 = 0.0]$), však už dochádza k miernemu poklesu výkonnosti klasifikátora, keďže grafom sa šíria výhradne stavy trénovacích objektov.

3 Zhodnotenie a ďalšia práca

Pre každý klasifikátor platí, že výber konkrétnej metódy klasifikácie dát by mal závisieť od povahy týchto dát, keďže neexistuje najlepší všeobecný klasifikátor [5]. Relačnú klasifikáciu je vhodné použiť v doménach, kde sú silno prítomné vzťahy medzi inštanciami. V tejto práci sme zaviedli a experimentálne overili

[1] `http://mapekus.fiit.stuba.sk/`
[2] The Association for Computing Machinery: `http://www.acm.org/class/`



Obrázok 2. Vplyv sily obmedzenia výmeny informácií na kvalitu klasifikácie.

moderáciu šírenia informácií v klasifikovanom grafe objektov, ktorá prispieva k zvýšeniu robustnosti a presnosti klasifikácie. Tento prístup je použiteľný pre všetky relačné grafovo-viazané klasifikátory so spoločným odvodzovaním a zároveň zachováva ich konvergentnosť a neovplyvňuje rýchlosť klasifikácie. V ďalšej práci sa chceme venovať vedľajším pozitívnym účinkom relačnej klasifikácie, ktorým je odvodenie informácií aj pre objekty, ktoré sú v grafe prítomné, ale nie sú priamo klasifikované (napr. autori publikácií). Týmto spôsobom je možné prispieť napr. k vytvoreniu presnejšieho modelu používateľa.

Táto práca bola čiastočne podporená zo zdrojov VEGA VG1/0508/09 a SEMCO APVV-0391-06.

Referencie

1. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In Nédellec, C., Rouveirol, C., eds.: Proceedings of ECML-98, 10th European Conference on Machine Learning. Number 1398, Chemnitz, DE, Springer Verlag, Heidelberg, DE (1998) 137–142
2. Macskassy, S.A., Provost, F.: Classification in networked data: A toolkit and a univariate case study. J. Mach. Learn. Res. **8** (2007) 935–983
3. Jensen, D., Neville, J., Gallagher, B.: Why collective inference improves relational classification. In: KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM Press (2004) 593–598
4. Xue, G., Yu, Y., Shen, D., Yang, Q., Zeng, H., Chen, Z.: Reinforcing web-object categorization through interrelationships. Data Min. Knowl. Discov. **12**(2-3) (2006) 229–248
5. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation **1**(1) (April 1997) 67–82

Towards Automatic Workflow Composition and Execution

Tomáš Drenčák, Marek Paralič

Faculty of Electrical Engineering and Informatics
Technical University of Košice, Letná 9
042 00 Košice, Slovakia
{Tomas.Drencak,Marek.Paralic}@tuke.sk

Abstract. The automatic composition and execution of workflows composed of web and grid services is nowadays a challenging task that is hardly achievable. One of the popular solutions is to utilize semantic annotations to the services and apply them during the composition phase. However, most of the efforts end up with a semi-automatic solution that needs human participation. Our approach briefly described in the paper is based on utilization not only the semantic annotations of services, but also the produced results in previous executions of workflows and their meta-information in form of QoS parameters. This information helps us to minimize the human participation that is left only from the reasons of utilization of domain experts directly and better workflow control during the execution phase[1].

Keywords: workflow composition, semantic services, web service, grid service

1 Introduction

The problem of automatic composition of a workflow of computing tasks is very attractive especially in software engineering applied to scientific research, where complicated simulations and parameter studies often require tens of single steps in order to obtain the solution desired by the scientist ([1, 2, 3]). However, most of the many works on this topic have concerned themselves only with the composition of a workflow of computer processes – represented by grid jobs, calls to web service interfaces, or other tasks – solving the "how", and have omitted the "what" of this problem, in this case "what" being the data, on which these processes operate. This has been left to the user.

The work described here is part of the SEMCO-WS system that is trying to expand and refine several components of K-Wf Grid [3] (adding also other features, not present in K-Wf Grid). In this paper we focus on construction and execution of workflows composed from web and grid services.

[1] The work described in this paper is supported by the SEMCO-WS APVV 0391 06 project and project Nr. 1/3135/06 of the Slovak Grant Agency.

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 49–52, ISBN 978-80-227-3027-3.

2 Workflow Composition and Execution Module Architecture

In K-Wf Grid system Grid Workflow Execution Service (GWES) coordinates the workflow creation and delegate parts of the composition and refinement process to the external system components Workflow Composition Tool (WCT) and Automatic Application Builder (AAB). The basis of the Grid workflow execution is the Grid Workflow Description Language (GWorkflowDL) which uses the semantic of High Level Petri Nets to describe the logic, data and state of distributed workflows. The WCT is able to semi-automatically build abstract workflows which deliver specific user-requested data using available grid and web services which have been described by means of ontologies. The AAB maps abstract workflows onto web service candidates taking into account user policies and technical constraints. The GWES is also responsible for the workflow execution – it analyses the workflow description and gathers all activated transitions which trigger a workflow refinement or the invocation of an activity. The GWES delegates the selection of resources to an external Scheduler that uses monitoring information provided by the Performance Monitoring and Analysis Services [4].

In SEMCO-WS system we use a modified architecture, whereby the basic components are:

- Abstract Workflow Composition Tool (AWCT),
- Executable Workflow Composition Tool (EWCT) and
- Workflow Execution Engine (WEE).

As far as we preserve the Petri net model for the workflow description, workflow composition and execution module operates on a representation consisting from transitions and places. Transitions correspond to activities, which consumes input data in form of data tokens from input places and produce output data in form of tokens placed into output places. The Abstract Workflow Composition Tool is responsible for workflow construction and uses backtracking from the final activity to the initial activities of the workflow. It also includes backtracking from the final token to the initial tokens of the workflow. We can abstract tokens and activities as data providers, with the difference that tokens provide data and require no inputs, and activities also provide data, but require input data – other tokens.

Extended process of workflow refinement that changes an abstract workflow into the executable one is offered by the Executable Workflow Composition Tool. Workflow refinement architecture consists of two components: Workflow refiner and Transition refiner. These are implemented in three implementations which uses each other for specific tasks.

At first Service candidates refiner tries to find appropriate services for the transition. Transition is translated into OWL-S profile by Transition translator. This profile is then passed to the *OwlsProfileMatcher*. Returned services are then pushed into the transition as the list of service candidates.

Next *Precondition/Effect workflow refiner* removes candidates which does not fulfill preconditions and effects described in OWL-S process of the web service of the candidate. Preconditions and effects are usually described in ontology language (such as SWRL), so reasoning can be performed.

QoSRefinement is used to determine the best service among service candidates which were left after previous refinements. QoS refinement orders service candidates according to some static (e.g. precision/recall of the classification model, computer performance, service reliability) or dynamic (e.g. document statistics) indicators and chooses the best one.

Service matching is closely tight to the ontology repository. It uses it's searching capabilities by SPARQL query language and reasoning over the search result. Service matching is done in two steps: Parameter type matching and Parameter type reasoning. In Parameter type matching step the *Input/Output profile matcher* component is used. It creates SPARQL query where each inputs' and outs' parameter type is taken into account.

Parameter type step reasoning exists due to difficulty to create SPARQL query which will reason parameter type based on class inheritance as OWL-S Parameter class's parameterType range is defined as AnyURI. *Input/Output reasoning matcher* creates SPARQL which will get all OWL-S services in the repository. Services which were got in the first step (I/O profile matcher) are removed from the reasoning. Others are tested by reasoning input/output to be subclass of the wanted service profile's input/output.

At the end matching services from the first and second step are merged into one collection and returned.

Workflow Execution Engine is the place where workflow is running in. Environment works on Petri-net principle of passing tokens. Whenever one transition has got tokens from all inputs, Workflow environment executes this transition by *Transition executor*. Transition executor manages passing of the tokens from input places towards output places (*PassTokensExecutor*). When executors finds out that the current transition is web service call, rather than just passing tokens transition, it also executes the transition by Web service transition executor. Instead of WEE also the Grid Workflow Execution Service from the K-Wf Grid project could be used directly.

3 Use-case text mining workflow

In this workflow text mining services are linked together. There are 2 branches – train documents and test documents branch (see Fig. 1). Each branch starts with parsing document collection into appropriate format. These parsed documents are then subjected to tokening and token filtering. Tokens are then indexed. The same index must be used for both branches. Therefore indexer in the train branch must wait for the indexer in the test branch. After indexation, document-term matrix is created. At first sequence of numbers produced by indexing service are transformed into document-term matrix with term frequency. This matrix is then passed to the TF-IDF weighting service. Finally classification model is built (e.g. perceptron) in the train branch and used in test branch for document classifications which are the results of the workflow.

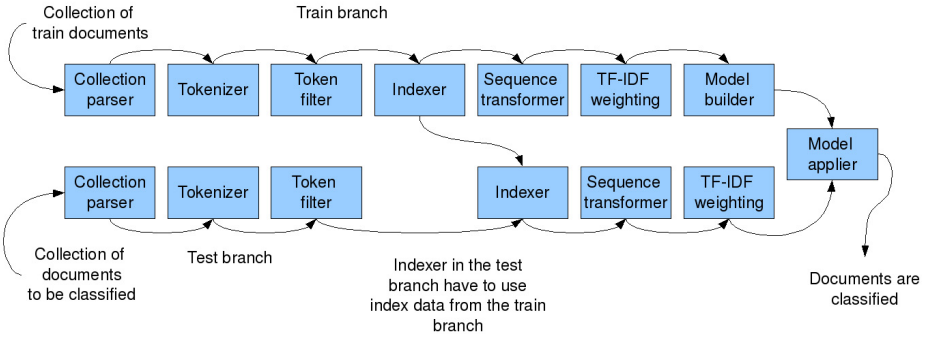


Fig. 1. Text mining workflow scheme.

We used our components to create executable workflow from the abstract workflow which contains only places annotated with assumed OWL classes and transitions, connecting these places into workflow. Workflow was then refined. There were some catches in deciding which service must be used. For instance Tokenizer and Token filter have the same combination of input/output (documents in this case). Semantically they differ only in preconditions and effects. Token filter expects that input documents are already tokenized. Tokenizer produces tokenized documents. These things are expressed as precondition/effects in OWL-S profiles of Token filter and Tokenizer.

4 Conclusions

Our approach briefly described in this paper is based on utilization not only the semantic annotations of services, but also the produced results in previous executions of workflows and their meta-information in form of QoS parameters. This information helps us to minimize the human participation that is left only from the reasons of utilization of domain experts directly and better workflow control during the execution phase.

References

1. Bubak, M., Gubala, T., Kapalka, M., Malawski, M., Rycerz, K.: Grid Service Registry for Workflow Composition Framework. ICCS 2004. LNCS vol. 3038, pp. 34-41. Springer (2004)
2. VDS - The GriPhyN Virtual Data System. http://www.ci.uchicago.edu/wiki/bin/view/VDS/VDSWeb/WebMain
3. Knowledge-based Workflow System for Grid Applications (K-Wf Grid). EU 6th FP Project, 2004-2007. http://www.kwfgrid.eu
4. Nowakowski, P., Hoheisel, A.: Report on Implementation and Testing of Basic System Components. Public Deliverable, K-Wf Grid project (2007)

Architecture of a System Supporting Business Alliances

Karol Furdík[1], Marián Mach[2], Tomáš Sabol[3]

[1] InterSoft, a.s., Floriánska 19
040 01 Košice, Slovakia
karol.furdik@intersoft.sk

[2] Technical University of Košice
Faculty of Electrical Engineering and Informatics
Letná 9, 042 00 Košice, Slovakia
marian.mach @tuke.sk

[3] Technical University of Košice
Faculty of Economics
Letná 9, 042 01 Košice, Slovakia
tomas.sabol@tuke.sk

Abstract. The paper describes the architecture of a system supporting ad-hoc creation of short-term networked enterprises and business alliances, as it was designed within the SPIKE EU FP7 project. The approach combines the business process modelling with Semantic Web services. Overview of the main system components and data elements is presented, together with a brief functional description.

Keywords: Business alliances, networked enterprises, semantic modeling, identity management, service-oriented architecture.

1 Introduction

Standardization of the Semantic Web services and especially its connection with business process modeling frameworks gives an opportunity to develop solutions supporting the interoperability between the organizations cooperating in a business world [2], [3]. The *Secure Process-oriented Integrative Service Infrastructure for Networked Enterprises* (SPIKE, www.spike-project.eu) international EU IST project No. FP7-ICT-217098 addresses this issue, aiming at design and implementation of a system for enterprises of all sizes to be used for realizing competitive advantage via forming business alliances. The project started in January 2008 and will last 3 years. Consortium consists of eight partners from five European countries. The project is coordinated by University of Regensburg, Germany.

Goals of the SPIKE project can be summarized on organizational as well as scientific and technological levels. The organizational objective is to allow secure fast set-up and management of networked enterprises, whereas the main technical objective is to develop generic solutions for inter-enterprise interoperability and

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 53–57, ISBN 978-80-227-3027-3.

collaboration through reference scenarios and guidelines for their use. The envisioned SPIKE system should bring flexibility to the project-based collaboration between networked enterprises and will allow to gain business opportunities with previously inaccessible customers and partnering organizations.

2 Functionality, Scope and Context of the System

Following the methodology of Rozanski and Wood for system architecture design [4], we have identified the viewpoints, perspectives, and stakeholders of the SPIKE system. Then, based on the description of required functionality provided by the user partners of the project [1], we have specified the system scope and context. The scope of the SPIKE system, which roughly corresponds with the functional viewpoint, is determined by the envisioned functionality of the system as whole. The SPIKE system should provide a technical support for the collaboration of business entities in the form of a temporary business alliance involving technical set-up of the alliance, running, and final closing-down of the alliance. The following three different forms/levels of the collaboration are expected to be supported by the SPIKE system.

- *Collaborative processes*. The collaboration among alliance partners has the form of a set of complex collaboration patterns, which can be modeled as a (business) process enabling to produce an artefact (physical and/or intangible). This process is composed from different steps/activities, which represent contributions of the alliance partners to the collaborative output, where each step represents an activity (task) of one of the partners. These activities form a workflow, including definition of conditioning, relationships between, and an ordering of the activities.
- *Sharing services*. The alliance partners can offer their services to other partners in the scope of a given business process. The offered services can be retrieved, negotiated, contracted, and finally used by some of the alliance partners according to the conditions specified by the service contract.
- *Identity federation*. In this case, the SPIKE system serves as a mediator to enable alliance partners to get access to and to utilize resources of other partners. It allows individuals (employees of an alliance partner) to get access to a network of a collaborative partner using the same account and password as they use in their home company.

These three forms of collaboration can be used independently one from another, but can also be combined together. For example, it will be possible to offer a service, which is not atomic but can be composed from several steps (performed by one or more partners). These steps can be glued together in the form of a collaborative process – this way, the collaborative effort of some alliance partners can be offered to the other partners as a service to be shared. On the other hand, a collaborative process can be regarded as a (non-atomic, i.e. composed) service.

The system context is given by the external entities, actors that communicate with the system. The context diagram, presented on Fig. 1, provides the highest level view as a specification of the system's boundaries and its adjacent external entities. The roles of actors (both human and software agents) were identified, together with the

objects and data entities affected in the communication process. More detailed specification of the interaction of actors with the system was included into the concurrency and operational viewpoints, as well as into the usability perspective of the architecture design.

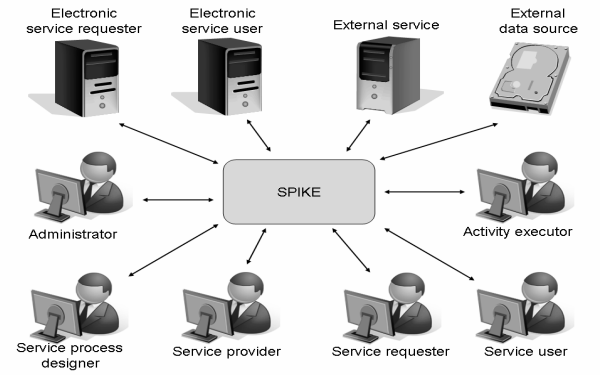


Fig. 1. System context diagram, external actors communicating with the SPIKE system.

3 System architecture, Data View

In the context of the information viewpoint, three main component groups were identified for the SPIKE solution, namely a) core of the SPIKE system, b) client applications, and c) external services. These components, as depicted on Fig. 2, are supposed to communicate via standardized API or Web Service interface provided and maintained by the system core.

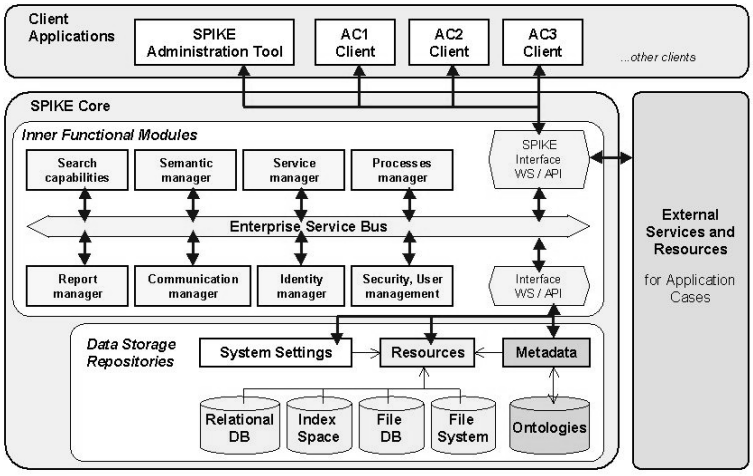


Fig. 2. Basic components of the SPIKE solution.

The *SPIKE core* component group consists of inner functional modules – managers that covering the business logic of the system. These modules will provide the temporary (real-time) instances of the data elements for business processes and their elements, inner representation of annotated services and other resources used in the workflow, security and identity management, retrieval, communication, and reporting capabilities. The in-memory data are composed, mediated and exchanged via the enterprise service bus module and are provided for client applications. The Data Storage Repositories module is used for persistent storage of physical information resources, system configuration data, and semantic metadata. Basic data repositories are assumed to be the ontologies, relational databases, file systems, and indexing space to store the content and properties of the resources used within a business process and to provide them to the functional modules.

Client applications component group covers the administration tools for maintenance and configuration of a single SPIKE installation, as well as the client-side tools that will be developed for particular application cases [1].

External services and resources component group represents the services located and hosted on the premises of users – alliance partners. These services are semantically annotated and registered in the SPIKE data repository. After that, the annotated external services can be included as resource elements into a workflow and used (consumed) within a business process, mediated by the components responsible for business logic – Inner functional modules.

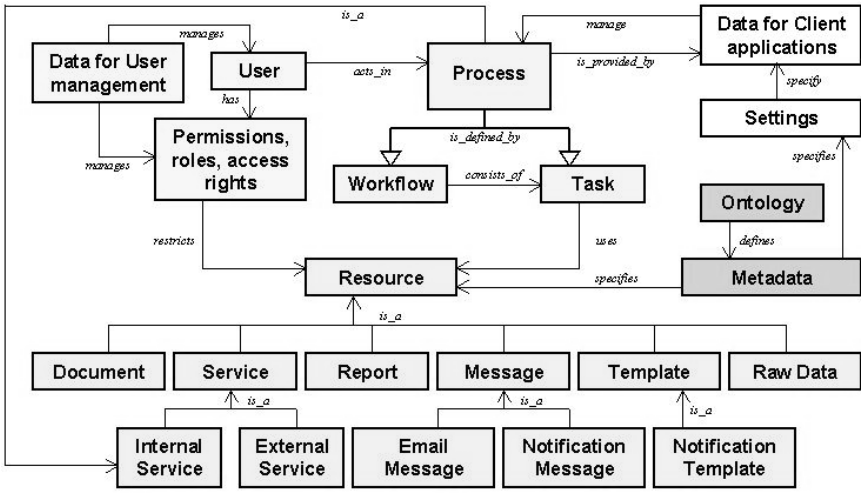


Fig. 3. Structure of basic information resources and data elements.

The structure of basic information resources and corresponding data elements is depicted on the Fig. 3. The *Process*, *Workflow*, and *Task* elements are basic logical data components and building blocks for specification of the collaborative business processes. The *Task* elements, being representations of workflow actions, are further specified by the inputs, transformations, and outputs, represented by the sub-types of the *Resource* data element. This element is a generic parent concept that defines a common set of properties inherited by the child data elements, particular resource

types as *Document*, *Service*, *Report*, etc. Properties of the resources are provided as semantic metadata, defined in the ontology schema. This solution enables to combine the standardized business process modeling with semantic descriptions according to the Semantic Web principles and as such, it should be flexible enough and capable to create a platform for collaboration and interoperability of business processes.

3 Conclusions and Future Work

In the paper we have briefly described some basic notions and principles of the architecture for the system supporting networked enterprises, as it was accomplished within the SPIKE project. Currently (October 2008), the architecture design is almost finished; we have already started the specification of functional components and semantic mark-up platform. First release of the implemented system should be ready in September 2009. After that, the system will be tested on two pilot applications in Finland (application case: documentation management) and in Austria (two application cases: business alliances and identity federation).

Acknowledgments. The SPIKE project is co-funded by the European Commission within the contract No. FP7-ICT-217098. The work presented in the paper was also supported by the Slovak Grant Agency of the Ministry of Education and Academy of Science of the Slovak Republic within the 1/4074/07 Project "Methods for annotation, search, creation, and accessing knowledge employing metadata for semantic description of knowledge".

References

1. Agudo, I., Broser, Ch., Fernàndez Gago, C., Fritsch, Ch., Furdik, K., Gmelch, O., Mach, M., Possegger, A., Sabol, T., Schillinger, R.: D2.2: User requirements analysis and development/test recommendations. Technical report of the SPIKE project, FP7-ICT-217098. University of Regensburg (2008)
2. Dimitrov, M., Simov, A., Stein, S., Konstantinov, M.: A BPMO Based Semantic Business Process Modelling Environment. In: Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management (SBPM-2007), Vol-251, CEUR-WS (2007)
3. Hepp, M., Leymann, F., Domingue, J., Wahler, A., Fensel, D.: Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management. In: Proceedings of the IEEE ICEBE 2005, October 18-20, Beijing, China, pp. 535--540 (2005)
4. Rozanski, N., Woods, E.: Software Systems Architecture. Working with Stakeholders Using Viewpoints and Perspectives. Addison Wesley (2005)

Spracovanie prirodzeného jazyka

JBowl as a Framework for Locally Informed Methods for Text Classification

Peter Smatana

Technical University of Košice
Faculty of Electrical Engineering and Informatics
Department of Cybernetics and Artificial Intelligence
Letná 9, Košice, 040 01, Slovak Republic
Peter.Smatana@tuke.sk

Abstract. There are some different approaches for textual document analysis. Most of them are based on statistical analysis. Statistical based approaches are very useful and important but the main problem is that they are using term reduction methods which cause losing of information in represented textual document. Our goal is to improve classification methods to decrease of amount of lost information. We can divide document to smaller parts which will be represented individually and in the process of statistical analysis individual block will represent information covered by that block instead of whole document representation in traditional approach. This paper presents basic idea of implementation of that approach to JBowl framework.

1 Motivation

Text classification problem is still actual area for the research. That field is fed by data from many CMS systems which have to annotate uploaded documents and it is helpful to recommend users some annotation categories. There are lots of others fields where classification can be helpful. Basic motivation point was how to minimize of losing of information without increasing of processing complexity. Next sections will show you how to use distributional information for better text classification and design of implementation of these functionalities to JBowl library which supporting text mining and retrieval tasks.

2 Background

In text classification, we are given a description $d \in X$ of a document, where X is the document space; and a fixed set of classes $C = \{c_1, c_2, ..., c_j\}$. Using a learning method or learning algorithm, we then wish to learn a classifier or classification function γ that maps documents to classes [1]:

$$\gamma = X \rightarrow C$$

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 61–64, ISBN 978-80-227-3027-3.

The basic schema for solving classification problem of textual document is shown in the Fig.1. That schema shows few steps how to achieve assigning of the class to processed document. Document is typically unstructured (or semi-structured) textual content.

Step of preprocessing is the most important and language dependent block which prepare text for representation block. There could be full linguistic analysis (morphology analysis, syntactic analysis, semantic analysis, etc.) which can be described as lossless information preprocessing step or there could be used losing information preprocessing methods (stemming, stop words, etc.). Problem of time complexity and vocabulary dependent preprocessing algorithms cause that those text classification algorithms using just losing information preprocessing methods. After document preprocessing step we are going to represent document for classification algorithm. There are different types of representations: term document matrix, n-grams, LSI, etc.

Last step is used for classification. There are lots of Machine Learning techniques as SVM (the top classifier for text documents [1]), KNN, Decision Trees, etc.

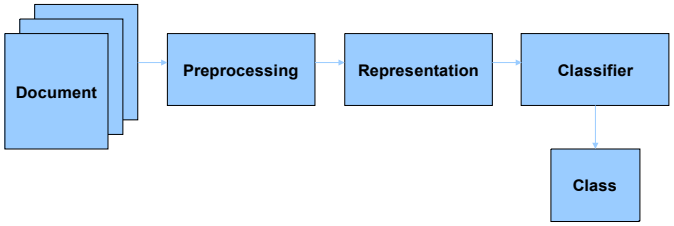


Fig. 1. Basic schema of statistical approach to text classification.

Traditional approach to text classification put all information to one bag but every document is divided to sections which are focused to different topics. This should be additional information for classifier and we miss it. Therefore it could be improving factor for better classification using distributional information about terms. Some techniques how to improve text classification by adding distributional term information to text representation are presented in different articles (Distributional features for text classification [2], Local word bag model [4, 5], Local Relevancy Weighed LSI [3]).

3 JBowl Library for Support of Text Mining and Retrieval

The JBowl Java library [7] was taken as an implementation platform for the improved representation methods for text classification. This open source software package was developed to support information retrieval and text mining tasks. The library is built on the modular framework architecture, which is highly extensible and supports SOA principles. Since it offers the required functionality for pre-processing, indexing and further exploration of text collections, it was chosen as a good candidate for implementing the classification tasks [8].

Jbowl has the same architecture as the standard Java Data Mining API (JSR 73 specification [9]). This architecture has three base components (application programming interface, text mining engine, mining object repository) that may be implemented as one executable or in a distributed environment.

Jbowl provides a set of common Java classes and interfaces that enable integration of various classification methods. The design of JBOWL distinguishes classification algorithms (i.e. SVM, linear perceptron, etc.) and classification models (i.e. linear classifier, rule based classifier).

4 Design and Implementation

JBowl framework was selected as implementation platform for modified representation method for text classification. We have added segmentation module (see Fig.2.) to analysis package for the sake of splitting document to continuous blocks. There could be different implementations of automatic text segmentation (see [10]). These parts of the document are individually classified. The final class/classes is/are chosen by classifier whose inputs are classes of document segments.

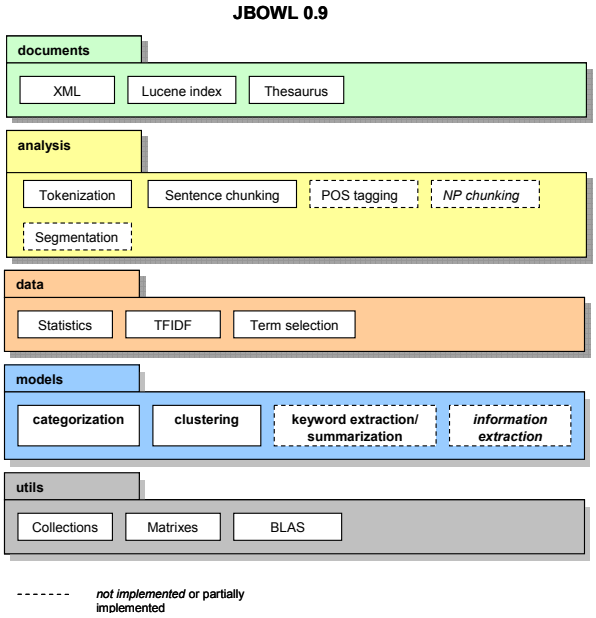


Fig. 2. JBowl library – platform for implementation of the method.

Sparsity is the major problem in the statistical text representation methods. That problem grows up in case of splitting document to smaller segments. We would like to solve problem of sparsity by using Gaussian function for weighting of term occurrence in part of the documents. It means that we will represent whole document when we would like to represent just specific segment of it but term which occurs in that segment will have higher value of weight.

5 Conclusion

There are lots of different approaches how to increase quality of results of classification task. It can be based on improving of a classification algorithm or based on a lot of other improving techniques but the most important point is to increase of quality of representation of textual documents. We could decrease of losing of information consisted in textual document by adding term distributional information to representation. We have decided to implement these methods as a part of the JBowl framework.

Locally informed algorithms still lose lot of information from textual document but it is compromise between losing information and computing complexity. The best approach for lossless representation and then categorization is using chain of natural language processing (NLP) techniques where the result of preprocessing is fully semantic representation of textual document. That approach is dependent on language of the documents because all preprocessing blocks are built on language specific characteristics. Therefore locally informed methods should be compromise between NLP methods and methods which use statistical approach over whole document.

References

1. Manning Ch. D., Raghavan P., Schütze H.: Introduction to Information Retrieval, 2008 Cambridge University Press, ISBN 0-521-86571-9
2. Xiao-Bing Xue, Zhi-Hua Zhou: Distributional Features for Text Categorization, IEEE Transactions on Knowledge and Data Engineering, 31 July 2008.
3. Tao Liu, Zheng Chen, Benyu Zhang, Wei-ying Ma, Gongyi Wu: Improving Text Classification using Local Latent Semantic Indexing, icdm, pp. 162-169, Fourth IEEE International Conference on Data Mining (ICDM'04), 2004
4. Wen Pu, Ning Liu, Shuicheng Yan, Jun Yan, Kunqing Xie, Zheng Chen: Local Word Bag Model for Text Categorization, icdm,pp.625-630, 2007 Seventh IEEE International Conference on Data Mining, 2007
5. Lebanon G., Mao Y., Dillon J.: The Locally Weighted Bag of Words Framework for Document Representation. Journal of Machine Learning Research 8 (2007)
6. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. ACM Press New York /Addison-Wesley, 1999. ISBN 0-201-39829-X
7. Bednár, P., Butka, P. (2005): JBOWL - Java Bag-Of-Words Library. In: 5th PhD student conference and scientific and technical competition of students of FEI TU Košice, Proc. from conference and competition, Košice, Slovakia, ISBN 80-969224-4-0 (2005) 19-20
8. P. Smrž, J. Paralič, P. Smatana, K. Furdík: Text Mining Services for Trialogical Learning, In: P. Mikulecký, J. Dvorský, M. Krátký (eds.), Proceedings of the Czech-Slovak scientific conference Znalosti (Knowledge) 2007, Ostrava, Česká republika, február 2007, s. 97-108, ISBN 978-80-248-1279-3.
9. JSR 73: Data Mining API: URL: http://www.jcp.org/en/jsr/detail?id=73, 2004
10. Reynar, J. C.: Topic segmentation: algorithms and applications. PhD thesis. University of Pennsylvania, 1998

Task-Based Execution Engine for JBOWL

Peter Bednár, Peter Butka

Centre for Information Technologies
Technical University of Košice
Boženy Němcovej 3, 040 01 Košice, Slovakia
{Peter.Butka,Peter.Bednar}@tuke.sk

Abstract. This paper describes design aspects for extension of our original software system developed in Java for support of information retrieval and text mining with specialized execution engine for different type of tasks. Some of our experiences and specific requirements of the real applications lead us to idea give the system possibility to run the tasks in distributive/parallel way. The result of the idea is task-based execution engine, which represents middleware-like transparent layer (mostly for programmers who want to re-use functionality of our package) for running of different tasks in multi-thread or distributive environment.

1 Introduction

Our research and education goals in the area of text mining and information retrieval with the emphasis of advanced knowledge technologies for the semantic web resulted in design and implementation of library for support of text-mining and information retrieval called JBOWL (Java Bag-Of-Words Library). More information about original library can be found in [1].

Some of our experiences and specific requirements of the real applications lead us to idea give the system possibility to run the tasks in multi-thread way. One example is our need to prepare education portal for text-mining for students on lectures of knowledge discovery from texts in domain of knowledge management. In our project (project Poznať [2]) one of the goals is to provide portal for the students (lectures), where many users could run several text-mining tasks with different collections and evaluate the results. This type of application then logically needs to run tasks on machine in multi-thread or distributive way. The result of this idea for extension is task-based execution engine, which represents middleware-like layer for running of different tasks in multi-thread/distributed environment.

2 Conceptual Architecture of JBOWL

JBOWL has the same architecture like standard Java Data Mining API (JSR 73 specification). Also new specification is in preparing phase, but it is not finished yet,

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 65–68, ISBN 978-80-227-3027-3.

so we would stick to JSR73 (with extensions if some new concepts seem to be interesting for our purposes). This architecture has three base components that may be implemented as one executable or in a distributed environment.

- *Application Programming Interface (API)* – The API is set of user-visible classes and interfaces that allow access to services provided by the text mining engine (TME). An application developer using JBOWL requires knowledge only of the API library, not of the other supporting components.
- *Text Mining Engine (TME)* – A TME provides the infrastructure that offers a set of text mining services to its API clients. TME can be implemented as a local library or as a server of client-server architecture.
- *Mining Object Repository (MOR)* – The TME uses a mining object repository which serves to persisting of text mining objects.

JBOWL provides set of common java classes and interfaces (API) that enable integration of various pre-processing, classification and clustering methods. The design of the library in case of model-building algorithms (classification and clustering models) distinguishes between algorithms (i.e. SVM, linear perceptron, etc.) and models (i.e. linear classifier, rule based classifier, SOM, etc.). Models are built using algorithms and then created models could be used in applying or evaluation on data inputs. More specific task is transformation of input data – processing of input sets of documents. According to this we have *Data Processing Tasks* (input datasets are pre-processed using these tasks), *Build Model Tasks* (responsible for building of appropriate model using chosen algorithm, several text-mining approaches have been already implemented in our package), and *Apply Model Tasks* (models created by algorithms are applied to new data inputs in order to classify/process new documents). Conceptually, also evaluation of models and their processing (with some evaluation metrics) can be seen in this set of tasks (although in implementation they will be differently modelled).

TME manages execution of common text mining tasks. It can be done as usage of local library or as running tasks in server-client architecture. Our main point is to extend JBOWL with its own layer for running of tasks in multi-thread and potentially distributive manner, where application developers are able to run tasks easily and they probably do not need to know, where the tasks are really executed, they expect results and place where they can find them.

Mining Object Repository (MOR) is used as a persistent storage for the processed text content and all artefacts generated during the text-mining process. Persistent objects include annotations of analyzed texts, data and evaluation statistics, indexed instances, text mining models and task settings. The implementation of the MOR is based on the Java Content Repository (JCR) specifications, which provide seamless integration with the existing content repositories.

In order to simplify integration of JBOWL and JCR, MOR contains Mining Object Manager component, which maps Java objects to JCR nodes for serialization/de-serialization. The mapping mechanism is generic and can be used to map arbitrary Java objects to JCR in the similar way, how the object-relational mapping frameworks (like Hibernate) are used to map Java objects to relational database.

3 Design and Implementation of Execution Engine

API of the Execution Engine is divided to client part and server part to simplify implementation of remote task execution. The main interfaces are (Fig.1):

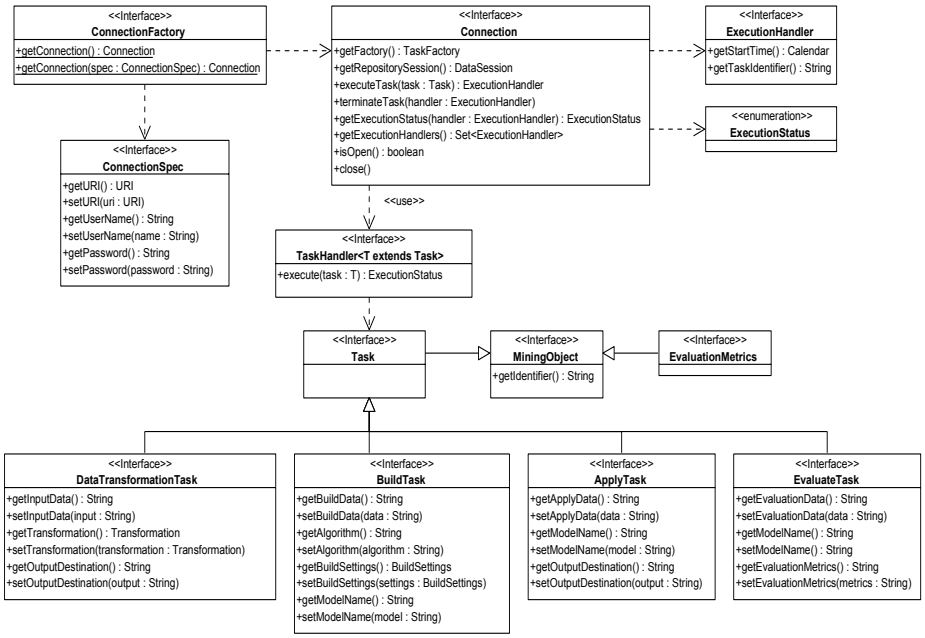


Fig. 1. UML design of Execution Engine interfaces.

Connection

To start working with Text Mining Engine (TME), client has to obtain *Connection* object which represent one text-mining session. Connection can be obtained in various ways, for example it can be created directly without user authentication or registered on the client environment using the JNDI. Client can specify details for connection specification like URI of the executed engine in the case that there are more TME instances, user name and password. *Connection* interface will allow client user to obtain factory class to create new mining objects (i.e. data, tasks, build and task settings etc.), obtain MOR session to save or load mining objects stored in the Mining Object Repository, and execute, inspect and terminate text-mining tasks.

Task and TaskHandler

API for tasks is divided to interfaces for task specification (*Task* interface) and for task execution (*TaskHandler* interface). Task objects are part of the client API and follow JavaBeans patterns, which allow simple encoding of the objects in the remote protocols. Task objects specify all parameters required for the specific task, like references to the input data, path where to store output data or models and all associated settings (build settings for algorithms, settings for data processing and settings for evaluation metrics). Each type of the Task object has associated

TaskHandler object responsible to execute this task. According to parameters specified in the Task object, TaskHandler object will create new parallel execution process and perform all operations defined for the task, e.g. Build Model Tasks task handler will load training data, create new instance of the algorithm specified as the task parameter, and pass training data and build settings to the algorithm to produce new text mining model. Model is then stored in the MOR on the path specified as the task parameter.

Execution Handler and Execution Status
When the Execution Engine creates thread for new task and execute task handler, the client invocation of the Execution Engine is immediately finished and task is executed on the background. Execution Engine will return to the client Execution handler, which identify running task and can be used to inspect execution status of the task process or to terminate task.

4 Conclusions

In this paper we have introduced task-based execution engine extension of JBOWL for support of multi-thread/distributed running of text-mining tasks. Important features of the extension is usage of Java Content Repository in Mining Object Repository as basic space for persisting of mining objects like documents and created models. This allows text mining engine layer (Execution Engine) to work with objects in more flexible way and then run (conceptually organized and encapsulated) tasks more easily. Implementation of our Execution Engine as internal and logical component of JBOWL provides support for wide types of applications.

Acknowledgement

The work presented in the paper is supported by the Slovak Research and Development Agency under the contract No. RPEU-0011-06 (project PoZnaŤ) and by the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic within the project No. 1/4074/07 ”Methods for annotation, search, creation, and accessing knowledge employing metadata for semantic description of knowledge”.

References

1. Bednar, P., Butka, P., Paralic, J.: Java Library for Support of Text Mining and Retrieval. In Proceedings of Znalosti 2005, pp.162-169, Stará Lesná, Slovakia (2005)
2. Babic, F., Furdik, K., Paralic, J., Wagner, J.: Podpora procesov tvorby nových znalostí. In Proceeding of DATAKON 2007, pp.198-201, Brno, Czech Republic (2007)

Sémantická anotácia pre podnikové aplikácie*

Michal Laclavík, Marek Ciglan, Zoltán Balogh, Martin Šeleng

Ústav informatiky, Slovenská akadémia vied
Dúbravská cesta 9, Bratislava, 845 07
laclavik.ui@savba.sk

Abstrakt. Sémantická anotácia textov sa snaží o identifikáciu a vytváranie metadát o objektoch na ktoré sa text odkazuje. Primárnym cieľom sémantickej anotácie je anotovanie alebo značkovanie dokumentov na webe. Sémantické metadáta je však možné využiť aj vo vnútro firemných aplikáciách. V článku popisujeme sémantickú anotáciu pomocou architektúry Ontea, kde je možné získať metadáta z rôznych dátových zdrojov v podnikoch a iných organizáciách ako sú databázy, web aplikácie alebo informačné systémy v organizáciách. Článok tiež opisuje experiment s anotáciou podnikovej emailovej komunikácie.

Kľúčové slová: sémantická anotácia, extrakcia informácií, metadáta, email

1 Úvod

Tvorba sémantických metadát je nutnou podmienkou pre využitie technológií sémantického webu. Prehľad v tejto oblasti je možné nájsť v článku z WIKT 2007 [1]. Na UISAV bola v minulosti vytvorená metóda na automatickú sémantickú anotáciu Ontea [2,3], ktorá je aj naďalej rozvíjaná[1]. Architektúra systému Ontea je navrhnutá tak aby v nej bolo možné využiť a zapojiť rôzne algoritmy a dátové zdroje na transformáciu extrahovaných dát. Výsledkom anotačnej metódy Ontea je vždy dvojica kľúč-hodnota, kde táto dvojica môže reprezentovať:

- Typ Objektu – Inštancia objektu
- Inštancia objektu – Vlastnosť objektu

Takéto dvojice môžu byť potom transformované rôznymi algoritmami na zisťovanie relevancie, lematizácie alebo transformácie na inštancie v ontológii RDF/OWL.

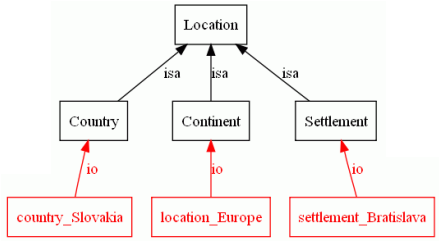
Príklad takejto transformácie môže byť nasledovný:

- Text: Slovensko je v Európe=>
- Extrakcia pomocou vzoru „(v|pri) +([A-Z][a-z]+)": *Location – Európe* =>
- Transformácia pomocou lematizácie: *Location – Európa* =>
- Transformácia pomocou hľadania alebo vytvárania objektov v rámci ontologického modelu za použitia odvodzovania (pozri obr. 1): *Continent – Europe*

* This work is supported by projects AIIA APVV-0216-07, Commius FP7-213876, SEMCO-WS APVV-0391-06, VEGA 2/7098/27.

[1] http://ontea.sourceforge.net/

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 69–73, ISBN 978-80-227-3027-3.



Obr. 1. Príklad jednoduchej ontológie geografických lokalít s inštanciami.

2 Adaptácia architektúry Ontea pre podnikové aplikácie

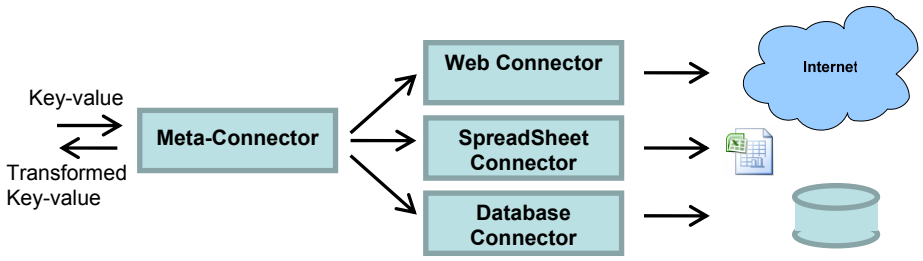
Tak ako v jednoduchom príklade uvedenom v kapitole 1, je možné vytvoriť aj iný typ transformácií, ktoré transformujú extrahovaný pár kľúč-hodnotu na iný pár. V rámci projektu Commius[2] [4], na podporu medzi podnikovej interoperability sú vytvárané tzv. Systémové konektory (SC – System Connectors), ktoré umožňujú jednoduchý prístup k dátam z web aplikácií, databáz a Excel dokumentov. Pomocou takýchto SC v kombinácií s automatickou anotáciou je možné napĺňať ontologický model aplikácie (tzv. Automatic Instance Population).

Systémové konektory sú komponenty, ktoré sprostredkúvajú prístup k existujúcim podnikovým systémom a verejným elektronickým informačným zdrojom. Cieľom je získať dodatočné parametre pre vstupy definované softvérovými komponentmi využívajúce SC. Konektory boli navrhnuté primárne pre obohacovanie výstupov anotačných nástrojov a systémov pre získavanie informácií dátami z externých systémov. Napríklad, v prípade, že textová anotačná metóda identifikuje v texte (e-mail, objednávka tovaru) meno osoby, SC môžu byť použité pre prístup k databáze obchodných kontaktov spoločnosti pre získanie mena a identifikátora spoločnosti, v ktorej je nájdená osoba zamestnaná. Ďalší konektor môže byť použitý pre získanie dodatočných informácií o spoločnosti z elektronického obchodného registra, nasledujúci konektor sa môže dopytovať na účtovný systém pre získanie informácií o n posledných transakciách s partnerskou spoločnosťou. Dodatočné informácie zo systémových konektorov sú následne spracované anotačným nástrojom a sú prezentované užívateľovi.

SC je špecifikovaný typom zdroja, ku ktorému sprostredkováva pripojenie (napr. relačná databáza, dokument tabuľkového procesora, webová aplikácia), identifikátor konkrétneho zdroja a metadáta popisujúce vstupy a výstupy konektora. Metadáta vstupov - výstupov sú n-tica parametrov (názov a dátový typ). SC vyžaduje ako vstup n-ticu zodpovedajúcu metadátam vstupov, alebo množinu párov kľúč-hodnota, ktoré môžu byť namapované na vstupné metadáta. Výstup SC je množina n-tíc zodpovedajúcich popisu metadát výstupov.

Systémové konektory je možné združovať do meta-konektorov (obrázok 2), ktoré transformujú páry kľúč-hodnota na nový pár podľa nastavenia meta-konentora. Funkcionalitu si môžeme ukázať na príklad transformácie podobnom ako v nasledujúcom experimente v kapitole 3.

[2] http://www.commius.eu/



Obr. 2. Schéma Systém konektorov a ich prepojenia do meta-konnektora.

Napríklad pri komunikácii s potenciálnymi firemnými zákazníkmi by sme chceli zistiť základné informácie o firme odosielateľa:

- Z textu emailu pomocou vzoru pre doménu zistíme internetové domény ([_-a-zA-Z0-9]+\\.sk), napr.: *domain:SK – toyota.sk*
- Pomocou stránky slovenského registrátora SK-NIC transformujeme pár na: *IČO – 31585973*
- Pomocou SC pre web aplikáciu a stránky obchodného registra transformujeme pár na: *company:Name – TOYOTA MOTOR SLOVAKIA s.r.o.* alebo prípadne ďalšie údaje o objekte firmy ako adresa, telefónne číslo, a pod.
- Tieto údaje môžu byť následne použité na nájdenie alebo vytvorenie inštancie v ontologickom modeli aplikácie prípadne doplnenie ďalších vlastností objektu k ontologickému modelu.

Na tomto príklade sme si ukázali ako môžu informácie dostupné na webe alebo v rámci organizácie obohatiť sémantickú anotáciu ako aj prispieť k identifikácii a lepšej sémantike objektov reprezentovaných a odkazovaných v texte anotovaných emailových dokumentov.

V tomto článku ako aj v našom výskume sa zameriavame hlavne na anotáciu emailovej komunikácie. Podľa najnovších prehľadov, ľudia vo firmách pracujúci z informáciami denne odošlú a príjmu 133 emailov [8]. Email je najpoužívanejšou službou internetu. Email sa v súčasnosti nepoužíva len na komunikáciu ale aj na ďalšie funkcie [5,6] ako upozornenia, archivovanie, manažovanie úloh, na podporu spolupráce alebo podporu medzi podnikovej interoperability. Okrem toho veľa súčasných Web 2.0 systémov používa emailovú komunikáciu [7] počítač – človek na zasielanie upozornení, objednávok, faktúr, registrácií, a pod.

3 Experiment

Experiment bol uskutočnený na podnikových dátach poskytnutých firmou zaoberajúcou sa poskytovaním internetových služieb hlavne webhostingom. Dáta obsahovali sadu 8579 emailov a databázu zákazníkov a služieb. Firma poskytla štruktúru databázy a príklady na základe akých objektov môže byť identifikovaný zákazník. Na základe týchto informácií bol vytvorený program ktorý spracovával emaily vo formáte mbox (http://www.qmail.org/man/man5/mbox.html) a výstupom boli dáta potrebné na vytvorenie štatistiky ako v tabuľke 1. Potom bol tento program

spustený v rámci firmy na citlivých firemných dátach a výstupy boli poskytnuté pre tvorbu tabuľky 1.

Cieľom experimentu bolo identifikovať zákazníka z emailovej správy. Zákazník sa identifikoval na základe nasledovných štyroch typov objektov:

- Adresa odosielateľa
- Meno firmy
- Telefónne číslo
- Internetová doména ktorá reprezentuje službu poskytovanú zákazníkovi

Pre tieto objekty boli definované vzory pomocou regulárnych výrazov pre nástroj Ontea. Následne tieto výsledky boli transformované pomocou Systém konektorov pre databázy a ich integrácie do SC Meta-konektora. Výsledkom bola identifikácia zákazníka z databázy zákazníkov.

Kombináciu podnikových dátových zdrojov (databáz, informačných systémov alebo web aplikácií) možno dosiahnuť identifikáciu dôležitých objektov v e-mailovej komunikácii alebo v archívoch dokumentov ako sú produkty, služby, dodávatelia alebo odberatelia. Tieto objekty je možné využiť na tvorbu lepšieho vyhľadávania, kategorizovania alebo tvorbu poradenských aplikácií, ako napr. Acoma [9].

Tabuľka 1. Experiment identifikácie zákazníka z e-mailovej komunikácie.

	Všetkých nájdení	%	**Správnych**	%	**Jeden zákazník**	%
Doména	4658	77%	4658	81%	3142	79%
Odosielateľ	516	9%	516	9%	356	9%
Firma	767	13%	507	9%	413	10%
Telefón	88	1%	88	2%	54	1%
Spolu	6029		5769		3965	
Emailov	**Najdený zákazník**		**Správnych**		**Práve jeden**	
8579	4608	54%	4348	51%	3965	46%

Z experimentu na reálnych dátach je vidieť, že zákazník bol identifikovaný v 50% e-mailových správ. Nesprávne bol zákazník identifikovaný v 3% prípadoch a to v prípade identifikácie podľa mena firmy, kde boli v texte nesprávne identifikované mená firiem pomocou druhého slova z názvu firmy ako *Slovakia*, *Slovensko*, *Systems* alebo *Consulting.* V prípade identifikácie podľa odosielateľa, telefónu alebo domény bol zákazník identifikovaný vždy správne. Z tabuľky je vidieť že v niektorých emailoch boli identifikovaní viacerí zákazníci, jednalo sa o emaily ktoré sa týkali viacerých zákazníkov z dôvodu komunikácie z dodávateľmi, prípadne sa jednalo o zákazníkov, kde došlo k rôznym zmenám. Posledný stĺpec ukazuje prípad keď bol identifikovaný práve jeden zákazník pomocou jedného z objektov extrakcie. Vidíme že aj keď väčšina zákazníkov bola identifikovaná pomocou internetovej domény – teda služby, ktorú si zákazník objednával. Nezanedbateľná časť (20%) identifikácií sa udiala pomocou iných objektov ako telefón, odosielateľ alebo meno firmy.

Využitie tohto konkrétneho experimentu je možné napríklad pri použití poradenského systému Acoma [9], ktorý by pri identifikácii zákazníka z prichádzajúcej správy hneď poskytol zamestnancovi informácie o zákazníkovi ako aj linky a kroky ktoré je potrebné urobiť na vyriešenie úlohy ktorú email reprezentuje.

4 Zhrnutie a ďalšia práca

V článku sme v krátkosti opísali ako je možné využiť pri sémantickej anotácii iné dátové zdroje ako text anotovaného dokumentu a ontologický model. Takéto postupy sémantickej anotácie sú zvlášť vhodné v podnikových aplikáciách, čo sme demonštrovali na jednoduchom príklade ako aj experimentálne overovali na podnikových dátach.

V ďalšej práci by sme sa chceli venovať využitiu takto získaných sémantických metadát v prostredí spracovania podnikovej emailovej komunikácie. Emailová komunikácia obsahuje kontext úlohy ktorú treba riešiť. Pomocou sémantickej anotácie a použitia dát z externých systémov vieme lepšie formalizovane opísať kontext emailu. Na základe tohto kontextu vieme užívateľovi poskytnúť rady a akcie (pomocou systému Acoma [9]), ktoré môže vykonať na splnenie úlohy obsiahnutej v emailovej správe.

V rámci emailovej komunikácie je tiež možno extrahovať graf sociálnej siete s prepojením na objekty ako úlohy, projekty, zákazníci, služby produkty a iné biznis objekty. Obohatený sémantický graf je možné získať iba s použitím externých dátových zdrojov v organizácii tak ako je opísané v tomto článku. Graf môže potom slúžiť na lepšie vyhľadávanie a manažment emailovej komunikácie.

Literatúra

1. Laclavík, M.: Tvorba sémantických metadát. In WIKT 2007 proceedings, Košice, 2008. ISBN 978-80-89284-10-8, pp. 2-6.
2. Laclavík, M., Šeleng, M., Hluchý, L.: Ontea: Platforma pre sémantickú anotáciu založenú na vzoroch. In ZNALOSTI 2008, pp. 351-354.
3. Laclavík, Michal, Šeleng, M., Hluchý, L.: Towards large scale semantic annotation built on MapReduce architecture. In Computational Science – ICCS 2008, ISSN 0302-9743, 2008, LNCS 5102, part III.
4. Laclavik, M.: Commius: ISU Via Email, Workshop on Enterprise Interoperability Cluster: Advancing European Research in Enter. Interoperability II at eChallenges 2007 conference; http://laclavik.net/publications/laclavik_isu_email.ppt
5. Whittaker, S., Sidner, C.: Email Overload: Exploring Personal Information Management of Email. In Proceedings of ACM CHI'96 Conference on Human Factors in Computing Systems, (1996) 276-283.
6. Fisher, D., Brush, A.J., Gleave, E., & Smith, M.A. (2006). Revisiting Whittaker & Sidner's "email overload" ten years later. In CSCW2006, New York ACM Press.
7. Matt Blumberg, Case Study: Web 2.0 Runs on Email, Return Path, Inc., 2008 http://www.returnpath.net/blog/2008/07/case-study-web-20-runs-on-emai.php
8. White paper by The Radicati Group, Inc.; Taming the growth of email: An ROI analysis; https://h30046.www3.hp.com/campaigns/2005/promo-evolution/1-1LRYR/images/Preview_Radicati.pdf, 2008
9. Laclavik, M., Seleng, M., Gatial, E., Hluchy, L.: Future Email Services and Applications; Proc. of the Poster and Demonstration Paper Track of the 1st Future Internet Symposium (FIS'08), CEUR-WS, Vol. 399, (2008) 33-35.

Monitorovanie správ rozšírené o metódy analýzy sentimentu*

Ivana Budinská, Zoltán Balogh, Emil Gatial

Ústav informatiky SAV, Dúbravská cesta 9, 845 07 Bratislava, Slovensko
{Ivana.Budinska,Zoltan.Balogh,Emil.Gatial}@savba.sk

Abstrakt. S rozvojom metód a techník na spracovanie prirodzeného jazyka a to v písanej aj hovorenej forme, viaceré komerčné firmy ponúkajú mediálny monitoring na sledovanie ohlasov v tlačených a iných médiách. Tieto služby sú využívané ako veľkými firmami a organizáciami, tak aj politickými stranami a subjektami verejnej a štátnej správy, rôznymi odvetviami hospodárstva a podobne. Monitorovanie mediálnych ohlasov je však len prvým krokom v spracovaní získaných údajov. Omnoho dôležitejším faktorom je určenie charakteru príspevku, či ide o pozitívnu, resp. negatívnu reakciu. Takéto vyhodnotenie je možné využiť v riadení firiem, pri vypracovávaní politík, biznisových plánov, na ovplyvňovanie verejnej mienky a pod.. Výskum v tejto oblasti na Ústave informatiky SAV bol iniciovaný prípravou návrhu projektu 7RP a nadväzuje na skúsenosti v oblasti spracovania textových dokumentov v projekte NÁZOU. Výskum v oblasti analýzy sentimentu je interdisciplinárny výskum, ktorý okrem informatických technológií, predovšetkým z oblasti spracovania textu, zahŕňa aj výskum jazykovedný a čiastočne aj psycho-sociologický. Jedine dokonalou súhrou všetkých zložiek môže výskum v tejto oblasti priniesť užitočné výsledky.

Kľúčové slová: spracovanie prirodzeného jazyka, strojové učenie, analýza sentimentu

1 Úvod

Metóda nazývaná v anglickej literature "sentiment analysis" sa používa na vyhodnotenie názoru autora textu k predmetu článku. V roku 2004 publikovali Bo Pang a Lillian Lee použitie tejto metódy na analýzu recenzií filmov a určenie či ide o kladnú alebo zápornú recenziu. Neskôr sa táto metóda začala využívať na analýzu názorov zákazníkov na produkty. Možnosti tejto metódy sú však omnoho rozsiahlejšie. Aplikačná oblasť pripravovaného projektu zahŕňa oficiálne aj neoficiálne správy na internete. Dvaja z najväčších poskytovateľov priestoru pre publikovanie správ, Yahoo a Google, uverejňujú denne okolo 100 000 správ (zdroj http://news.yahoo.com.,http://news.google.com). Spracovanie veľkého množstva

* Príspevok vznikol za podpory projektov VEGA 2/7101/27, SEMCO-WS APVV 0391-06.

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 74–77, ISBN 978-80-227-3027-3.

textových správ metódami analýzy sentimentu a porovnanie výsledkov analýzy textov z tzv.oficiálnych zdrojov a osobných neoficiálnych zdrojov prezentovaných napríklad prostredníctvom blogov, diskusných skupín a pod., umožní predikovať vývoj nálad spoločnosti ale napr. aj stanoviť miery na určenie slobody prejavu v tlačených médiách. Práca bude sústredená výlučne na spracovanie písaných textových zdrojov na internete.

2 Koncept emócií v písanom texte

Systémy využívajúce metódu analýzy sentimentu sa snažia určiť pocity a názory vyjadrené v texte formou prirodzeného jazyka. Súčasné systémy sa zameriavajú na spracovanie textov z určitej konkrétnej domény. Kompletná analýza textu sa musí robiť s ohľadom na spracovávanú doménu. Po prvýkrát bola metóda automatickej analýzy sentimentu popísaná v roku 2004 v dielach [6, 5]. Aplikačná doména na ktorú bola použitá metóda popisovaná boli recenzie filmov. Pomocou metódy analýzy sentimentu boli kategorizované recenzie na kladné a záporné.

Na vyhodnotenie sentimentu textového dokumentu je potrebné určiť miery. Podľa uznávaného diela Osgooda z roku 1957 [10] (citované v [2]), v ktorom skúmal význam slov a ich mapovanie do sémantickho priestoru, rozoznávame tri emočné rozmery písaného textu: (1) *hodnotenie* - pozitívne alebo negatívne, (2) *účinok, potencia* - silný alebo slabý a (3) *intenzita* – sila vyjadrenej emócie. Druhý rozmer – potencia - obsahuje tri ďalšie podrozmery: *vzdialenosť* – vyjaruje vzťah autora k téme, *špecifickosť* – jasná, vágna formulácie, *určitosť* – autor si je istý, autor pochybuje. Pomocou týchto troch rozmerov bol Osgoodom definovaný sémantický priestor. Rovnako sa však tieto rozmery dajú použiť na organizovanie lingvistických kategórií vyjadrujúcich pocity a následne na automatickú detekciu pocitov a analýzu sentimentu daného textu. Metódy klasifikácie založené na analýze sentimentu pracujú so slovami a výrazmi, ktoré nie sú priamym vyjadrením pocitov, ale označujú hodnotenie, ocenenie a úsudok o danej téme. Sentiment daného textu je možné vyjadriť aj používaním iných lingvistických prostriedkov, ako je irónia, sarkazmus, obrazná reč a podobne. Špecifickým prostriedkom na vyjadrenie pocitov v textoch používaných na internete je používanie grafík – tzv. emotikonov (smajlíkov). Predpokladom úspešnej analýzy sentimentu je vytvorenie preddefinovaných slovníkov, kde slová, resp skupiny slov sú zoskupené podľa psycho-sociálnych kritérií, ktoré sa použijú na meranie polarity textov.

3 Metódy analýzy sentimentu

Analýza sentimentu v textových správach využíva dva základné prístupy k spracovaniu prirodzeného jazyka: (1) *symbolické metódy* a (2) *metódy strojového učenia*.

Symbolické metódy založené na lexike spracovávajú text ako súbor slov bez akéhokoľvek vzťahu medzi slovami. Medzi tieto techniky patrí používanie WebSearch a WordNet. Pri tejto metóde je potrebné určiť sentiment jednotlivých

slov a sentiment celého dokumentu sa získava pomocou vhodnej agregačnej funkcie. *Symbolické metódy založené na vetách* kombinujú lexikografický prístup s gramatikou. Lexika je dôležitá pre určenie významu jednotlivých slov a gramatika umožňuje odhaliť nový, resp presnejší význam jednotlivých slov podľa vetnej gramatickej konštrukcie.

Experimenty urobené Pang a Lee ukázali (prevzaté z [2]), že je veľmi ťažké vytvoriť slovník, resp. znalostnú bázu, ktorá (takmer) kompletne pokryje cieľovú doménu. Presnosť týchto techník sa pohybuje od 65,8 do 84 %. Experimenty boli robené pomocou WebSearch v doméne filmových recenzií a recenzií na automobily. Ďalšie experimenty boli robené na porovnanie manuálne a automaticky vytvoreného slovníka, ktorý sa používa na strojovú klasifikáciu dokumentov. Jednoduchá automatická metóda (počítanie frekvencie slov v kladných a záporných textoch) priniesla o niečo lepšie výsledky ako manuálne vytvorenie slovníka. Prekvapujúco boli indexované aj niektoré neutrálne slová (napr. slovo still ako pozitívne slovo, pretože sa vyskytovalo vo vetách "Still, though, it was worth seeing.")
Metódy strojového učenia boli testované Boyiom a kol. [2]. Na základe zverejnených výsledkov testov ([2, 1]) sa pre ďalší výskum v rámci pripravovaného projektu navrhuje použitie metódy dolovania textov. Metóda "suffix arrays" (polia prípon), popísaná v [11], spĺňa požiadavky na výkonnosť a použiteľnosť rámci predpokladanej domény. Výsledky publikované v [1], ktoré sa týkajú vytvárania a použitia kompresovaných polí prípon ([4, 8, 9]) ukazujú, že použitie polí prípon je efektívnejšie ako používanie stromov prípon ([3]). Callison a kol. [7] používajú metódu polí prípon aj na ukladanie, rýchle objavovanie a počítanie informácií, potrebné pre štatistické strojové prekladanie.

4 Výzvy na výskum v oblasti analýzy sentimentu

Medzi veľké výzvy vo výskume metód analýzy sentimentu patrí spracovanie textov z rôznych domén a vo viacerých jazykoch. V ideálnom prípade by išlo o vytvorenie systému na analýzu sentimentu podporujúceho všetkých 23 oficiálnych jazykov EÚ.

Vzhľadom na požiadavku spracovania veľmi veľkého počtu dát, výskum a vývoj systému na analýzu sentimentu zahŕňa aj špecifické požiadavky na paralelné spracovanie dát a vyladenie testovacieho prostredia, ktoré je odlišné od tradičných vysoko-výkonných paralelných výpočtových systémov. Otázky, ktoré je potrebné riešiť v rámci výskumu:

1. Efektívne vyhľadávanie a spracovanie, paralelné spracovanie a paralelný vstup/výstup pre veľmi veľké množstvo textových správ.
2. Výskum alternatívnych spôsobov rozdeľovania a replikovania textu
3. Integrácia distribuovaných heterogénnych komponentov vo vysoko interaktívnom prostredí s podporou vykonávanie na lokálnych klastroch (pre experimentovanie) a na rozsiahlych gridoch (na zabezpečenie prístupu k distribuovaným zdrojom textových dokumentov a na vykonávanie veľkého počtu výpočtových úloh pre analýzu sentimentu, strojový preklad a predikciu názorov).

5 Záver

Pripravovaný projekt má za cieľ rozšíriť súčasné možnosti analýzy sentimentu na široko použiteľnú a multilingválnu verziu. Zapojením psycho-sociologického výskumu sa plánuje otestovať nové možnosti analýzy sentimentu v rámci zverejňovaných správ a to najmä porovnávaním tzv. oficiálnych zdrojov ako sú tlačené médiá zverejnené na internete s neoficiálnymi zdrojmi – napríklad blogmi, diskusnými skupinami a pod. Ďalší modul pripravovaného projektu má za cieľ na základe analýzy sentimentu a porovnávania oficiálnych a neoficiálnych správ predpovedať vývoj spoločenskej situácie v blízkej budúcnosti. Interdisciplinárnosť a medzinárodná spolupráca sú dôležité faktory pre úspešnosť pripravovaného projektu.

Referencie

1. Abouelhoda M. I. and S. Kurtz and E. Ohlebush. 2004. "Replacing Suffix Trees with Enhanced Suffix Arrays". Journal of Discrete Algorithms (Elsevier), Vol. 2, pp. 53—86.
2. Boiy E., Hens P, Deschacht K., Moens M.F.: Automatic Sentiment Analysis in On-line Texts, Proc ELPUB 2007 Conference on Electronic Publishing – Vienna, Austria – June 2007, pp.349
3. Martin Kay. 2004. "Sub string Alignment Using Suffix Trees". In: A. Gelbukh (ed.). CICLing. Lecture Notes in Computer Science, Vol. 2945, pp. 275-282. Berlin: Springer-Verlag.
4. Manzini G. and P. Ferragina. 2002. "Engineering a lightweight suffix array construction Algorithm",Technical Report TR-INF-2003-02-02-UNIPMN, Dipartimento di Informatica, Università del PiemonteOrientale, Italy (17 pages).
5. Mullen T., Collier N. (2004), Sentiment Analysis using Support Vector Machines with Diverse Information Sources, In Proceedings of the EMNLP 2004.
6. Pang B., Lee L. (2004), A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts, In Proceedings of the ACM 2004.
7. Callison-Burch C. and Bannard C. and Schroeder J. 2005. "Scaling Phrase-Based Statistical MachineTranslation to Longer Corpora and Longer Phrases". Proceedings of the 43rd Annual Meeting of Association for Computational Linguistics. pp. 255-262.
8. Sadakane Kunihiko. 2000. "Compressed Text Databases with Efficient Query Algorithms based on the Compressed Suffix Array". Proceedings of the 11th Annual International Symposium on Algorithms and FP7-ICT-2007-3 STREP proposal 08/04/08 EU-NeWa Proposal Part B: page 9 of 47 Computation (ISAAC'00). Lecture Notes on Computer Science, Vol. 1969, pp. 410-421, Berlin: Springer-Verlag.
9. Sadakane Kunihiko. 2002. "Succinct Representation of LCP Information and Improvements in the Compressed Suffix Arrays ". Proceedings of the thirteenth Annual ACM-SIAM Symposium on DiscreteAlgorithms (SODA02). ACM. pp. 225-232.
10. Osgood, C.E., Suci G.J., Tannenbaum, P.H. The measures of Meaning. University of Illinois Press, 1971 – citované v Boiy 2007
11. Yamamoto, Mikio, and Kenneth W. Church. 2001. "Using Suffix Arrays to Compute Term Frequency and Document Frequency for All Substrings in a Corpus". Computational Linguistics 27(1), pp. 1-30.

Establishing Semantic Annotation and Execution of the Text-Mining Services

Marian Babik[1], Martin Sarnovsky[2], Zoltan Durcik[2]

[1] Department of Parallel and Distributed Computing
Institute of Informatics, Slovak Academy of Sciences
[2] Center for Information Technologies, Technical University, Kosice
Marian.Babik@saske.sk, Martin.Sarnovsky@tuke.sk, Zoltan.Durcik@tuke.sk

Abstract. This paper covers an idea of estabilishing semantic annotation of web services in text mining application domain. The main aim is to leverage the advances in the Semantic Web to establish an automated framework for mining the textual data. Semantic description of text mining service should be a key-enabling technology for such frameworks by providing methods for automated discovery, composition and execution of the complex text mining processes. This paper provides an overview of estabilished semantic description of a set of text mining web services. This semantic description is then used in process of workflow execution, where we provide the two different approaches – corresponding mapping of OWL-S processes to Petri-net based workflow and BPEL.

1 Introduction

In the presented paper we introduce a semantic model intended to describe text mining web services, whereby the semantic information about the functional models and data can be utilized to construct the resulting executable workflow. When text mining tasks are represented as a web services, such approach opens ways to a plenty of various possibilities for building the text mining scenarios that can be represented as a workflows. The paper presents two various approaches to workflow execution, the first one based on Petri-Nets and BPEL. BPEL is very well supported by the software vendors as the choice for the execution of the web services, but on the other hand, it is not designed for addressing the challenges of the semantic web. Our approach is based on models of the semantics of the system, that are expressed in OWL-S and the execution is conducted in BPEL standard.

2 Adding Semantics to Text-mining Services

We have designed and developed a tool for generating the OWL-S description for stateful and stateless services from the corresponding web service descriptions (WSDLs). Such tool is inevitable in the text-mining area hosting a vast number of services, which have to be semantically described in order to enable automated

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 78–82, ISBN 978-80-227-3027-3.

discovery, composition and invocation. In the initial stage of the SEMCO-WS project we have successfully used the tool to create semantic descriptions of the text-mining services. The architecture of the so called Java Axis Semantic Annotation (JAX-SA) tool. The main components of the architecture are JAX-SA engine, translator and GOMOWL-S API. GOMOWL-S API is an extension of the Mindswap's OWL-S API; it defines additional vocabulary, converters and extensions needed by the WSRF (e.g. SimpleEffect, DataObjectInput, etc.). The translation procedure is quite complex and covers the areas already described in previous sections. The translation starts with a configuration and an URL of the WSDL document. The translator parses the WSDL document extracting the operations, port-types, inputs, outputs as well as resource properties. A combination of the WSDL4J [1], Axis WSDL [2] and Globus Toolkit WSDL utilities [5] are used in the process. The translator then generates for each WSDL operation a skeleton of the OWL-S document. Then it creates the inputs, outputs, preconditions and effects and maps the elements to the ontological concepts defined in the configuration. If needed, it will create an ontology, which models the resource properties of the given services. The GOMOWL-S API can be used to extend the OWL-S by the domain dependent constructs, e.g. FloodForecasting-WSRFProfile, DataObjectInput, SimpleEffect, etc. The outcome of the process is OWL-S document describing the web service operations, which are then be composed into the workflow [3]. Additionally a GridSphere portlet was developed to provide a graphical user interface for the tool [4]. It supports browsing of the concepts for any given ontology, associating the concepts with the WSDL elements and generation of the OWL-S documents. An automated annotation procedure based on the case-based reasoning is also integrated.

Different types of semantics can be used to represent the capabilities, requirements, effects and execution of Web services. We proposed conceptual model intended to describe web services in context of text mining application. In general, our model describes data and models as well, which correspond to inputs required by services and to the outputs produced by the various mining algorithms implemented as the Web services. Data package contains various concepts used to describe both physical data set and logical data passed as the input to the mining algorithms. Physical data refers to data physically stored in some data storage like formatted file, relational database table, XML file or OLAP cube. In the model we also consider the statistics sub-package, which is intended to constrain various limitations of the particular algorithm implementation. The statistics it self can be specified as the output of the services (i.e. it is possible to find services which can compute/estimate statistics used for constraining selection of the services in the next phase of the mining process). Also, supported text mining functions including various supervised models for predictive mining, clustering, association rules as well as attribute selection models for descriptive mining are included. In our case, Web service in general represents particular task in the text mining application. The whole life cycle of presented application consists of several steps (document indexation, preprocessing) in order to achieve the goal (building of the classifcation model). From different perspective, such

processes can be viewed as a complex workflows and involving the semantics can bring the benefits in areas of workflows construction.

3 Automated execution based on BPEL

Important task in this part is to develop the specific tool that will be able to perform the transformation of OWL-S descriptions into the BPEL specifications, as these two languages provide complimentary capabilities. BPEL is very well supported by the software vendors as the choice for the execution of the web services. BPEL is not designed for addressing the challenges of the semantic web. Some studies about attempting to adapt BPEL4WS for the Semantic Web Services are already available [8],[9], but our aim is to present an different approach by performing transformation from OWL-S to BPEL, so models of the semantics are expressed in OWL-S the execution is conducted in BPEL standard. BPEL makes use of the *activity* for process modeling. Basically, BPEL supports two types of activities : *Basic* and *Structured*. Basic activities (e.g. reply, receive) can be grouped together into structured ones in order to define BPEL workflow. Then the OWL-S atomic processes can be grounded using the WSDL. Using this we can obtain building blocks to create BPEL process models. The linking between building blocks of the workflow is represented by the plan produced during the workflow construction. Fully automated workflow construction is not actually implemented yet. Concrete workflow with all concrete services involved with concrete methods with particular parameters can be passed to the workflow engine and executed.

4 Automated execution based on Petri-nets

Petri nets are directed bipartite graph, which include two types of nodes, concrete transitions and places. Moreover they includes edges, which connect transitions with places. An edge from place p toward transitions t is marking as incoming edge of t, and the place p is marking as input place. Likewise are define output edge and places. Each place may containt several individual tokens that represent data items in a nets. The maximum number of tokens for one place represent its capacity. The transition is activate, if all tokens for its input places are availalbe, and for all output place hold, that they have not reachced their capacity. Then the activate transition may process input tokens and produce a new tokens for all output places. The number and values of tokens are given by current state of the workflow. Following markings are obtained by firing transitions. Each edge in the Petri Net can be adjuct an expression edge. For input edge are used as expression edges variable names, those values carries given tokens by this edge. In addition, each transition may have some boolean condition functions. Therefore, the transition can be activated only in case, if all conditions are fulfilled.

A common classification of Petri Nets distinguishes three levels of Petri Nets:

1. Level 1: Petri Nets are characterized by places, which may contain only boolean value. A places are marked by unstructured tokens. Example of

level 1 nets are Condition/Event systems, Elementary Net systems and State Machine
2. Level 2: A places in the Petri Nets may represent integer value, i.e. a places are marked by several unstructured tokens. Example of level 2 nets are Place/Transition nets and ordinary Petri nets
3. Level 3: A places in the Petri Nets may represent high-level values, i.e. a places marked by a sets of structured tokens. For example Colored Petri Nets (CPNs) and High-Level Petri Nets (HLPNs).

GWorkflowDL is a description language developed as a XML-based language for representation Grid workflows. It is based on High-Level Petri Nets. This language consist of two main part:

1. a generic part, that are used to define the structure of the workflow and this part reflecting data and control flow in the aplpication
2. a platform-specific part, which define how have to be processed a concrete worklow in dependence on a specific Grid computing platform

The root element, which is called *workflow*, includes three types of elements. The element *description* is optional and its content describes workflow in a human-readable form. Further it includes several elements of *places* and *transition*, which define the Petri Nets of the workflow. The places *place* have a attribute ID, futher they may include a element *description* and some elements *token*, which contain data for given place. The transitions *transition* may include element *description*. Further they must include minimal one element *inputPlace*, which attribute placeID refers to specific place in the Petri Nets. Likewise they must include minimal one element *outputPlace* with attribute placeID. The elements *transition* next include element *operation*, which carries information about operation on processing input tokens and to generate output tokens for the output places. Simple workflow from the text-mining domain, which describes a tokenization of some documents can be represented as a Petri Net and includes two places, concrete *parsedDocuments* and *tokenizedDocuments*, and one transition, which is called *tokenizeDocuments*. The place parsedDocuments carries the parsed documents, which serve as input token into transition tokenizeDocuments. Given transition includes an information about service, which from intput parsed documents makes a tokenized documents, and these next assigns to the place tokenizedDocuments.

5 Conclusion

This paper presented an overview of proposed approach to semantic annotation and execution of web services in text mining domain. Our aim was to present the idea of semantic description in OWL-S and to present the corresponding mappings to Petri-nets and BPEL in order to execute the workflows. This paper contains mostly the mivation and idea behind the approach, various implementation aspects will be introduced in the future.

Acknowledgements

The research reported in this paper has been partially financed by Slovak national projects SEMCO-WS APVV-0391-06, APVV RPEU-0024-06, APVV RPEU-0029-06, VEGA 2/6103/6, VEGA 2/7098/27.

References

1. IBM WSDL4J Project, http://oss.software.ibm.com/developerworks/projects/wsdl4j
2. Apache WebServices – Axis Project, http://ws.apache.org/axis/
3. Gubala, T., Bubak, M., Malawski, M., Rycerz, K., Semantic-based Grid Workflow Composition, In: Proc. of 6-th Intl. Conf. on Parallel Processing and Applied Mathematics PPAM'2005, R.Wyrzykowski et.al. eds., 2005, Springer-Verlag, Poznan, Poland
4. GridSphere portal framework, http://www.gridsphere.org/gridsphere/gridsphere
5. Globus Toolkit, http://www-unix.globus.org/toolkit/
6. A. Hoheisel, Grid Workflow Execution Service – Dynamic and interactive execution and visualization of distributed workflows. In Proceedings of the Cracow Grid Workshop 2006, Cracow, 2006.
7. Martin Alt, Andreas Hoheisel, Hans-Werner Pohl, Sergei Gorlatch, A Grid Workflow Language Using High-Level Petri Nets. In: Proceedings of the PPAM05, Poznan, 2005
8. D. Mandell, S. McIlraith, Adapting BPEL4WS for the Semantic Web: The Bottom-Up Approach to Web Service Interoperation. In Proceedings of the Second International Semantic Web Conference (ISWC2003), 2003
9. M.A. Aslam, S. Auer, J. Shen, From BPEL4WS Process Model to Full OWL-S Ontology. Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006), Budva, Montenegro, June 11-14, 2006
10. Antonio Brogi, Sara Corfini, Stefano Iardella, From OWL-S descriptions to Petri nets. Proceedings of the Third International Workshop on Engineering Service-Oriented Applications (WESOA) Wien, Austria 2007
11. A. Hoheisel, User tools and languages for graph-based grid workflows, Proc. Workflow in Grid Systems Workshop in GGF10, Berlin, Germany, March 2004.

Nástroje

SAKE project – Second Prototype of Groupware System

Peter Butka, Gabriel Lukáč

Technical University of Košice, Letná 9, 04001, Košice, Slovakia
{Peter.Butka,Gabriel.Lukac}@tuke.sk

Abstract. In this article we propose the second prototype of a groupware system within the SAKE project. SAKE (Semantic Agile Knowledge-based E-government) is a STREP Project sponsored by the European Union starting in March 2006. The overall objective of SAKE is to specify, develop and deploy a holistic framework and supporting tools for an agile knowledge-based e-government that will be sufficiently flexible to adapt to changing and diverse environments and needs. We give a brief overview of details regarding the global architecture of second prototype particularly of our component and provided features.

1 SAKE Project – Introduction and Conceptual Architecture

Existing approaches for knowledge management in e-government focus mainly on the efficient management of a particular, isolated knowledge resource and on supporting only message-based communication between public administrators. Project SAKE (Semantic Agile Knowledge-based E-government) is a STREP Project sponsored by the European Union starting in March 2006. The overall objective of SAKE is to specify, develop and deploy a holistic framework and supporting tools for an agile knowledge-based e-government that will be sufficiently flexible to adapt to changing and diverse environments and needs [1].

In this article we will concentrate on architecture of the whole system and the second prototype of our groupware component (GWS). Based on the analysis of the problems addressed in e-government domain and needs for applying semantic technologies, we have identified these main SAKE's technological components (Figure 1) [2]:

- *Semantic-based attention (change) management* – ensures a high precision of information delivered to a public administrator by formal and explicit modelling of preferences of a public administrator regarding provided information.
- *Semantic-based content management system* – enables efficient provision of knowledge in the context of a public administrative process.
- *Semantic-based Groupware system* (GWS) – supports more efficient knowledge sharing by developing methods and tools for effective interaction between public administrators and for enabling building community of practice, as well as methods and tools for pushing of knowledge and for searching for experts.

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 85–88, ISBN 978-80-227-3027-3.

– SAKE will also develop a *conceptual framework* for a semantics-enabled agile knowledge-based e-government that will comprise an analysis of the knowledge infrastructure and knowledge sources in e-government. Several ontologies have been developed in order to effectively share information sources, business process details, people resources information, etc.
– *Workflow Management System* (WfMS) – responsible for context awareness and working with business process – based on Jboss BPM.

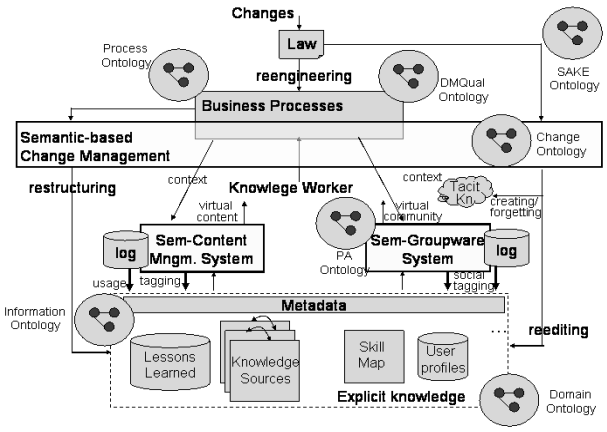


Fig. 1. Architecture of the SAKE system.

3 Second prototype of GWS in SAKE -Implementation

The structure of our groupware system can be described in a sense of four basic layers with several components in particular layers:

– *Back-end Persistence Layer* – this is the back-end storage necessary for supporting GWS actual work and functionalities in storing different types of data. As a persistent middleware layer the Hibernate framework is used.
– *Back-end Business Logic Layer* – the logic behind provided functionalities of the groupware component is covered by this layer, mainly management of a shared workspace and user access control to resources.
– *Tool Functionalities Back-end Layer* – this is the layer responsible for back-end support of different tool functionalities which are provided to the users. Different types of communication are provided in this layer like discussion forums, e-mail or other tools for communication as well as the shared calendar or file sharing, provided by corresponding modules.
– *Interfaces Layer* – this layer is mainly responsible for integration with other SAKE components. It covers connections to services provided by other components as well as services which are provided by GWS for other parts of the whole integrated system.

Implementation of the Groupware System (GWS) in SAKE is based on the open source solution named Coefficient. This system has been selected after a detailed analysis as the best candidate. One of the problems of the selection process was that we have not found any other proper solution supporting all of required functionalities. The most notable example is the shared calendar, which is now implemented with the usage of an external open-source server solution; concretely it is based on the Jakarta Slide WebDAV server with iCal4j Java library supporting the iCal standard. A slightly complicated situation is solved by a creation of the 'GWS Adaptor' which should be an abstraction layer for a combination of systems e.g. Coefficient, calendar server, mail server and so on. Through an adaptor the GWS component is able to communicate with one or more systems realizing required functionalities defined by the adaptor interface. One of the benefits is that SAKE GWS system features (and whole SAKE, because CMS is using the same adaptor mechanism) could be provided also to an environment with existing groupware system. This will help in adopting it in different public administration sites.

4 Second prototype of GWS in SAKE – Features

According to usage of GWS in context of SAKE we have identified several aspects for which semantics and text-mining methods are interesting (within scope of internal public administration processes which are mainly reflected in SAKE). All of them are implemented and fully integrated with other subsystems.

The role of GWS in SAKE can be (more practically) viewed in:

- Management of shared workspace of users in specific process context – forums, shared calendar, mail, etc.
- Providing GWS-related services to other components
- Providing support for GWS-related activities in the workflow
- Providing semantically-enhanced features in order to achieve re-use of knowledge
- Capture the users behavior

Several of these aspects are interconnected directly through the usage of sematic technologies, e.g. capture of user's behaviour in a semantic way leads to a semantically enhanced feature known as semantic log. Then semantically enhanced features within GWS are:

1. *Full support/usage of semantic context within GWS.* It is one of the basic assumptions for semantic-based support in the whole integrated SAKE system. Context is defined according to current user and actual state of the current process instance. This context is then important for preparing of information which is shown to the user and logging of context-specific information into the integrated system (see next feature)
2. *Semantic log of user's activities within GWS.* All activities of users within GWS are semantically logged into ontologies.
3. *Annotation of discussion messages*. Contributions of users in discussion forums (it means discussion messages) are annotated in two different ways:

a) A set of keywords is attached as metadata to every contribution
b) Relevance feedback by users to discussion messages. This means that every user can add a feedback to a message representing user's own opinion of message's relevance to the current problem. It is based on a simple scaled annotation from negative to positive feedback and information is then used in other features like extraction of potential experts.

4. *Metadata search*. In order to achieve benefits from information provision to user metadata search for every subcomponent (functionality) is provided. It means that search is realized as a combination of metadata searches for different information resources like documents, discussion forums, threads, and messages, calendar events, and so on.
5. *Extraction of potential experts by ranking of users according to previous discussions*. Our new method for discussion forums analysis (introduced in [4]) is used for extraction of potential experts. Using the combination of user's feedback and filtering of threads (according to annotation of messages in threads) method provides 'argumentation'-like support solution which leads to feedback-based topic-sensitive ranking of discussion forums users. This information can be used by someone who wants to invite new members to process (e.g. expert) to help to distinguish between several users (and helps in building of Communities of Practice).

5 Conclusion

In this article second prototype of the GWS in SAKE project has been described. We have also presented simple overview of the architecture and some implementation details regarding second prototype of GWS in the project, as well as basic semantically-enhanced features of the system.

The work presented in the paper is supported by the EC within the FP6 IST 027128 project "SAKE – Semantic-enabled Agile Knowledge-based E-government".

References

1. Stojanovic, N., Mentzas, G., Apostolou, D.: Semantic-enabled Agile Knowledge-based e-government. In: *Proceedings of Semantic Web meets eGovernment 2006*, AAAI Spring Symposium Series Stanford University, California (2006)
2. SAKE consortium: *Annex 1: Description of Work*. (2005)
3. Butka, P., Hreňo, J.: Architecture of Groupware System in SAKE. In: Electronic Government – 6th International EGOV Conference: Proceedings of ongoing research, project contributions and workshops, DEXA conferences, September 2007, Regensburg, Germany, pp. 301-308 (2007)
4. Butka, P., Lukac, G.: Semantic-based Groupware System in SAKE for Support of Knowledge and Expert Intensive Public Administration Processes. In Proceedings of CECIIS 2008 conference, September 2008, Varazdin, Croatia (2008)

Sémantická integrácia služieb verejnej správy v projekte Access-eGov

Marek Skokan, Ján Hreňo

Technická Univerzita v Košiciach, Ekonomická fakulta
Letná 9, 042 00 Košice
{Marek.Skokan,Jan.Hreno}@tuke.sk

Abstract. Tento príspevok popisuje prístup ku integrácií služieb poskytovaných inštitúciami verejnej správy v tradičnej alebo elektronickej podobe z pohľadu občana alebo podnikateľského subjektu, ktorý je použitý v projekte Access-eGov. Jednoduché (atomické) služby sú integrované na sémantickej úrovni (s využitím sémantického popisu existujúcich služieb) do scenárov. Realizácia týchto scenárov vedie k riešeniu problému, ktorému čelí používateľ (občan alebo podnikateľský subjekt) v danej životnej situácii (napr. ako získať stavebné povolenie a pod.). Vďaka tomu je používateľovi poskytnutá hodnotnejšia služba pozostávajúca zo scenára služieb a nie iba jednoduchá služba. Druhou výhodou je, že služby v scenári sú personalizované (t.j. sú prispôsobené podľa dát o používateľovi a/alebo jeho situácii). V prípade, že sú niektoré služby v scenári prístupné elektronicky môžu byť spustené on-line. Prvý prototyp Access-eGov systému bol testovaný na troch pilotných aplikáciách v troch EU krajinách.

1 Úvod

V e-Governmente je v súčasnosti jednou z kľúčových výziev sémantická interoperabilita služieb. Tento termín označuje technickú spôsobilosť kooperácie poskytovaných služieb. Interoperabilita je explicitne adresovaná ako jedna zo štyroch hlavných výziev v stratégií EU i2010 [1]. Nevyhnutným predpokladom pre interoperabilitu služieb je integrácia relevantných služieb.

Jedným z najsľubnejších prístupov k integrácií a ku interoperabilite je nasadenie sémantických technológií [2]. Hlavnou výhodou tohto prístupu je jeho spôsobilosť formálne popísať význam a kontext vládnych služieb a to oboch typov, tak tradičných (tzv. „face-to face“ nazývaných tiež „papierové“) ako aj elektronických (poskytovaných ako elektronické formuláre alebo webové služby). Pritom nie je nevyhnutné modifikovať samotnú službu.

Projekt Access-eGov (www.access-egov.org) je výskumno-vývojový (V/V) projekt financovaný Európskou Komisiou v rámci 6. rámcového programu EÚ (6RP), program „Technológie Informačnej Spoločnosti“ (IST). Zámerom projektu je vývoj nástrojov a metodológie umožňujúcich sémantickú integráciu a interoperabilitu služieb verejnej správy (VS) v praktických aplikáciách.

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 89–92, ISBN 978-80-227-3027-3.

2 Metodológie a nástroje v Access-eGov

Hlavným cieľom projektu Access-eGov je poskytnúť občanom a podnikateľským subjektom v ich životných situáciách a podnikateľských udalostiach, platformu s prístupom ku vládnym službám (tak tradičným ako aj elektronickým) v integrovanej podobe. Access-eGov vychádza s konceptu „životných situácií". Prístup na báze životných situácií (*life event approach*) je považovaný za efektívnu a tiež často používanú metódu v na používateľa orientovaných riešeniach pre e-Government [3]. Životnú situáciu možno definovať ako situáciu v živote občana (podnikateľská udalosť – v životnom cykle organizácie), ktorá vyžaduje použitie vládnych služieb. Životná situácia je zvyčajne komplexná a môže byť dekomponovaná na mnoho vzájomne závislých podcieľov. Dosiahnutie týchto podcieľov vedie k riešeniu danej situácie. Takýto podcieľ môže byť dosiahnutý s nejakou množinu vládnych služieb.

Ako konceptuálna základňa bola pre projekt zvolená technológia WSMO (http://www.wsmo.org). Pre modelovanie služieb verejnej správy založenom na prístupe životných situácií vznikla potreba rozšíriť existujúci WSMO konceptuálny model. Preto boli pridané dva nové elementy. Prvým je koncept „Životná situácia", ktorý predstavuje formálny model administratívneho procesu (procesný model) vyjadrenom orchestračnou konštrukciou. Druhým konceptom je „Služba". Tento koncept predstavuje generalizáciu konceptu „Web služba" z WSMO. Jeho rozšírenie umožňuje popísať tak tradičné ako aj elektronické služby verejnej správy.

WSMO špecifikácia pre popis procesných modelov pozostáva z choreografického a orchestračného rozhrania WSMO cieľov založenom na abstraktných stavových automatoch. Bolo zistené, že takéto WSMO choreografické a orchestračné rozhrania nie je vhodné pre prezentovanie procesu a pre aktualizovanie stavu procesu používateľom. Preto bol v Access-eGov vytvorený nový návrh rozhrania, ktorý je založený na paradigme workflow-u. Ten bude rozširovať WSMO špecifikáciu.

Celý proces vytvárania ontológií v projekte, ktorý bol založený na používateľovi riadenom prístupe je popísaný detailne v [2]. Tento proces má 7 krokov. Vytvorené ontológie sú plne lokalizovateľné (t.j. existuje podpora popisov v rôznych jazykoch).

Z pohľadu Access-eGov systému, ontológie môžu byť považované za rozhrania medzi technickou infraštruktúrou a pilotnými aplikáciami. Poskytujú totiž špecifikáciu štruktúry dát používaných v systéme pre systémové komponenty zodpovedné za objavovanie, kompozíciu, mediáciu a spúšťanie služieb [4] (v [4] sa nachádza aj bližší popis architektúry systému).

Ontológie tiež poskytujú základňu pre sémantickú anotáciu služieb verejnej správy. Pričom sémantická anotácia môže byť definovaná ako formálny popis poskytovaných služieb pomocou elementov ontológie. V projekte bol pre potreby anotácie vládnych služieb vyvinutý špeciálny nástroj pre túto činnosť – Anotačný nástroj [5].

2.1 Anotačný nástroj

Anotačný nástroj bol implementovaný ako štandardná webová aplikácia s použitím rozšíreného WSMO objektového modelu a JSF technológie. Nástroj poskytuje sadu formulárov pre špecifikáciu vlastností služieb. Pre uľahčenie údržby pracovných sekvencií pri anotovaní služieb verejnej správy bol implementovaných mechanizmus

preddefinovaných šablón. Anotačný nástroj zahŕňa aj jednoduchý manažment rolí používateľov a viacjazyčnú podporu tak na dátovej úrovni (samotné anotácie) ako aj na úrovni používateľských rozhraní. Navyše Anotačný nástroj umožňuje spájať jednotlivé políčka vo formulároch s informáciami na existujúcich webových stránkach verejnej správy. Toto riešenie dáva možnosť anotovať externé webové stránky a sémanticky integrovať ich obsah do eGovernment aplikácie.

2.2 Klient osobného asistenta

Pre používateľov zo strany občanov bol vyvinutý „klient osobného asistenta“ (Personal Assistant Client - PAC). Tento klient umožňuje browsovanie, objavovanie a spúšťania služieb VS relevantných k špecifickej životnej situácii používateľa. Podobne ako Anotačný nástroj aj PAC bol implementovaný ako webová aplikácia s využitím JSF technológie. Usporiadanie, štruktúra a poradie „tabov“ v rozhraní sú dynamicky generované podľa definovanej ontológie životnej situácie a podľa anotácií služieb a ďalej prispôsobené podľa odpovedí používateľa. Po výbere životnej situácie používateľom sa zobrazí zodpovedajúca navigačná štruktúra pozostávajúca z podcieľov a služieb v podobe textu, hyperliniek a políčok pre vloženie príslušnej hodnoty alebo rozhranie pre spustenie webovej služby. Používateľ má možnosť prechádzať podciele a vložiť odpovede na otázky slúžiace na prispôsobenie scenára jeho špecifickej situácii v rámci vybranej životnej udalosti. Potom systém automaticky odvodí podcieľ a naviguje používateľa k ďalším podcieľom a službám odvodeným z konceptuálneho modelu. Systém môže priamo spustiť elektronickú službu (ak taká existuje) poskytovanú cez štandardizované webové rozhranie. Vďaka PAC používateľ získa všetky dostupné informácie o životnej situácii prispôsobenej jeho špecifickému prípadu a má tiež možnosť spustiť akcie požadované jednotlivými službami potrebné na vyriešenie životnej situácie.

3 Evaluácia pilotných aplikácií

Prvý prototyp Access-eGov platformy bol testovaný na troch pilotných aplikáciách v troch EU krajinách. Evaluácii boli podrobené oba vyvinuté nástroje – AT aj PAC.

AT slúžil počas testovania takmer bez technických problémov. Podľa spätnej väzby bol nástroj relatívne ľahko použiteľný, dokonca bol úspešne používaný aj netrénovanými anotátormi (pracovníkmi VS). Iba málo anotátorov nepovažovalo proces anotácie služieb pomocou tohto nástroja za ľahký. Ako dôvod bola uvádzaná „neintuitívnosť“ používateľského rozhrania AT a chýbajúce informácie požadované pri anotácii. Väčšina týchto problémov bola odstránená už počas skúšok.

PAC bol evaluovaný s využitím on-line dotazníka. Účastníci testovali kvalitu poskytovaných informácií, rýchlosť, štruktúru a rozvrhnutie webovskej stránky ako aj navigáciu a „použiteľnosť“ (usability) systému vo všeobecnosti. Podľa výsledkov nemal PAC uspokojivé používateľské rozhrania. Problematická bola štruktúra, dizajn a navigácia. Uvítaná by bola navigácia „krok za krokom“.

Spätná väzba od používateľov neindikovala žiadne problémy s ontológiami. Korektnosť ontológií poukazuje na to, že použitá metodológia bola vhodná.

Na základe vyhodnotenia prvého testovania pilotných aplikácií sa v súčasnosti (september 2008) končia práce na druhej, vylepšenej verzii systému Access-eGov. Systém bude následne otestovaný v druhej sérii testov koncom roku 2008. Projekt Access-eGov skončí vo februári 2009, keď by mala byť k dispozícii finálna verzia systému Access-eGov, použiteľná pre integráciu ľubovoľných služieb VS (ale v princípe aj v iných aplikačných oblastiach).

4 Záver

V príspevku bol prezentovaný prístup použitý v projekte Acces-eGov pre sémantickú integráciu vládnych služieb. Základňu umožňujúcu spojenie Access-eGov technickej infraštruktúry a pilotných aplikácií tvoria ontológie. Vyvinutá metodológia umožnila úspešné vytvorenie týchto ontológií. Boli vyvinuté webové aplikácie tvoriace grafické rozhrania pre používateľov tak zo strany občanov (PAC) ako aj zo strany pracovníkov VS (AT). Tieto boli podrobené testovaniu na pilotných aplikáciách v troch EU krajinách v rámci prvej sérii testov. Tieto testy ukázali, že AT je vyhovujúcou aplikáciou na anotovanie vládnych služieb. Výsledky testov PAC poukázali na niektoré nedostatky. V súčasnosti sú už výsledky testov zapracované do druhej verzie systému Access-eGov, ktorá bude čoskoro testovaná v druhej sérií testov.

Poďakovanie

Tento článok je podporovaný projektom FP6-2004-27020 Access-eGov (Access to e-Government Services Employing Semantic Technologies) a projektom číslo 1/4074/07 Methods for annotation, search, creation, and accessing knowledge employing metadata for semantic description of knowledge.

Referencie

1. i2010 - A European Information Society for growth and employment, COM (2005) 229 final of 1 June 2005, http://ec.europa.eu/i2010
2. Jochen Scholl, Ralf Klischewski, E-Government Integration and Interoperability: Framing the Research Agenda, in: International Journal of Public Administration, Vol. 30, Issue 8-9, 2007, pp. 889-920
3. Anamarija Leben, Mirko Vintar, Life-Event Approach: Comparison between Countries, in: Electronic Government, Springer LNCS 2739, 2003, pp. 434-437
4. Access-eGov Platform Architecture. Deliverable D3.1, Access-eGov Project, 2006. Available at http://www.accessegov.org/acegov/uploadedFiles/webfiles/cffile_10_17_06_9_41_59_AM.pdf
5. Karol Furdík, Ján Hreňo, Tomáš Sabol, Conceptualisation and Semantic Annotation of eGovernment Services in WSMO, in: V. Snášel (ed.), Proc. of Znalosti (Knowledge) 2008. STU, Bratislava, Slovakia, 2008, pp. 66-77

Combination of Curated and Collaborative System for Disease Determination

Pavol Jasem Jr.[1], Ján Paralič[1], Marek Dudáš[2], Saskia Dolinská[2]

[1] Department of Cybernetics and Artificial Intelligence
Faculty of Electrical Engineering and Informatics
Technical University of Košice, Slovakia
{Pavol.Jasem.Jr,Jan.Paralic}@tuke.sk

[2] Institute of Biological and Ecological Sciences
Faculty of Sciences, Pavol Jozef Šafárik University in Košice, Slovakia
{Marek.Dudas,Saskia.Dolinska}@upjs.sk

Abstract. This article describes an application for disease determination by selecting or entering its symptoms in a simple way. The most important biomedical information free source online, NCBI website, provides too high complexity to be used by biologists or physicians. Together with NCBI web service, both offer inconsistent or insufficiently categorized data, with many typographical errors. These are projected into long-lasting incorrect results, still not revised. There is a strong need for existence of a system able to elementarily allow users to state their simple queries to stored data, and to display retrieved data in a desired view. A combination of present curated and collaborative approach proposed in this article will significantly improve NCBI datasets.

Keywords: disease, defect, syndrome, congenital, hereditary, OMIM, clinical, synopsis, decision, determination, curated, collaboration.

1 Introduction

There are many free online biomedical information sources of various diseases and their details. Since 2004, U.S. National Institutes of Health started to provide all articles funded by NIH free of charge on their website [1]. Articles saved to PubMed database are available, as well as individual facts extracted from them, which are transcribed to other databases. Those are available in form of information with simple relations among them. For example, if an article describes new inventions or a new discovery regarding a hereditary disease and a list of related clinical synopses (symptoms), those are transcribed and saved to OMIM database, a database of congenital syndromes, defects and diseases. A user can find a reference to cited publication to most of this information.

Even though the editors from NCBI add new articles and their details to the databases on daily basis, there are still many publications to transcribe, and not only the newest ones. There is an evident need for a collaborative extension of the existing

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 93–96, ISBN 978-80-227-3027-3.

information. Proposed system described in this article allows also addition of non-curated (scientifically fully not confirmed) information, by community of physicians, and researchers from fields of biology, genetics and medicine. The system created by NIH, which offers information on hereditary diseases, contains tremendous amount of useful information. Unfortunately, it is exposed to user in confusing manner, even from a computer scientist's point of view.

2 Searching NCBI for diseases

Physicians, biologists, geneticists, and often common people need to find a disease by observed symptoms, or more diseases, a physician didn't take into consideration. Let's have an example: We want to find all known alternative syndromes or defects, where some clinical synopses are know. This is rather important ability, which a system with such data as OMIM should possess. However, the NCBI web application or web service provides only very restricted searching service in clinical synopses. This is the question we state:

Find a list of diseases those are likely to cause following observed symptoms: cleft palate, mental retardation, and hearing loss.

We could have used a full text search on OMIM website, and the search string would be:

```
"cleft palate"[cs] "hearing loss"[cs]
"mental retardation"[cs]
```

with [cs] meaning the attribute of search: Clinical synopsis. By submitting this search, we get 35 results. But the scientists demand all possible results, and we missed two syndromes. In one of them, result 117650, the clinical synopsis is "Cleft soft palate". In 122470, it's "Cleft lip/palate". It would be even worse if we entered "palate cleft" instead. The reason is simple: multiple words put in quotation marks are found, case-insensitive, only in case of exact form as they are quoted. Another option is to separate all words:

```
cleft[cs] palate[cs] hearing[cs] loss[cs] mental[cs]
retardation[cs]
```

For the NCBI system, it means to find all records with these words located anywhere in clinical synopses. This is not the way we could retrieve relevant results, as the words can be combined through all synopses categories in any way. Changed words could mix up each meaning, for example when searching for "Asymmetric breast, Absent fingernails, Short stature" as present in record number 305600.

Another way to retrieve desired results is to use Entrez Utilities. We can run three searches on OMIM database for connected search strings (originally in quotation marks), then programmatically compare these three results, and select the records present in all of three search results. This still doesn't resolve the mixing problem: one of results when searching for "Missing toes" can be a result e.g. with clinical

synopses “Missing fingernails” and “Widely spaced first and second toes”, even if it shouldn’t be.

Yet, apart from this, the cleft lip/palate result is missing when searching for “palate”. Full text search in NCBI application treats a slash as a connector of words into one.

3 Resolution

System we are working on uses own search engine, which doesn’t exclude slash-separated words and it can handle multiple clinical synopsis descriptions, enabling it to provide more exact filtering while considering individual descriptions. Unlike the original NCBI system, our database stores the synopsis descriptions in a tree topology, named and generated by original clinical synopsis keys in OMIM database. Thus, the application allows a user to narrow the search results by selecting displayed categories in a tree view, with a note consisting of sum of records in each category.

Data in the developed database are kept and provided unchanged with a respect to NIH authorship. Compared to NCBI system, only search methods and user-friendly interface are improved. The origin of data guarantees verity of information, but also a collaborative space is going to be created, where all users will be registered as members of various research centers. A user can decide whether to add or not another user and/or user to trusted zone and whether to display or not these non-curated information.

Acknowledgements

This vision is being realized thanks to support from “Genomic Data Mining on Birth Defects (GEMIN)” Slovak Ministry of Health project (2007/65/UPJŠ-02), “PoZnaŤ - Support for Knowledge Creation Processes” Slovak Research and Development Agency project (RPEU-0011-06), “Information Extraction and Knowledge Discovery for Support of Hereditary Diseases Research (Eco-Net)” International Research Cooperation project (Fr/ČR/SR/TUKE07), and “Methods for Knowledge Annotation, Searching, Creation and Accessing, Using Metadata for Semantical Knowledge Description” VEGA Slovak Grant Agency project (VEGA 1/4074/07).

References

1. Zerhouni E., A. “NIH Public Access Policy,” Science, vol. 306, pp.1895, 2004.
2. OMIM database. NCBI HomePage. [Online] NCBI. [Accessed: April 23, 2007.] http://www.ncbi.nlm.nih.gov/omim.
3. NCBI Web Service. NCBI HomePage. [Online] NCBI, February 2, 2007. [Accessed: April 24, 2007.] http://eutils.ncbi.nlm.nih.gov/entrez/query/static/esoap_help.html.

4. Becquet, C., Blachon, S., Jeudy, B., Boulicaut, J., F., Gandrillon, O. "Strong-association-rule mining for large-scale gene-expression data analysis: a case study on human SAGE data," Genome Biol, vol. 3, no. 12, 2002.
5. Velculescu, V., E., Zhang, L., Vogelstein, B., Kinzler, K., W. "Serial analysis of gene expression," Science, vol. 270, pp. 484-487, October 1995.
6. Kléma, J., Soulet, A., Crémilleux, B., Blachon, S., Gandrillon, O. "Mining Plausible Patterns from Genomic Data." In Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems (June 22-23, 2006). CBMS. IEEE Computer Society, Washington, DC, pp. 183-190, 2006.

Monitorovanie toku služieb v prostredí GRID

Martin Krajč[1], Peter Bednár[1], Tomáš Kasanický[2]

[1] Centrum pre informačné technológie, Technická univerzita, Košice
krajc.martin@gmail.com,peter.bednar@tuke.sk

[2] Ústav informatiky, Slovenská akadémia vied, Bratislava
tomas.kasanicky@gmail.com

Abstrakt. Táto práca popisuje jeden z možných prístupov ako v prostredí grid-u monitorovať beh služieb. Monitorované služby sú výsledkom kompozície a následného spustenia toku služieb. Získané údaje sú neskôr použité pre skvalitnenie procesu kompozície a rozhodovania. Prístup je založený na technológii Java JMX.

Kľúčové slová: monitoring, work-flow, Java JMX

1 Úvod

Automatická a semi-automatická kompozícia toku služieb sa stáva v súčasnosti stále častejšou témou mnohých prác. Patrí medzi základné potreby vyplývajúce z prostredia webových služieb. Kompozícia webových služieb je náročný proces z hľadiska samotného rozhodovania a plánovania. Z hľadiska správneho naplánovania a vyskladania toku služieb je potrebné, aby každá služba bola dôsledne popísaná. Za predpokladu, že táto podmienka je splnená, môžeme uvažovať o posudzovaní kvality jednotlivých služieb, čo dopomôže skvalitniť proces rozhodovania. Pod pojmom kvalita služby je myslená ich výkonnosť a často aj schopnosť vykonať úlohu, ktorá je deklarovaná v popise v konečnom čase. Pre získanie týchto parametrických opisov jednotlivých služieb je nutné pri ich behu monitorovať priebeh ich funkčnosti. Táto práca popisuje jeden z možných prístupov ako v prostredí grid-u monitorovať beh služieb, ktoré sú výsledkom kompozície a následného spustenia toku služieb, zo zámerom neskoršieho skvalitnenia procesov kompozície a rozhodovania. Tomuto problému bolo venované viacero projektov a publikácií (FP6: EGEE[3], DEISA[2], KWFGrid[7]).

2 QoS ontológia

Pri automatickej kompozícií pracovných tokov je potrebné vybrať službu, ktorá je schopná poskytnúť požadované výstupy a dosiahnuť daný cieľ. Okrem funkčných vlastností služby, ktoré určujú za akých podmienok služba dokáže poskytnúť

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 97–100, ISBN 978-80-227-3027-3.

požadované výstupy, ako sekundárne kritérium výberu služby je možné použiť rôzne indikátory kvality. Samotný proces automatického výberu služby pri plánovaní potom prebieha v dvoch krokoch:

- porovnanie funkčných vlastností služby, tj. či sú logicky splnené všetky podmienky pre to, aby daná služba poskytla požadované výstupy a
- porovnanie indikátorov kvality, na základe ktorého sa dodatočne odfiltrujú a preusporiadajú kandidáti z predchádzajúceho kroku.

Po týchto krokoch je možné automaticky a jednoznačne vybrať potrebnú službu. Porovnávanie si však vyžaduje, aby oba typy vlastností, tj. funkčné vlastnosti a indikátory kvality boli vhodne konceptualizované tak, aby sa umožnilo ich automatické porovnávanie a odvodzovanie. Ako konceptuálny model indikátorov kvality sme navrhli ontológiu, ktorá je uvedená na obrázku 1. Ďalšie delenie metrík je podľa entity, ktoré popisujú:

- *ActivityMetric,* popisuje atomickú aktivitu pri volaní služby (aktivita zodpovedá volaniu metódy pri daných vstupných dátach na konkrétnom uzle v gridovom prostredí)
- *ServiceMetric,* popisuje agregované indikátory pre celú službu získané agregovaním metrík aktivít a normovaním podľa metrík zdrojov. Umožňuje porovnávanie služieb podľa QoS.
- *ResourceMetric,* popisuje zdroje použité pri volaní služby (zodpovedajú kontextu pri spúšťaní aktivít), delia sa na metriky pre popis výpočtového uzla (NodeMetric) a na metriky pre popis vstupov/výstupov služby (tj. funkčných parametrov pri volaní služby).

3 Architektúra pre monitorovanie grid-ového prostredia

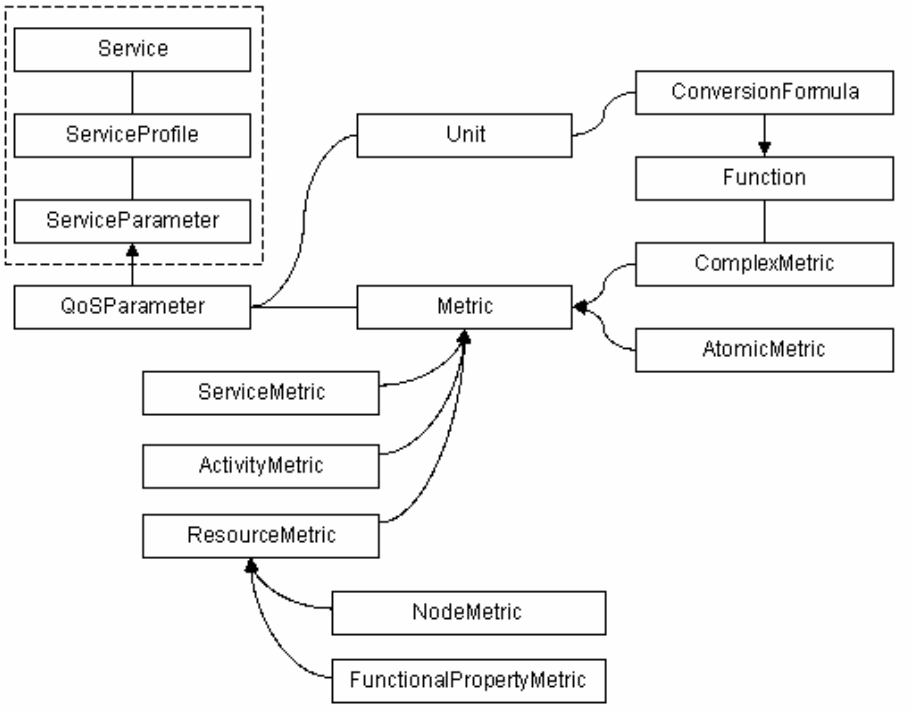
Navrhovaný spôsob sa opiera o nepriame posudzovanie kvality spustených služieb. Pomocou sledovania zaťaženia výpočtového zdroja, v ktorom sa vykonáva daná služba, je posudzovaná jej kvalita. Pre spúšťanie toku služieb je využívané prostredie GWES (Grid Workflow Execution Service)[4]. GWES na základe svojho vstupu, ktorý tvorí popis toku služieb, spúšťa jednotlivé služby v danej náväznosti a postupnosti. Vstup je definovaný pomocou GWorkflowDL[5] (Grid Workflow Description Language). Jedná sa o XML schému, ktorá pozostáva z popisu aktivít, jednotlivých spúšťaných služieb a ich parametrov. Za predpokladu, že jednotlivé webové služby sú napísané v jazyku Java tak ako aj GWES, môžeme proces monitorovania zúžiť na sledovanie stavu JVM (Java Virtual Machine). Pre tento účel existuje v jazyku Java technológia JMX[6] (Java Management Extension). Podobný prístup bol použitý aj v práci [1].Táto technológia je založená na tzv. Mbeanoch, tj. Entitách, ktoré sprístupňujú namerané metriky vzdialeným klientom. Jediné čo je nutné dodržať pre správnu funkčnosť v heterogénnom prostredí, je nastaviť JVM tak, aby podporoval JMX aj s možnosťou vzdialeného prístupu. Celý manažment registrácie Mbeanov a monitorovania prostredia je potom možné riadiť z klientskej aplikácie.

Monitorovanie je rozdelené na monitorovanie jednotlivých aktivít, ktoré sú spúšťané v prostredí GWES a na monitorovanie zdrojov, ktoré vykonávajú volané služby. Na sledovanie aktivít bol do prostredia GWES integrovaný modul, ktorý sprístupňuje jednak informácie o behu jednotlivých služieb z pohľadu prostredia GWES, a jednak agreguje monitorované informácie získané z jednotlivých výpočtových zdrojov, ktoré boli použité pri vykonávaní služieb zaradených do pracovného toku. Medzi monitorované údaje v prostredí GWES patrí:

- detailné informácie o využití pamäte JVM počas vykonávania požiadavky na danú službu
- detailné informácie o procesoch v JVM, ktoré umožňuje nepriame meranie zaťaženia procesora/procesorov na danom výpočtovom zdroji

Súčasťou systému monitorovania je klientská aplikácia, ktorej úlohou je ukladanie monitorovaných údajov pre ďalšie spracovávanie. Táto aplikácia využíva štandardné vzdialené pripojenie JMX v prostredí GWES, pomocou ktorého zozbiera informácie o behu jednotlivých aktivít pracovného toku a o výpočtových zdrojoch využívaných pri jeho vykonávaní. Tieto údaje sú potom uložené v štandardnej relačnej databáze, odkiaľ sú po agregovaní použité na odvodenie jednotlivých inštancií metrík.

Schematické znázornenie celej funkcionality implementovaného riešenia je zakreslené na obrázku 2.

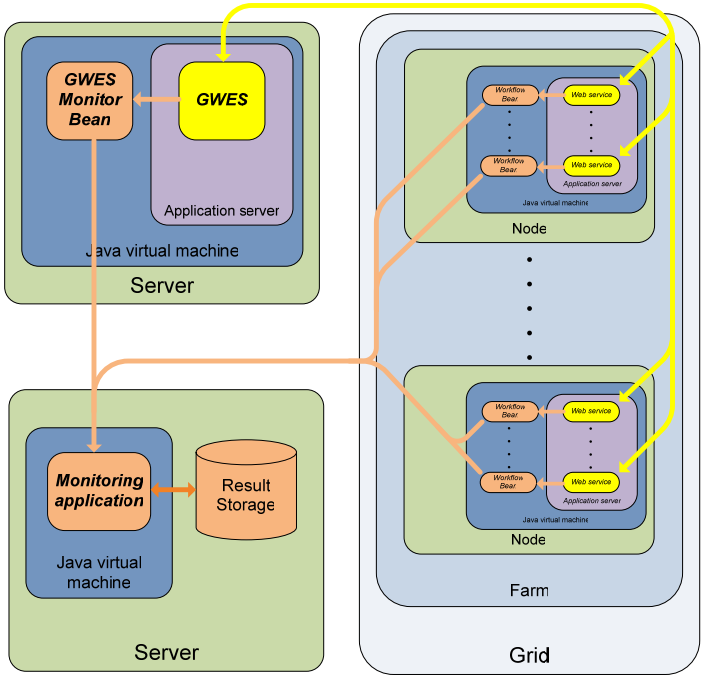


Obr. 1. QoS ontológia.

4 Zhodnotenie

Navrhovaná architektúra umožňuje nepriame sledovanie spúšťaných aktivít ako aj sledovanie programu GWES. Na základne získaných atribútov je možné zdokonaliť procesy rozhodovania a kompozície pri návrhu pracovného toku služieb.

Poďakovanie. Výskum prezentovaný v tomto článku bol čiastočne financovaný SEMCO-WS APVV-0391-06.



Obr. 2. Architektúra navrhnutého systému.

Referencie

1. Balos, K., Radziszowski, D., Rzepa, P., Zieliński, K., Zieliński, S.: JIMS Extensions for Resource Monitoring and Management of Solaris 10. In: TASK Quarterly: scientific bulletin of Academic Computer Centre in Gdansk 2004. LNCS, vol. 8,No 4, pp. 487-501. ISSN: 1428-6394
2. DEISA, http://www.deisa.eu/
3. EGEE, http://public.eu-egee.org/
4. GWES http://www.gridworkflow.org/kwfgrid/gwes/docs/apidocs/net/kwfgrid/gwes/GWES.html
5. GWorkflowDL, http://www.gridworkflow.org/kwfgrid/gworkflowdl/docs/
6. Java JMX, http://java.sun.com/javase/technologies/core/mntr-mgmt/javamanagement
7. KWFgrid, http://www.kwfgrid.eu//

Towards Automated System-Level Service Compositions

Pavol Mederly

Faculty of Informatics and Information Technologies
Slovak University of Technology
Ilkovičova 3, 842 16 Bratislava 4, Slovak Republic
mederly@fiit.stuba.sk

Abstract. Service composition in systems developed under service-oriented paradigm presents issues at the business as well as at the technical (system) level. Although business-level concerns will probably have to be solved by human participants for the foreseeable future, system-level ones possibly could be handled automatically. This paper presents an early research-in-progress report in this area: a problem statement, related work, and some ideas on future research.

Keywords: service-oriented computing, service composition, microflow, enterprise service bus.

1 Introduction

Service-oriented computing is a modern paradigm for creating distributed applications, especially in case of integrating existing (legacy) systems.

Services, as basic parts of such applications, can be connected in many ways. Typically, one (composite) service calls other (simpler) ones. Other means of service communication can be, for example, an event-based one when a service sends event messages to one or more other services, potentially not known in advance.

When creating a composite service from simpler ones many issues have to be solved. Some of these are concerned with the *business logic* of the composition, e.g. when creating a service to process an order (in a reselling company) we need to call "order validation" service first, then "check customer credit", and, depending on the result, to continue with order processing or return an error message to the client. Currently this business logic is, and very probably will continue to be, designed by human users, preferably by business analysts in cooperation with business owners.

Another kind of issues to be solved when creating a composite service can be characterized as *technical*, or *system-level*, ones: these are concerned with differences in transport protocols, application programming interfaces (APIs), message format and syntax, security requirements, reliability and performance properties, logging and auditing requirements, etc. Resolution of these issues is driven by requirements and capabilities of participating services, by technical infrastructure available, and by business requirements (partially embodied in business logic as described above). Although very powerful tools in this area have appeared recently, e.g. those grouped

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 101–104, ISBN 978-80-227-3027-3.

under umbrella term "Enterprise Service Bus" [2, 8], technical issues are currently being dealt with by people, mostly IT specialists.

Our proposition, stemming out from [6][1], is that system-level issues can be partially or wholly resolved automatically, based on business requirements and service composition at a business level.

1.1 An Example

Let us illustrate the idea on the following simple scenario: a composite service *C* wants to make use of services *A* and *B*, but:

1. *C* makes requests and receives responses using HTTP while *A* is accessible through Java Message Service (JMS) and *B* is accessible through SOAP over HTTP (an example of transport protocol difference);
2. *C* expects to get data in the form of XML while *A* sends its response as "comma-separated values" formatted text (an example of message format difference);
3. *B* and *C* protect their communication using standard SSL (Secure Socket Layer) technology, while *A* utilizes per-message encryption as a feature of JMS provider (an example of security requirements difference);
4. *C* requires that *A* and *B* are contacted with exactly-once message delivery semantics (a reliability property);
5. *C* sends its request in accordance with enterprise's canonical data model, while *A* expects to receive the data according to its own data model, differing e.g. in the naming of data elements, coding of values – like M/F vs. 1/2 for gender, etc. (examples of message syntax differences).

1.2 Problem Statement

We can state the following research problems:

1. How to specify technical (system level) properties of services provided and services requested in a way that is expressive enough yet practical to use?
2. How to match between these properties? I.e., to what extent is it possible to generate, automatically or semi-automatically, a code (e.g. in Java or in configuration language of a specific tool, for example Enterprise Service Bus) that will do the required mediation?
3. What level of adaptivity can be achieved? I.e., what kind of changes can the system cope with automatically?

2 Related Work

Enterprise Service Bus (ESB) [2, 8] provides a powerful infrastructure for service composition at the technical (or microflow, see [4]) layer. Technical differences, e.g. those described in section 1.1, are dealt with using so called mediation services.

[1] Please see "*Business-driven automated compositions* Grand Challenge".

Unfortunately, all implementations we know of as yet require the integration architect to define all the mediation services statically (even though authors of [8] mention some automatic matchmaking capabilities as a feature of an ESB).

2.1 Specification of Properties of Services Provided and Services Required

There are relevant Web Services (WS-*) standards in this area:

1. Web Services Description Language (WSDL, [1]) is used to describe abstract functional aspects of a service (e.g. content of input and output messages) along with some of technical aspects (e.g. transport protocol binding).
2. Web Services Policy Framework (WS-Policy, [9]) provides a general tool for describing capabilities and requirements of the services. Subsequent specifications would define domain-specific policy assertion languages e.g. in the area of security or reliable messaging.

Although these standards are specified primarily for XML Web Services, they can be easily extended to use with services implemented by other technologies as well.

These standards, mainly the first one, are broadly used to describe services *provided*. Their use for description of services *required* is not very common yet.

Schmidt et al. [8] describe a conceptual model of meta-data repository for an Enterprise Service Bus that would meet most (perhaps all) of the above mentioned requirements. Service requestors as well as service providers would publish relevant information as part of required and supported service interfaces, respectively, in the form of *policy declarations*.

There are also a couple of UML profiles designed for specifying functional and non-functional properties of services, e.g. [10].

2.2 Automatic Matching

Lee et al. [5] propose a service orchestration system that is able to function in the heterogeneous environment consisting of web services as well as legacy enterprise applications using technologies like Java RMI, EJB and CORBA. They have created a toolkit that automatically maps between these (and possibly other) technologies. This is an example of resolution of transport protocol and API differences.

Similarly, Gong et al. [3] have created a system for dynamic access to heterogeneous data resources, with mapping between generic access API and resource-specific API defined in the form of ontology. Again, this can be seen primarily as a resolution of a transport protocol and API differences.

Wada et al. [10] provide a mapping from models created using their UML profile into specific technologies, namely MuleESB – an open-source ESB implementation. (This is not a fully automatic matchmaking, as the designer has to specify the mapping, although at rather abstract level.)

The (semi-)automatic resolution of message syntax differences is a well researched problem; see for example [7]. We do not expect to contribute directly to this area, but appropriately use existing results (and tools) instead.

3 Some Ideas on Future Research

After finding suitable formalism to describe system-level properties of services provided and requested we plan to concentrate on the problem of their automatic or semiautomatic matching. We expect to define a set of primitive mediation elements (or possibly more sets, each specific to one target platform, e.g. ESB implementation), and describe their semantics. Then we would try to develop algorithms for resolving mismatches in service interfaces by a sequence of mediation elements.

References

1. Booth, D., Liu, C.K.: Web Services Description Language (WSDL) Version 2.0 Part 0: Primer. W3C Recommendation (2007)
2. Chappell, D.A.: Enterprise Service Bus. O'Reilly Media, Inc., Sebastopol, CA, USA (2004)
3. Gong, P., Gorton, I., Feng D.D.: Dynamic Adapter Generation for Data Integration Middleware. In: SEM '05: Proceedings of the 5th international workshop on Software engineering and middleware, pp. 9-16. ACM, New York, NY, USA (2005)
4. Hentrich, C., Zdun, U.: Patterns for Process-Oriented Integration in Service-Oriented Architectures. In: Proceedings of 11th European Conference on Pattern Languages of Programs (EuroPlop 06), pp.141--189 (2006)
5. Lee, Y., Kim, S., Min, D.: A Hybrid Service-based Orchestration System for Composition of Legacy Services. In: Proceedings of the Fifth International Conference on Grid and Cooperative Computing Workshops, pp.219-226. IEEE Computer Society, Los Alamitos, CA, USA (2006)
6. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F., Krämer, B.J.: Service-Oriented Computing Research Roadmap, In: Cubera, F., Krämer, B.J., Papazoglou, M.P. (eds.) Dagstuhl Seminar Proceedings 05462. Internationales Begegnungs-und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany (2006)
7. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. The VLDB Journal 10, pp.334-350. (2001)
8. Schmidt, M.-T., Hutchison, B., Lambros, P., Phippen, R.: The enterprise service bus: making service-oriented architecture real. IBM Systems J. 44, pp.781-797. (2005)
9. Vedamuthu, A.S. et al.: Web Services Policy 1.5 – Framework. W3C Recommendation (2007)
10. Wada, H., Suzuki, J., Oba, K.: Modeling Non-Functional Aspects in Service Oriented Architectures. In: IEEE International Conference on Services Computing (SCC'06), pp.222-229. IEEE Computer Society, Los Alamitos, CA, USA (2006)

Bezpečné vykonávanie procesov prostredníctvom mobilného kódu v nedôveryhodnom distribuovanom výpočtovom prostredí

Zoltán Balogh, Ivana Budinská, Branislav Šimo

Ústav informatiky, Slovenská akadémia vied
Dúbravská cesta 9, 845 07 Bratislava, Slovensko
{Zoltan.Balogh,Ivana.Budinska,Branislav.Simo}@savba.sk

Abstrakt. V príspevku popisujeme problematiku bezpečného vykonávania procesov v nedôveryhodnom distribuovanom výpočtovom prostredí. Nastolený problém sa navrhuje riešiť prostredníctvom mobilného kódu, ktorý by bol vykonávaný v dôveryhodnom hardvérovom module pripojiteľnom k ľubovoľnému výpočtovému zdroju, ktorý explicitne považujeme za nedôveryhodný. Mobilný kód navrhujeme riešiť prostredníctvom mobilných webových služieb založených na návrhových vzoroch z oblasti multiagentových systémov. Príspevok je písaný v kontexte aplikačnej oblasti, v ktorej takáto potreba vznikla, a to bezpečného vykonávania procesov pre riadenie krízových situácií. Uvádzame aj návrh architektúry pre nedôveryhodné distribuované prostredie pre ktoré bezpečné vykonávanie procesov riešime. Stručne sa zaoberáme aj rozdielmi nášho prístupu od existujúcich prístupov podobného zamerania.

Kľúčové slová: mobilný kód, agenti, webové služby, trusted platform module

1 Úvod

V súčasnosti stále existujú oblasti bezpečnosti dát a výpočtov, ktoré nie sú dostatočne riešené. Tieto problémy súvisia hlavne s distribuovaným charakterom vykonávania výpočtov a spracovania dát. V oblasti bezpečnosti sú riešené viaceré úlohy dostatočne, napr. bezpečné prenosové kanály alebo autentifikácia a autorizácia prostredníctvom asymetrických šifier. Pri tých problémoch, ktoré ešte dostatočne riešené nie sú, je potrebné riešiť viacero vzájomne závislých bezpečnostných úloh, ktoré sa dajú rozdeliť na dve hlavné triedy: zabezpečenie utajenia (Privacy) a zabezpečenie dôvery (Trust). Obidve tieto bezpečnostné triedy sa zaoberajú bezpečnosťou jednak dát a jednak kódu, a to tak na strane klienta (iniciátora), ako aj na strane servera (vykonávateľa). V kontexte tohto príspevku sa venujeme zabezpečeniu utajenia a dôvery tak dát, ako aj výpočtu na strane vykonávateľa – servera, ktorá je podrobnejšie rozpísaná v časti 2.

Komunikácia, bezpečnosť a dostupnosť informácií je kľúčovým faktorom pri riešení a riadení krízových situácií. Zabezpečenie komunikácie predstavuje

P. Návrat and V. Vranić (Eds.): WIKT 2008, pp. 105–109, ISBN 978-80-227-3027-3.

technologický problém, ktorý je potrebné riešiť veľmi komplexne. Je potrebné riešiť jednak prepojenie viacerých komunikačných kanálov a jednak ochranu pred náhodným alebo cieleným zneužitím informácií a komunikačných tokov. V príspevku je prezentovaná architektúra distribuovaného systému pre bezpečné vykonávanie mobilného kódu na báze agentov v nedôveryhodnom výpočtovom prostredí prostredníctvom dôveryhodnej bezpečnostnej hardvérovej platformy práve pre aplikačnú oblasť riadenia krízových situácií.

2 Bezpečné vykonávanie kódu v nedôveryhodnom prostredí

Tabuľka 1 sumarizuje základné výzvy spojené s nedôveryhodným prostredím na strane servera a vykonávaním kódu v ňom:

Tabuľka 1. Bezpečnostné výzvy zabezpečenia dôvery a utajenia dát a kódu v nedôveryhodnom prostredí na strane vykonávateľa – servera.

	Dáta	**Programový kód**
Trust (T)	Zabezpečiť, že server bude pracovať práve s tými dátami, ktoré mu klient pošle.	Zabezpečiť, že server spustí ten programový kód, ktorý mu klient pošle. Sem patri aj ochrana pred modifikáciou a poškodením vykonávaného kódu.
Privacy (P)	Zabezpečiť, že dáta poslané serveru na spracovanie nebude vedieť server identifikovať a interpretovať.	Zabezpečiť, že server nebude schopný vykonávaný kód zneužiť (interpretovať), resp. znova použiť na ten istý účel, na ktorý ho použil klient. Možné ochrany: podpísaný kód, kryptovanie, vykonávanie po častiach a pod.

Vo všeobecnosti je zložitejšia práve situácia pre zabezpečenie T a P na strane servera. Jedným zo spôsobov ako vytvoriť z nedôveryhodného prostredia dôveryhodné na strane vykonávateľa je použitie externého zariadenia, tzv. dôveryhodného bezpečného modulu, ktorý dokáže na seba prevziať kľúčové úlohy súvisiace s bezpečnosťou, dôverou a utajením tak výpočtov, ako aj spracovávaných dát.

Jedným z existujúcich prístupov je použitie tzv. Trusted platform module (TPM) [1]. TPM je hardvérové riešenie – čip, v ktorom sú uložené bezpečnostné prvky zariadenia, na ktoré je pripojené (PC, server alebo mobilné komunikačné zariadenie), pričom toto zariadenie je pripojené na sieť. Bezpečnostné prvky zahŕňajú heslá, certifikáty a kryptovacie kľúče. TPM sa tiež používa na autentifikáciu a atestáciu danej platformy, čo je nutná podmienka bezpečného počítania vo všetkých prostrediach. Širokému aplikovaniu TPM, ako bezpečnostného prvku sieťového počítania bránia viaceré špecifické požiadavky na zachovanie súkromia. TPM nekontroluje softvér, ktorý beží na kontrolovaných platformách. V TPM môže byť uložené konfiguračné parametre, ale aké politiky budú použité na tieto parametre musí ošetriť samostatná aplikácia.

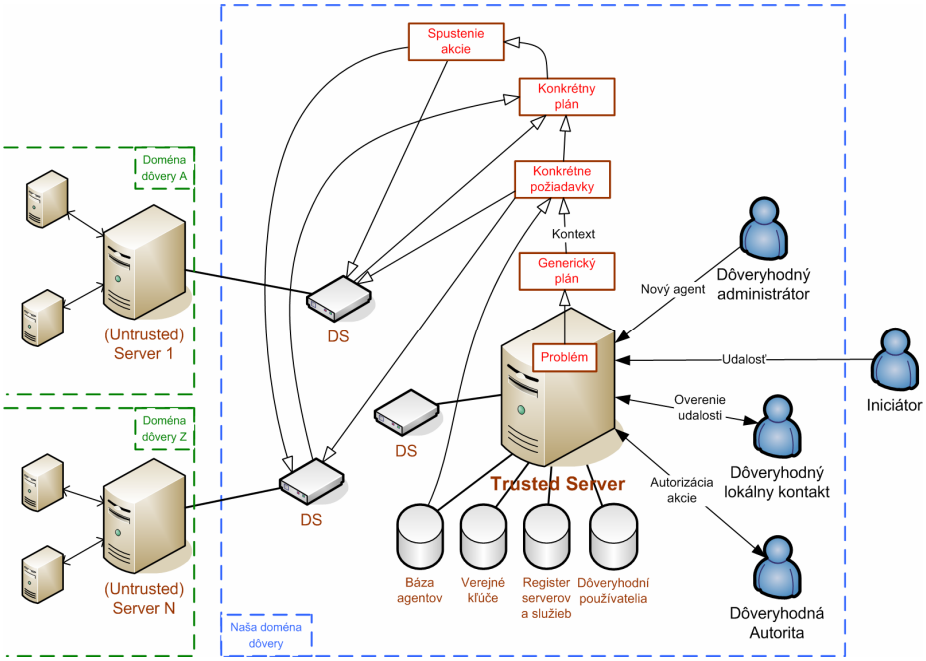
Existuje teda priestor na rozšírenie a zlepšenie modulu TPM, a to vytvorením výpočtového modulu, ktorý by dokázal na seba prevziať aj výpočtovo náročnejšie

operácie. Navrhované riešenie pomocou bezpečnostného čipu by malo riešiť bezpečnosť v rámci konkrétnej siete pripojených zariadení. V rámci DS by boli okrem bezpečnostných prvkov uchovávané aj bezpečnostné politiky. Výhodou navrhovaného riešenia je aj pripojiteľnosť na zabezpečovaný hardvér pomocou zvoleného rozhrania (napr. USB). Navrhované zariadenie by malo byť v súlade so štandardmi, ktoré vydáva Trusted Computing Group (TCG) [2]. Základnými požadovanými funkciami takéhoto bezpečného modulu sú: *generovanie asymetrických kľúčov*, ktoré sa nedajú nijak scudziť ani zmeniť, *autentifikácia a autorizácia* na oboch stranách – klienta aj servera (paralelná autentifikácia), *šifrovanie a dešifrovanie dát*, ukladanie dát, *zabezpečenie bezpečného vykonávania kódu* napr. formou virtualizácie, *priame vykonávanie zadaných funkcií* a *tvorba a udržiavanie bezpečnostných politík*.

Takýto modul by mal byť tiež schopný dynamicky spúšťať rôzne kódy, a to spôsobom bezpečným, dôveryhodným a v požadovanej miere utajenia. Na tento účel plánujeme použiť poznatky z technológie mobilných agentov. Samotná implementácia však bude založená na technológii webových služieb, pričom mobilita je v tomto kontexte pre nás najdôležitejšia vlastnosť. Využívanie mobilných agentov má oproti klasickým agentom niekoľko výhod. Medzi najväčšie výhody patria: schopnosť vykonávať sa na serveri, kde sú uložené dáta – redukuje prenos veľkých objemov dát po sieti, asynchrónne vykonávanie na viacerých heterogénnych serveroch zapojených do siete, adaptabilita – prispôsobenie sa stavu a prostrediu vykonávacieho servera, tolerancia voči chybám v sieti – môžu sa vykonávať bez aktívneho spojenia servera a klienta, alebo flexibilita – vykonanie zmeny v rámci jedného agenta nevyžaduje zmeny na vykonávacom serveri.

3 Architektúra

V tejto časti predstavujeme architektúru systému, ktorý by mohol z kombinácie vykonávania mobilného kódu na izolovanom bezpečnom hardvérovom module (typu rozšíreného TPM) pripojeného k nedôveryhodným výpočtovým zdrojom jednoznačne profitovať. Celá architektúra je založená na mobilných webových službách spúšťaných na bezpečnom zariadení s vlastnosťami multiagentového správania (autonómnosť, proaktivita a pod.). Takáto architektúra (Obr. 1) vo všeobecnosti pozostáva zo siete pospájaných dôveryhodných (Trusted Server) aj nedôveryhodných serverov (Untrusted Server). TS zabezpečujú registre služieb, uzlov, používateľov a bezpečnostných modulov a ich verejných kľúčov, ako aj bázu agentov (mobilný kód webových služieb) a všeobecné systémové bezpečnostné politiky. Každý agent má priradené vlastnosti a schopnosti, ktoré sú pri vykonávaní konkrétnych procesov využívané podľa potreby. Vykonávanie procesu sme zasadili do scenára manažmentu krízovej situácie, kde je potrebné získavať informácie z viacerých nedôveryhodných zdrojov. Celý proces začína špecifikáciou typu problému formou dialógu. Ďalej sa systém (prostredníctvom zvoleného agenta) bude snažiť formou dialógu vyšpecifikovať, aký najzávažnejší problém daná krízová situácia vytvorila. Po identifikovaní druhu krízovej situácie a oblasti sa spustia akcie podľa generického plánu pre každú krízovú situáciu.



Obr. 1. Architektúra distribuovaného systému pre bezpečné vykonávanie mobilného kódu na báze agentov v nedôveryhodnom výpočtovom prostredí prostredníctvom dôveryhodnej bezpečnostnej hardvérovej platformy (DS).

Systém bude teda vedieť vygenerovať (autonómne alebo formou dialógu) možné generické plány riešenia vzniknutých problémov. V ďalšom kroku bude prebiehať špecifikácia kontextu pre vygenerovanie konkrétnych požiadaviek. Na základe požiadaviek sa vyhľadajú servery, ktoré sú relevantné pre vzniknutú situáciu a ktoré sú registrované v centrálnej databáze. Z databázy agentov sa vyberú také, ktoré sú schopné zo zistených serverov dôveryhodne „vytiahnuť“ požadované informácie o dostupných kapacitách a zaslať ich späť do centra na spracovanie. Systém na základe potvrdených disponibilných kapacít (personálnych, materiálnych, atď.) zostaví konkrétny plán riešenia krízovej situácie a problémov s ňou spojených. Posledným krokom je spustenie pripraveného krízového plánu pre danú situáciu.

4 Záver

V tomto príspevku sme sa venovali rôznym aspektom zabezpečenia utajenia a dôvery tak dát, ako aj výpočtu na strane vykonávateľa – servera. TPM je hardvérová platforma – čip, ktorý umožňuje riešiť niektoré z bezpečnostných výziev spracovania dát a vykonávania kódu v nedôveryhodnom prostredí. TPM však nespĺňa niektoré dôležité požiadavky potrebné pre bezpečnostný prvok sieťového počítania. Uvedené požiadavky sú zároveň požiadavkami na rozšírený modul typu TPM, ktorý by bol schopný spracovávať náročnejšie operácie, ako aj bezpečným spôsobom vykonávať

mobilný kód. Nakoniec uvádzame architektúru distribuovaného systému pre bezpečné vykonávanie mobilného kódu na báze agentov v nedôveryhodnom výpočtovom prostredí prostredníctvom dôveryhodnej bezpečnostnej hardvérovej platformy. Prezentované výsledky sú predovšetkým súčasťou prebiehajúcej práce (work in progress) na IP projekte 7. RP EÚ Secricom.

Táto práca bola podporená z nasledovných projektov SECRICOM SEC-2007-4.2-04, SEMCO-WS APVV-0391-06, VEGA No. 2/6103/6, VEGA 2/7098/27, AIIA APVV-0216-07, Commius FP7-213876.

Použitá literatúra

1. Trusted Computing Group, URL: https://www.trustedcomputinggroup.org/home
2. Trusted Platform Module, URL: https://www.trustedcomputinggroup.org/groups/tpm/Trusted_Platform_Module_Summary_04292008.pdf

Register autorov (Author Index)

Pavol Návrat, Valentino Vranić (Eds.)

WIKT 2008: 3rd Workshop on Intelligent and Knowledge Oriented Technologies

Zborník príspevkov/Conference proceedings

1. vydanie/1st edition

Vydalo/Published by
Nakladateľstvo STU

126 strán/pages, 100 kópií/copies

Tlač/Print Nakladateľstvo STU Bratislava
2009

ISBN 978-80-227-3027-3